

Lecture Notes in Networks and Systems 787

Abhishek Swaroop
Zdzislaw Polkowski
Sérgio Duarte Correia
Bal Virdee *Editors*

Proceedings of Data Analytics and Management


ICDAM 2023, Volume 3

 Springer

Lecture Notes in Networks and Systems

Volume 787

Series Editor

Janusz Kacprzyk , Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA, School of Electrical and Computer Engineering—FEEC, University of Campinas—UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering, Bogazici University, Istanbul, Türkiye

Derong Liu, Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering, KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Abhishek Swaroop · Zdzislaw Polkowski ·
Sérgio Duarte Correia · Bal Virdee
Editors

Proceedings of Data Analytics and Management

ICDAM 2023, Volume 3

 Springer

Editors

Abhishek Swaroop
Department of Information Technology
Bhagwan Parshuram Institute
of Technology
New Delhi, Delhi, India

Sérgio Duarte Correia
Polytechnic Institute of Portalegre
Portalegre, Portugal

Zdzislaw Polkowski
Jan Wyzkowski University
Polkowice, Poland

Bal Virdee
Centre for Communications Technology
London Metropolitan University
London, UK

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-981-99-6549-6

ISBN 978-981-99-6550-2 (eBook)

<https://doi.org/10.1007/978-981-99-6550-2>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.

The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

ICDAM-2023 Steering Committee Members

Patrons

Prof. (Dr.) Don MacRaid, Pro-Vice Chancellor, London Metropolitan University, London

Prof. (Dr.) Wioletta Palczewska, Rector, The Karkonosze State University of Applied Sciences in Jelenia Góra, Poland

Prof. (Dr.) Beata Telązka, Vice-Rector, The Karkonosze State University of Applied Sciences in Jelenia Góra

General Chairs

Prof. Dr. Janusz Kacprzyk, Polish Academy of Sciences, Systems Research Institute, Poland

Prof. Dr. Karim Ouazzane, London Metropolitan University, London

Prof. Dr. Bal Virdee, London Metropolitan University, London

Prof. Cesare Alippi, Polytechnic University of Milan, Italy

Honorary Chairs

Prof. Dr. Aboul Ella Hassanien, Cairo University, Egypt

Prof. Dr. Vaclav Snasel, Rector, VSB-Technical University of Ostrava, Czech Republic

Prof. Chris Lane, London Metropolitan University, London

Conference Chairs

Prof. Dr. Vassil Vassilev, London Metropolitan University, London
Dr. Pancham Shukla, Imperial College London, London
Prof. Dr. Mak Sharma, Birmingham City University, London
Dr. Shikun Zhou, University of Portsmouth
Dr. Magdalena Baczyńska, Dean, The Karkonosze State University of Applied Sciences in Jelenia Góra, Poland
Dr. Zdzislaw Polkowski, Adjunct Professor KPSW, The Karkonosze State University of Applied Sciences in Jelenia Góra
Prof. Dr. Abhishek Swaroop, Bhagwan Parshuram Institute of Technology, Delhi, India
Prof. Dr. Anil K. Ahlawat, Dean, KIET Group of Institutes, India

Technical Program Chairs

Dr. Shahram Salekzamankhani, London Metropolitan University, London
Dr. Mohammad Hossein Amirhosseini, University of East London, London
Dr. Sandra Fernando, London Metropolitan University, London
Dr. Qicheng Yu, London Metropolitan University, London
Prof. Joel J. P. C. Rodrigues, Federal University of Piauí (UFPI), Teresina—PI, Brazil
Dr. Ali Kashif Bashir, Manchester Metropolitan University, United Kingdom
Dr. Rajkumar Singh Rathore, Cardiff Metropolitan University, United Kingdom

Conveners

Dr. Ashish Khanna, Maharaja Agrasen Institute of Technology (GGSIPTU), New Delhi, India
Dr. Deepak Gupta, Maharaja Agrasen Institute of Technology (GGSIPTU), New Delhi, India

Publicity Chairs

Dr. Józef Zaprucki, Prof. KPSW, Rector's Proxy for Foreign Affairs, The Karkonosze State University of Applied Sciences in Jelenia Góra
Dr. Umesh Gupta, Bennett University, India
Dr. Puneet Sharma, Assistant Professor, Amity University, Noida

Dr. Deepak Arora, Professor and Head (CSE), Amity University, Lucknow Campus
João Matos-Carvalho, Lusófona University, Portugal

Co-conveners

Mr. Moolchand Sharma, Maharaja Agrasen Institute of Technology, India
Dr. Richa Sharma, London Metropolitan University, London

Preface

We hereby are delighted to announce that The London Metropolitan University, London, in collaboration with The Karkonosze University of Applied Sciences, Poland, Politécnico de Portalegre, Portugal, and Bhagwan Parshuram Institute of Technology, India, has hosted the eagerly awaited and much coveted International Conference on Data Analytics and Management (ICDAM-2023). The fourth version of the conference was able to attract a diverse range of engineering practitioners, academicians, scholars, and industry delegates, with the reception of abstracts including more than 7000 authors from different parts of the world. The committee of professionals dedicated toward the conference is striving to achieve a high-quality technical program with tracks on Data Analytics, Data Management, Big Data, Computational Intelligence, and Communication Networks. All the tracks chosen in the conference are interrelated and are very famous among present-day research community. Therefore, a lot of research is happening in the above-mentioned tracks and their related sub-areas. More than 1200 full-length papers have been received, among which the contributions are focused on theoretical, computer simulation-based research, and laboratory-scale experiments. Among these manuscripts, 190 papers have been included in the Springer proceedings after a thorough two-stage review and editing process. All the manuscripts submitted to the ICDAM-2023 were peer-reviewed by at least two independent reviewers, who were provided with a detailed review pro forma. The comments from the reviewers were communicated to the authors, who incorporated the suggestions in their revised manuscripts. The recommendations from two reviewers were taken into consideration while selecting a manuscript for inclusion in the proceedings. The exhaustiveness of the review process is evident, given the large number of articles received addressing a wide range of research areas. The stringent review process ensured that each published manuscript met the rigorous academic and scientific standards. It is an exalting experience to finally see these elite contributions materialize into the four book volumes as ICDAM proceedings by Springer entitled “Proceedings of Data Analytics and Management: ICDAM 2023”.

ICDAM-2023 invited four keynote speakers, who are eminent researchers in the field of computer science and engineering, from different parts of the world. In addition to the plenary sessions on each day of the conference, 17 concurrent technical sessions are held every day to assure the oral presentation of around 190 accepted papers. Keynote speakers and session chair(s) for each of the concurrent sessions have been leading researchers from the thematic area of the session. The delegates were provided with a book of extended abstracts to quickly browse through the contents, participate in the presentations, and provide access to a broad audience of the audience. The research part of the conference was organized in a total of 22 special sessions. These special sessions provided the opportunity for researchers conducting research in specific areas to present their results in a more focused environment.

An international conference of such magnitude and release of the ICDAM-2023 proceedings by Springer has been the remarkable outcome of the untiring efforts of the entire organizing team. The success of an event undoubtedly involves the painstaking efforts of several contributors at different stages, dictated by their devotion and sincerity. Fortunately, since the beginning of its journey, ICDAM-2023 has received support and contributions from every corner. We thank them all who have wished the best for ICDAM-2023 and contributed by any means toward its success. The edited proceedings volumes by Springer would not have been possible without the perseverance of all the steering, advisory, and technical program committee members.

All the contributing authors owe thanks from the organizers of ICDAM-2023 for their interest and exceptional articles. We would also like to thank the authors of the papers for adhering to the time schedule and for incorporating the review comments. We wish to extend my heartfelt acknowledgment to the authors, peer-reviewers, committee members, and production staff whose diligent work put shape to the ICDAM-2023 proceedings. We especially want to thank our dedicated team of peer-reviewers who volunteered for the arduous and tedious step of quality checking and critique on the submitted manuscripts. We wish to thank our faculty colleague Mr. Moolchand Sharma for extending their enormous assistance during the conference. The time spent by them and the midnight oil burnt is greatly appreciated, for which we will ever remain indebted. The management, faculties, administrative and support staff of the college have always been extending their services whenever needed, for which we remain thankful to them.

Lastly, we would like to thank Springer for accepting our proposal for publishing the ICDAM-2023 conference proceedings. Help received from Mr. Aninda Bose, the acquisition senior editor, in the process has been very useful.

New Delhi, India
Polkowice, Poland
Portalegre, Portugal
London, UK

Abhishek Swaroop
Zdzislaw Polkowski
Sérgio Duarte Correia
Bal Virdee

Contents

Enhancing Computational Thinking Based on Virtual Robot of Artificial Intelligence Modeling in the English Language Classroom	1
Muthmainnah, Ahmad J. Obaid, Ahmad Al Yakin, and Mohammed Brayyich	
Software Change Prediction Model Using Ensemble Learning	13
Sanjay Patidar, Madhvan Sharma, and Himesh Mahabi	
Discerning Monkeypox from Other Viruses of the Poxviridae Family in a Deep Learning Paradigm	23
Malti Bansal, Riyanshi Arora, Sai Keshari, and Sakshi Panchal	
An Exploratory Study to Classify Brain Tumor Using Convolutional Neural Networks	43
Manmeet Singh, Manav Misra, Jayesh Jain, Mayank Goel, and Kumud Kundu	
Skin Cancer Detection with Edge Devices Using YOLOv7 Deep CNN	55
Dhruba Datta, Harsh Prakash, and Priya Singh	
Effective Image Captioning Using Multi-layer LSTM with Attention Mechanism	65
Janpit Singh, Kishan Kumar Garg, and Arahant Panwar	
A Hybrid Approach for Sentiment Analysis Using Game Theory in Word Sense Disambiguation	75
Aryan Singhania, Harsh Gupta, and Minni Jain	
A Sequential News Capture and Summarization Model (SNCSM)	85
Narayan Jee Jha, Rishav Sinha, Sameer Kumar, and Trasha Gupta	

A Novel Explainable Artificial Intelligence-Based Deep Reinforcement Learning for Secured Smart City Applications 103
Vandana Sharma, Tamizharasi Seetharaman, K Mohammed Essam, and Ahmed Alkhayat

A Review of IoT Security Solutions Using Machine Learning and Deep Learning 115
Anamika Chauhan and Kapil Sharma

A Study on High-Resolution Algorithms MUSIC, MVDR, ESPRIT, Beamscan, and Root-MUSIC for Narrowband Signals 133
Meenal Job and Ram Suchit Yadav

TwT: A Texture weighted Transformer for Medical Image Classification and Diagnosis 145
Mrigank Sondhi, Ayush Sharma, and Ruchika Malhotra

Automatic Keyphrase Extraction Using Fuzzy-Based Evolutionary Game Theory Approach 159
Minni Jain, Rajni Jindal, and Amita Jain

Effective Machine Learning-Based Heart Disease Prediction Model 169
Sandeep Kumar Saini and Garima Chandel

Internet of Things (IoT) Based Smart Agriculture and Automatic Irrigation Monitoring System Using LoRa 181
Kalathiripi Rambabu, Sanjay Dubey, Keshavagari Srujana, Gunnala Rajesh, and Mohammed Imran

Image Restoration Using ResNet–VGG Autoencoder Model 195
K. Venu Gopal, Mullangi David, Shaik Abdul Riyaz, and Perepi Durga Teja

DWT-HOG-Based Facial Expression Recognition System 205
Ahmed Abdulateef Mohammed, Faiz Al-Alawy, and Hashem Bedr Jehlol

A Multi-level Optimized Strategy for Imbalanced Data Classification Based on SMOTE and AdaBoost 223
A. Sarvani, Yalla Sowmya Reddy, Y. Madhavi Reddy, R. Vijaya, and Kampa Lavanya

A Protocol for Mutual Authentication in Remote Keyless Entry Systems that Employs Random Variables 239
A. Nguyen Thi Thuy

Approximated Sparsity Regularization Factor for Monaural Speech Separation 251
Garima Chandel, P. P. Muhammed Shanir, Yash Vardhan Varshney, and Setu Garg

Depression Level Analysis Using Face Emotion Recognition Method . . . 265
 Sudarshan Khandelwal, Shridhar Sharma, Suyash Agrawal,
 Gayatri Kalshetti, Bindu Garg, and Rachna Jain

**Early Prediction and Detection of Anxiety Level Using Support
 Vector Machine** 279
 Tisha Sadariya and Shanti Verma

**Empirical Analysis of Depression Detection Using Deep Learning
 on Twitter** 293
 Arunima Jaiswal, Payal Porwal, Anushka Singh, Pooja Kumari,
 Priyadeep Bhalla, and Nitin Sachdeva

**Assessment of Driver Fatigue and Drowsiness Based on Eye Blink
 Rate** 311
 Samarpit Karar and Tirupathiraju Kanumuri

**Selection of Robust Text-Based CAPTCHA Using TensorFlow
 Object Detection Method** 325
 R. Menaka and G. Padmavathi

**Performance Analysis of ECC-Based Security Solutions
 for Internet of Medical Things** 337
 Anuj Kumar Singh and Sachin Kumar

**Water Quality Monitoring and Evaluation Using Internet
 of Things and Machine Learning** 349
 Pravin Vilasrao Sawant and Y. M. Patil

**A QoS Enabled Automatic Fallback Handover Mechanism
 for Future Generation Wireless Networks** 365
 Ronitt Mehra, Palash, Reshav Kalyani, and Manjeet Kumar

Bone Fracture Detection Using CNN 379
 Sai Prudhvi Vallurupalli and T. Anuradha

Smart Parking System Using YOLOv3 Deep Learning Model 387
 Rishabh Tater, Preeti Nagrath, Jyoti Mishra,
 Victor Hugo C. de Albuquerque, and José Wally M. Menezes

**Crop Prediction Using Machine Learning with CRISP-DM
 Approach** 399
 Lendy Rahmadi, Hadiyanto, Ridwan Sanjaya, and Arif Prambayun

**Lung Cancer Detection and Classification Model Using Inception
 V3 Algorithm** 423
 Sitaram Meena, Amod Kumar, Meenakshi Sood,
 and Rajesh Kumar Meena

Real-Time Recognition of Handwritten Characters Using CNN 435
 Ritesh Kumar and Ritik Rao

Grid Search-Optimized Artificial Neural Network for Heterogeneous Cross-Project Defect Prediction	447
Ruchika Malhotra and Shweta Meena	
Design of Smart Weed Detection and Evacuation Robot Using TensorFlow Model Maker	459
P. Jothilakshmi, C. Gomatheeswari Preethika, and R. Mohanasundaram	
Document Store Schema Design Alternatives and Their Impact	471
Monika Shah and Amit Kothari	
On Significance of Subword Tokenization for Low-Resource and Efficient Named Entity Recognition: A Case Study in Marathi	483
Harsh Chaudhari, Anuja Patil, Dhanashree Lavekar, Pranav Khairnar, Raviraj Joshi, and Sachin Pande	
Analysis and Study of Bug Classification Quintessence and Techniques for Forecasting Software Faults	495
Shallu Juneja, Gurjit Singh Bhathal, and Brahmaleen K. Sidhu	
Hybrid Cryptography and Steganography Method to Provide Safe Data Transmission in IoT	513
Atrayee Majumder Ray, Sabyasachi Pramanik, Biplab Das, and Ashish Khanna	
Assessing the Efficacy of Different BERT Variants for Distinguishing Types of Cyberbullying on Twitter	525
Ashwin Prajeeth, Binav Gautam, and Garima Chhikara	
Design and Development of Evolutionary Algorithms for MPPT in a Solar PV System	537
Anurag Singh, Anirudh Saxena, and Anshika	
A Blockchain-Based Custom Clearance Solution for International Trade Using IPFS and Non-fungible Tokens	551
Mansimran Rehal, Rohit Ahuja, Divya Gandhi, and Ayush Sharma	
Probability-Based Load-Distribution Framework: To Reduce Latency in Fog Computing	565
Isha Dubey, Ekta Singh, Monika, and Deepak Kumar Sharma	
Edge-Graph Convolution Network: An Intrusion Detection Approach for Industrial IoT	585
Nilutpol Bora and Anamika Chauhan	
Retinal Blood Vessel Segmentation Using an EDADCN Architecture—Encoder–Decoder Architecture with Dilated Convolutions and Attention Mechanism	599
M. J. Carmel Mary Belinda, S. Alex David, E. Kannan, and N. Ruth Naveena	

SDB-RGSO: Swarm-Based Data Balancing and Randomized Grid Search Optimization for IoT NetFlow Malware Detection with Ensemble Machine Learning Model 615
D. Santhadevi and B. Janet

Fault Diagnosis of Electric Drives Using Ensemble Machine Learning Techniques 633
Shashank Paul and Abhishek Chaudhary

Using Modified Whale Optimization Algorithm for Improving the Performance of Ambulance Service 647
Hina Gupta and Zaheeruddin

An Integrated Model for Acceptance of QR Code Mobile Payment: A Comparative Study Between Male and Female 659
Priyanka Yadav, Anshul Jain, and Khyati Kochhar

Performance Comparison of Various YOLO Models for Vehicle Detection: An Experimental Study 677
Sourajit Maity, Arpan Chakraborty, Pawan Kumar Singh, and Ram Sarkar

Author Index 685

Editors and Contributors

About the Editors

Prof. (Dr.) Abhishek Swaroop completed his B.Tech. (CSE) from GBP University of Agriculture and Technology, M.Tech. from Punjabi University Patiala, and Ph.D. from NIT Kurukshetra. He has industrial experience of 8 years in organizations like Usha Rectifier Corporations and Envirotech Instruments Pvt. Limited. He has 22 years of teaching experience. He has served in reputed educational institutions such as Jaypee Institute of Information Technology, Noida, Sharda University Greater Noida, and Galgotias University Greater Noida. He has served at various administrative positions such as Head of the Department, Division Chair, NBA Coordinator for the university, and Head of training and placements. Currently, he is serving as Professor and HoD, Department of Information Technology in Bhagwan Parshuram Institute of Technology, Rohini, and Delhi. He is actively engaged in research. He has more than 60 quality publications, out of which eight are SCI and 16 Scopus.

Prof. (Dr.) Zdzislaw Polkowski is Adjunct Professor at Faculty of Technical Sciences at the Jan Wyzykowski University, Poland. He is also Rector's Representative for International Cooperation and Erasmus Program and Former Dean of the Technical Sciences Faculty during the period of 2009–2012 His area of research includes management information systems, business informatics, IT in business and administration, IT security, small medium enterprises, CC, IoT, big data, business intelligence, and block chain. He has published around 60 research articles. He has served the research community in the capacity of Author, Professor, Reviewer, Keynote Speaker, and Co-editor. He has attended several international conferences in the various parts of the world. He is also playing the role of Principal Investigator.

Prof. Sérgio Duarte Correia received his Diploma in Electrical and Computer Engineering from the University of Coimbra, Portugal, in 2000, the master's degree in Industrial Control and Maintenance Systems from Beira Interior University, Covilhã, Portugal, in 2010, and the Ph.D. in Electrical and Computer Engineering from the

University of Coimbra, Portugal, in 2020. Currently, he is Associate Professor at the Polytechnic Institute of Portalegre, Portugal. He is Researcher at COPELABS—Cognitive and People-centric Computing Research Center, Lusófona University of Humanities and Technologies, Lisbon, Portugal, and Valoriza—Research Center for Endogenous Resource Valorization, Polytechnic Institute of Portalegre, Portalegre, Portugal. Over past 20 years, he has worked with several private companies in the field of product development and industrial electronics. His current research interests are artificial intelligence, soft computing, signal processing, and embedded computing.

Prof. Bal Virdee graduated with a B.Sc. (Engineering) Honors in Communication Engineering and M.Phil. from Leeds University, UK. He obtained his Ph.D. from University of North London, UK. He was worked as Academic at Open University and Leeds University. Prior to this, he was Research and Development Electronic Engineer in the Future Products Department at Teledyne Defence (formerly Filtronic Components Ltd., Shipley, West Yorkshire) and at PYE TVT (Philips) in Cambridge. He has held numerous duties and responsibilities at the university, i.e., Health and Safety Officer, Postgraduate Tutor, Examination's Officer, Admission's Tutor, Short Course Organizer, Course Leader for M.Sc./M.Eng. Satellite Communications, B.Sc. Communications Systems, and B.Sc. Electronics. In 2010. He was appointed Academic Leader (UG Recruitment). He is Member of ethical committee and Member of the school's research committee and research degrees committee.

Contributors

Suyash Agrawal Department of CSBS, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

Rohit Ahuja Thapar Institute of Engineering and Technology Patiala, Patiala, India

Faiz Al-Alawy PSC-Inc., Michigan, USA

Ahmad Al Yakin Teacher Training and Education Faculty Civic Education Study Program, Universitas Al Asyariah Mandar, Polewali Mandar, Indonesia

Ahmed Alkhayyat College of Technical Engineering, The Islamic University, Najaf, Iraq

Anshika Department of Electrical Engineering, Delhi Technological University, New Delhi, India

T. Anuradha Department of IT, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, AP, India

Riyanshi Arora Department of Electronics and Communication Engineering, Delhi Technological University (DTU), Delhi, India

Malti Bansal Department of Electronics and Communication Engineering, Delhi Technological University (DTU), Delhi, India

M. J. Carmel Mary Belinda Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India

Priyadeep Bhalla Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, India

Gurjit Singh Bhathal Computer Science Engineering Department, Punjabi University, Patiala, India

Nilutpol Bora Department of Information Technology, Delhi Technological University, New Delhi, Delhi, India

Mohammed Brayyich National University of Science and Technology, Thi-Qar, Iraq

Arpan Chakraborty Department of Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Mullick Road, Kolkata, 700032 West Bengal, India

Garima Chandel Department of Electronics and Communication Engineering, Chandigarh University, Mohali, Chandigarh, India

Harsh Chaudhari Pune Institute of Computer Technology, Pune, Maharashtra, India;
L3Cube, Pune, Maharashtra, India

Abhishek Chaudhary Delhi Technological University, Rohini, Delhi, India

Anamika Chauhan Department of Information Technology, Delhi Technological University, New Delhi, Delhi, India

Garima Chhikara Department of Computer Science and Engineering, Delhi Technological University, New Delhi, Delhi, India

Biplab Das Haldia Institute of Technology, Haldia, India

Dhruba Datta Delhi Technological University, Rohini, New Delhi, India

Mullangi David Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

S. Alex David Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunathala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India

Victor Hugo C. de Albuquerque Department of Teleinformatics Engineering (DETI), Fortaleza, Brazil

Isha Dubey Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India

Sanjay Dubey Department of Electronics and Communication Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India

Divya Gandhi Thapar Institute of Engineering and Technology Patiala, Patiala, India

Bindu Garg Department of CSBS, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

Kishan Kumar Garg Delhi Technological University, New Delhi, India

Setu Garg Department of Electronics and Communication Engineering, I.T.S. Engineering College, Greater Noida, India

Binav Gautam Department of Computer Science and Engineering, Delhi Technological University, New Delhi, Delhi, India

Mayank Goel Inderprastha Engineering College, Sahibabad, Uttar Pradesh, India

C. Gomatheeswari Preethika Sri Venkateswara College of Engineering, Sriperumbudur, Tamilnadu, India

Harsh Gupta Department of Computer Engineering, Delhi Technological University, Delhi, India

Hina Gupta Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi, Delhi, India

Trasha Gupta Department of Applied Mathematics, Delhi Technological University, New Delhi, India

Hadiyanto Diponegoro University, Semarang, Indonesia

Mohammed Imran Department of Electronics and Communication Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India

Amita Jain Computer Science and Engineering, Netaji Subhas University of Technology, Delhi, India

Anshul Jain FMS-WISDOM, Banasthali Vidyapith, Tonk, Rajasthan, India

Jayesh Jain Inderprastha Engineering College, Sahibabad, Uttar Pradesh, India

Minni Jain Department of Computer Engineering, Delhi Technological University, Delhi, India

Rachna Jain Department of Information Technology, Bhagwan Parshuram Institute of Technology, New Delhi, India

Arunima Jaiswal Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, India

B. Janet National Institute of Technology, Tiruchirappalli, India

Hashem Bedr Jehlol Mustansiriyah University, Information Technology Center, Baghdad, Iraq

Narayan Jee Jha Department of Applied Mathematics, Delhi Technological University, New Delhi, India

Rajni Jindal Computer Science and Engineering, Delhi Technological University, Delhi, India

Meenal Job Department of Electronics and Communication, University of Allahabad, Prayagraj, India

Raviraj Joshi Indian Institute of Technology Madras, Chennai, Tamilnadu, India; L3Cube, Pune, Maharashtra, India

P. Jothilakshmi Sri Venkateswara College of Engineering, Sriperumbudur, Tamilnadu, India

Shallu Juneja Computer Science Engineering Department, Punjabi University, Patiala, India;
Computer Science Engineering Department, Maharaja Agrasen Institute of Technology, New Delhi, Delhi, India

Gayatri Kalshetti Department of CSBS, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

Reshav Kalyani Delhi Technological University, Delhi, India

E. Kannan Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunathala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India

Tirupathiraju Kanumuri Department of Electrical Engineering, National Institute of Technology, Delhi, India

Samarpit Karar Department of Electrical Engineering, National Institute of Technology, Delhi, India

Sai Keshari Department of Electronics and Communication Engineering, Delhi Technological University (DTU), Delhi, India

Pranav Khairnar L3Cube, Pune, Maharashtra, India;
Pune Institute of Computer Technology, Pune, Maharashtra, India

Sudarshan Khandelwal Department of CSBS, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

Ashish Khanna Maharaja Agrasen Institute of Technology, New Delhi, India

Khyati Kochhar FMS-WISDOM, Banasthali Vidyapith, Tonk, Rajasthan, India

Amit Kothari Gujarat Technological University, Ahmedabad, India

Amod Kumar Department of ECE, NITTTR, Chandigarh, India

- Manjeet Kumar** Delhi Technological University, Delhi, India
- Ritesh Kumar** Delhi Technological University, New Delhi, India
- Sachin Kumar** South Ural State University, Chelyabinsk, Russian Federation
- Sameer Kumar** Department of Applied Mathematics, Delhi Technological University, New Delhi, India
- Pooja Kumari** Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, India
- Kumud Kundu** Inderprastha Engineering College, Sahibabad, Uttar Pradesh, India
- Kampa Lavanya** Department of Computer Science and Engineering, University College of Sciences, Acharya Nagarjuna University, Guntur District, Andhra Pradesh, India
- Dhanashree Lavekar** Pune Institute of Computer Technology, Pune, Maharashtra, India;
L3Cube, Pune, Maharashtra, India
- Himesh Mahabi** Department of Software Engineering, Delhi Technological University, Delhi, India
- Sourajit Maity** Department of Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Mullick Road, Kolkata, 700032 West Bengal, India
- Ruchika Malhotra** Department of Software Engineering, Delhi Technological University, New Delhi, India
- Rajesh Kumar Meena** Department of Electronics, Rajesh Pilot Government Polytechnic College, Dausa, India
- Shweta Meena** Department of Software Engineering, Delhi Technological University, Delhi, India
- Sitaram Meena** Department of Electronics, Rajesh Pilot Government Polytechnic College, Dausa, India
- Ronitt Mehra** Delhi Technological University, Delhi, India
- R. Menaka** Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India
- José Wally M. Menezes** Federal Institute of Ceará, Fortaleza, CE, Brazil
- Jyoti Mishra** Department of Mathematics, Gyan Ganga Institute of Technology and Sciences, Jabalpur, M.P., India;
Federal Institute of Education, Science and Technology of Ceara, Fortaleza, Brazil
- Manav Misra** Inderprastha Engineering College, Sahibabad, Uttar Pradesh, India

Ahmed Abdulateef Mohammed Mustansiriya University, Information Technology Center, Baghdad, Iraq

K Mohammed Essam Department of AIML, Acharya Institute of Technology, Bangalore, India

R. Mohanasundaram Sri Venkateswara College of Engineering, Sriperumbudur, Tamilnadu, India

Monika Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India

P. P. Muhammed Shanir Department of Electrical and Electronics Engineering, TKM College of Engineering, Kollam, India

Muthmainnah Teacher Training and Education Faculty Civic Education Study Program, Universitas Al Asyariah Mandar, Polewali Mandar, Indonesia

Preeti Nagrath Bharati Vidyapeeth's College of Engineering, New Delhi, India

N. Ruth Naveena Department of Mathematics, Hindustan Institute of Technology & Science, Chennai, Tamil Nadu, India

A. Nguyen Thi Thuy Ho Chi Minh City University of Foreign Languages - Information Technology, Ho Chi Minh City, Vietnam

Ahmad J. Obaid Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq

G. Padmavathi Dean School of Physical Sciences and Computational Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India

Palash Delhi Technological University, Delhi, India

Sakshi Panchal Department of Electronics and Communication Engineering, Delhi Technological University (DTU), Delhi, India

Sachin Pande Pune Institute of Computer Technology, Pune, Maharashtra, India

Arahant Panwar Delhi Technological University, New Delhi, India

Sanjay Patidar Department of Software Engineering, Delhi Technological University, Delhi, India

Anuja Patil Pune Institute of Computer Technology, Pune, Maharashtra, India; L3Cube, Pune, Maharashtra, India

Y. M. Patil KIT College of Engineering, Kolhapur, Maharashtra, India

Shashank Paul Delhi Technological University, Rohini, Delhi, India

Payal Porwal Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, India

Ashwin Prajeeth Department of Computer Science and Engineering, Delhi Technological University, New Delhi, Delhi, India

Harsh Prakash Delhi Technological University, Rohini, New Delhi, India

Sabyasachi Pramanik Haldia Institute of Technology, Haldia, India

Arif Prambayun Sriwijaya State Polytechnic, Palembang, Indonesia

Lendy Rahmadi Diponegoro University, Semarang, Indonesia

Gunnala Rajesh Department of Electronics and Communication Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India

Kalathiripi Rambabu Department of Electronics and Communication Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India

Ritik Rao Delhi Technological University, New Delhi, India

Atrayee Majumder Ray Netaji Subhash Engineering College, Ranabhatia, India

Y. Madhavi Reddy Department of S&H, CVR College of Engineering, Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana, India

Yalla Sowmya Reddy Department of CSE-AIML, CVR College of Engineering, Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana, India

Mansimran Rehal Thapar Institute of Engineering and Technology Patiala, Patiala, India

Shaik Abdul Riyaz Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

Nitin Sachdeva Department, Galgotias College of Engineering and Technology, Greater Noida, India

Tisha Sadariya Department of Computer Applications, L. J. University, Ahmedabad, India

Sandeep Kumar Saini Department of Electronics and Communication Engineering, Chandigarh University, Mohali, India

Ridwan Sanjaya Soegijapranata Catholic University, Semarang, Indonesia

D. Santhadevi SCOPE, VIT-AP University, Amaravathi, Andhra Pradesh, India

Ram Sarkar Department of Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Mullick Road, Kolkata, 700032 West Bengal, India

A. Sarvani Department of Information Technology, Lakireddy Bali Reddy College of Engineering (Autonomous), Mylavaram, NTR District, Andhra Pradesh, India

Pravin Vilasrao Sawant Department of Technology, Shivaji University Kolhapur, Kolhapur, Maharashtra, India

Anirudh Saxena Department of Electrical Engineering, Delhi Technological University, New Delhi, India

Tamizharasi Seetharaman Department of CSE, School of Engineering and Technology, CMR University, Bangalore, India

Monika Shah Computer Science and Engineering Department, Nirma University, Ahmedabad, India

Ayush Sharma Department of Software Engineering, Delhi Technological University, New Delhi, India;
Thapar Institute of Engineering and Technology Patiala, Patiala, India

Deepak Kumar Sharma Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India

Kapil Sharma Department of Information Technology, Delhi Technological University, Delhi, India

Madhvan Sharma Department of Software Engineering, Delhi Technological University, Delhi, India

Shridhar Sharma Department of CSBS, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India

Vandana Sharma Amity Institute of Information Technology, Amity University, New Delhi, India

Brahmaleen K. Sidhu Computer Science Engineering Department, Punjabi University, Patiala, India

Anuj Kumar Singh Amity University Madhya Pradesh, Gwalior, India

Anurag Singh Department of Electrical Engineering, Delhi Technological University, New Delhi, India

Anushka Singh Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, India

Ekta Singh Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India

Japnit Singh Delhi Technological University, New Delhi, India

Manmeet Singh Inderprastha Engineering College, Sahibabad, Uttar Pradesh, India

Pawan Kumar Singh Department of Information Technology, Jadavpur University, Jadavpur University Second Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Salt Lake City, Kolkata, 700106 West Bengal, India

Priya Singh Delhi Technological University, Rohini, New Delhi, India

Aryan Singhania Department of Computer Engineering, Delhi Technological University, Delhi, India

Rishav Sinha Department of Applied Mathematics, Delhi Technological University, New Delhi, India

Mrigank Sondhi Department of Software Engineering, Delhi Technological University, New Delhi, India

Meenakshi Sood Department of Curriculum, NITTTR, Chandigarh, India

Keshavagari Srujana Department of Electronics and Communication Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India

Rishabh Tater Bharati Vidyapeeth's College of Engineering, New Delhi, India

Perepi Durga Teja Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

Sai Prudhvi Vallurupalli Department of IT, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, AP, India

Yash Vardhan Varshney Psychophysiology Lab, IIT Mumbai, Mumbai, India

K. Venu Gopal Department of Information Technology, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

Shanti Verma Department of Computer Applications, L. J. University, Ahmedabad, India

R. Vijaya Department of AI and IT, DVR & Dr HS MIC College of Technology, Kanchikacherla, NTR Distirct, Andhra Pradesh, India

Priyanka Yadav FMS-WISDOM, Banasthali Vidyapith, Tonk, Rajasthan, India

Ram Suchit Yadav Department of Electronics and Communication, University of Allahabad, Prayagraj, India

Zaheeruddin Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi, Delhi, India

Enhancing Computational Thinking Based on Virtual Robot of Artificial Intelligence Modeling in the English Language Classroom



Muthmainnah, Ahmad J. Obaid, Ahmad Al Yakin,
and Mohammed Brayyich

Abstract There has been a lack of both educational focus on computational thinking (CT) and research into effective methods of assessing CT competence. This research takes a comprehensive look at the methods used to grade CT in higher education. This study examines the use of educational robotics (ER) as a teaching strategy in higher education, with particular attention to how undergraduate students might respond differently to this kind of learning. Students collaborate using artificial intelligence to find out how to program robots (start installing, interact, sharing information, feedback, and evaluate). Of 181 undergraduates' students from Indonesia, Saudi Arabia, Riyadh, Pakistan, the Philippines, Egypt, Algeria, Lesoto, Cameron, Bangladesh, Sudan, Uganda, Portugal, Ukraine, Pakistan, and England make up the study groups. These students are enrolled in the first semester of the 2022 academic year. In this work, researchers employed a quantitative method technique. During the exercise, the participants were closely monitored to assess their performance in relation to the core CT constructs in the robotics class. These constructs include confidence in dealing with complexity, persistence in problem-solving, tolerance for ambiguity, ability to handle open-ended problems, effective communication skills, and teamwork abilities. In order to enhance the implementation of strategic initiatives. To facilitate more strategic learning, this study found that the use of visual robotics of artificial intelligence helped to developing CT and the respondents strong believe it to increase students' confidence in dealing with complexity, persistence in working through difficult questions, tolerance for ambiguity, ability to handle open problems,

Muthmainnah (✉) · A. Al Yakin
Teacher Training and Education Faculty Civic Education Study Program, Universitas Al Asyariah
Mandar, Polewali Mandar, Indonesia
e-mail: muthmainnahunasman@gmail.com

A. J. Obaid
Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq
e-mail: ahmedj.aljanaby@uokufa.edu.iq

M. Brayyich
National University of Science and Technology, Thi-Qar, Iraq
e-mail: m.r.brayyich@nust.edu.iq

communication skills, and teamwork. It is thought that this study would be valuable to academics, curriculum designers, and teachers because students who play artificial intelligence games exhibit considerably greater cognitive flexibility than before in higher education.

Keywords Computational thinking skills · Educational robotics · Artificial intelligence · Foreign language and higher education

1 Introduction

The advancement of computer technology has had far-reaching consequences on all facets of modern society and economy. But, in today's world, practically everyone, regardless of age, is expected to have at least rudimentary computer skills to keep up with the constantly evolving state of technology [1]. To tackle issues in the future, [2–4] argue that we must educate students to be “future-ready” in the sense that they have the skills to make use of technology about which we know nothing at this time. It is to be assumed that individuals will have varying levels of skill and knowledge in this setting. As amazing as these technological advancements are, it is essential that individuals of all ages have at least some competences with computers.

Computational thinking is becoming increasingly important in education and daily life. In fact, it's predicted that by the middle of the twenty-first century [1, 5], it will be more important than traditional skills like reading, writing, and arithmetic. Educational robots are being used to teach students teamwork, problem solving, creativity, critical thinking, and computational thinking. However, [6, 7] the use of educational robotics is still limited in many countries, despite the abundance of related research projects.

In Reference [5], computational thinking applies computer science concepts to solve problems, design systems, and understand human behavior. It's not just coding but includes the mindset of a problem solver with cognitive abilities to analyze and interpret data. It involves imperative thinking abilities like abstraction, decomposition, and heuristic reasoning to address complex problems. According to [8], CT is crucial in the twenty-first century for professionals in all fields. Educators stress its importance for analytical skills, logical reasoning, and problem solving.

Having a strong education in computer technology (CT) is crucial in the twenty-first century, as digital technologies are present in almost all aspects of modern life. Governments around the world are emphasizing the importance of CT education for current and future generations, including the USA where the Computer Science Teachers Association and International Society for Technology in Education offer resources and workshops for teachers [9]. European countries are also implementing CT to improve analytical and conceptual thinking, while Asian nations such as Korea, Taiwan, Hong Kong, and China are making national curricular changes to address the current movement in CT education due to its importance to the ICT sector [10, 11].

Identifying appropriate activities and materials for different ages of Gen Z learners is crucial as new technology, like smartphones and electronic toys, become more popular; [12, 13] studies show that introducing robotics and computer science to students can have a positive impact on their development. Even young children can learn to build and program simple robotics projects. For undergraduates, robotics learning institutions can help them gain exposure to important engineering and technology concepts while improving their computational thinking skills [14, 15]. Using robots in education can promote teamwork, coordination, and creativity. Students can even embed robots in their smartphones to develop critical thinking and imagination.

This research emphasizes the importance of confidence, persistence, ambiguity tolerance, problem solving, communication skills, and teamwork. While computational thinking has been increasing in popularity, its transferability to other contexts is still uncertain. This paper aims to integrate visual robotics of artificial intelligence within the framework of computational thinking theory to address the challenges of implementing CT in language learning. It outlines the process of solving CT problems, proposes a framework, and explores how to apply this methodology in higher education. The paper concludes with recommendations for further research.

2 Method

2.1 Participants

This study examined the effectiveness of visual robotics and artificial intelligence in improving computational thinking skills among undergraduate students from different countries. The study sample consisted of 181 students aged between 18 and 21 years old, with 84 females and 97 males, who had no prior experience in accessing CT activities. To maximize CT skills and demonstrate the relevance of computer science principles in daily life, six six-week courses were designed and conducted in English. Observations were made of the students' CT activities using a guide to gain an understanding of the problem and develop workable solutions. The study's findings demonstrate the potential of visual robotics and artificial intelligence in enhancing students' computational thinking skills. These results suggest that incorporating such technologies into educational curricula can be a promising approach to improve students' CT skills and prepare them for future technological advancements.

2.2 Questionnaire

This study used a survey via Google Form to measure students' computational thinking (CT). Participants rated their level of agreement with 33 CT statements based

on [16] using a 5-point scale. The statements focused on the use of virtual robots and artificial intelligence to enhance CT. After six meetings, participants reported on their development of CT and language skills using a virtual robot in a diary uploaded to each undergraduate student's YouTube channel. Learning outcomes were monitored during the exercise.

2.3 The Procedure of Robotics Instructional Design (RID) in English Class

Interactive activities in foreign language classes promote the development of students' critical thinking skills and English language proficiency. The AI application called "My Virtual Dream Friends" Booble, John English, and Andy to engage students in conversations about self-introduction, food, travel, work, sports, shapes, and colors. Before starting the CT assignment, students are taught AI guidelines and programming. Through this project, we aim to study a transition phase between robotics and students and test if students could confidently use target language learning while developing their CT skills [17, 18].

Data Analysis

This study used research instruments to gather data on computational thinking skills. A diagnostic questionnaire was administered to participants from various universities using Google Form and social media; they are Majmaah University Saudi Arabia, Universitas Raden Fatah Palembang, Indonesia, Ngaoundere University, University PGRI Ronggolawe Tuban, Indonesia, Adrar African University, Bara Gobindapur Ali Miah Bhuiyan High School, Cristina B Gonzales Memorial High School, University of Hafr Al Batin, Queen Mary College, Dongola, Polytechnic Institute of Portalegre, Yes You Can International Academy, Saint Victor Academy, National university of Lesotho, Kyiv National Economic University, Magmaah University, University of South Asia Lahore Pakistan, Annamalai University, Mercuri Buana University Yogyakarta, Ramah College, LCWU, Gaddani National High School, Hs Durbal bimna Budgam, Siliwangi University, National Institute for Biotechnology and Genetic Engineering, Redeemed College of Mission, Hafar Al Batin University, Magma University, IAIN Pontianak, Indonesia, University of Abra Philipina, and IAIN Curup Bengkulu, Indonesia.

The study used virtual robotics of artificial intelligence to improve computational thinking and linguistic abilities of undergraduate students. The instructor's use of robotics-AI strategy in the classroom was observed to enhance the proficiency of students in computational thinking. Participant quotations were used as case examples throughout the findings section (Fig. 1).

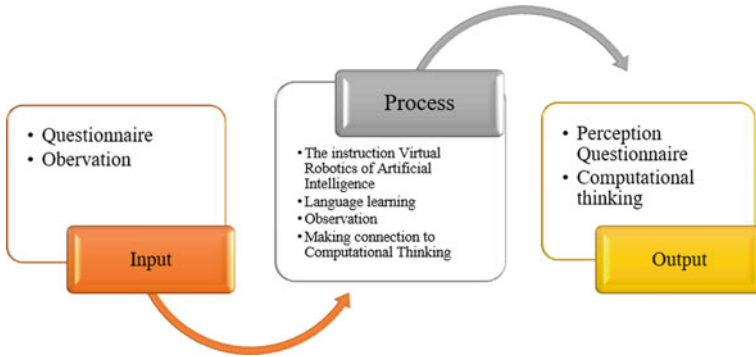


Fig. 1 Research procedure

3 Results and Discussions

3.1 Questionnaire and Observation Results of CT Through Virtual Robotics of Artificial Intelligence

In the next section, we present the results of our investigation of the problem we posed, namely how effective virtual robotics of artificial intelligence is on computational thinking skills of undergraduate students, and how participants feel about using them to learn English as a foreign language.

Table 1 shows that the average score for the five questions regarding the impact of computational thinking on language learning using virtual robotics for online learning was between 63 and 74. This implies that most students believe that their computational thinking skills improve with this teaching method, as they responded with “strongly agree” or “agree.” The standard deviation indicates that most students have similar levels of understanding.

Based on Table 2, the nine measures of persistence in working on difficult questions were considered useful by respondents. The average calculated is in the range of 63 to 65 indicating that most respondents find virtual robotics from artificial intelligence useful for improving their computational thinking on this criteria. Since their positive response on persistence in working on difficult questions, most of them 50% took the answer. It can be concluded that their opinions about the effectiveness of AI virtual robotics are distributed in the same way.

Table 3 reveals the mean and standard deviation for the four items. The tolerance for ambiguity aspects ranged from 67 to 68, and most of them their answer took strongly agree and agree, respectively. This shows that the opinion of respondents about the use of artificial intelligence virtual robotics as an effective learning strategy to develop computational thinking.

Table 4 described the mean and standard deviation for the four items. Ability to handle open problems aspects ranged from 63 to 67. Of 50 percent respondent believe

Table 1 Students' perception on computational thinking skill of confidence in dealing with complexity aspect

Confidence in dealing with complexity	SA	A	N	D	SD
When I learn English, I have ability to think differently	37.6	38.7	21.5	1.1	1.1
When I learn English, I have ability to think differently	28.2	32.6	28.7	5.5	3.9
When I learn English, I can change of organizational structure	24.3	26.0	34.3	11.0	2.2
When I learn English, I have innovative vision of learning	28.2	33.7	26.0	7.7	2.8
When I learn English, I have innovative vision of learning	29.3	29.8	28.2	8.3	1.7

Table 2 Students' perception on computational thinking skill of persistence in working on difficult questions aspect

Persistence in working on difficult questions	SA	A	N	D	SD
I study English in order to formulating problems and using computers and other applications to solve them	32.0	28.2	27.1	5.5	4.4
I study English I would feel logically organizing and analyzing	27.6	37.0	22.7	6.6	4.4
Logically organizing and analyzing	24.9	31.5	28.7	7.2	3.9
Deciding which details in a problem need to be highlighted and which ones can be ignored, also known as the abstraction process in language learning	24.3	34.8	26.5	9.4	3.3
Representing data with models and simulations in language learning	21.5	29.8	35.4	7.2	3.9
Automating solutions through a series of steps	27.1	28.2	32.0	8.3	2.2
I learn language with identifying, analyzing, and implementing possible solutions by using the most efficient and effective combination of steps and resources	24.9	35.9	24.3	6.6	4.4
Identifying, analyzing, and implementing possible solutions by using the most efficient and effective combination of steps and resources	29.3	30.4	27.6	7.2	3.3
Generalizing and transferring this problem-solving process to a wide variety of problems	23.8	27.1	33.1	8.3	6.1

Table 3 Students' perception on computational thinking skill of tolerance for ambiguity aspect

Tolerance for ambiguity	SA	A	N	D	SD
My ability in problem solving increase very well in language learning	30.4	28.7	28.2	7.7	3.3
I believe my orientation increases by reading and understanding the problem more active language learner	32.6	29.3	27.6	6.6	2.2
I felt my skill to organize my planning into the right action (metacognitive skill) in language learning	29.8	29.8	29.3	6.6	1.7
My ability to implementing and verifying increase in language learning	98.3	35.9	27.6	2.2	5.5

Table 4 Students' perception on computational thinking skill of ability to handle open problems aspect

Ability to handle open problems	SA	A	N	D	SD
When I learn English, I am able to describe the approach that used to solve the problem (steps to be taken or general strategy to be used)	23.8	29.3	32.0	6.6	7.2
I felt by obtaining an intermediate correct or incorrect solution by checking the solution	27.1	27.1	30.9	6.6	6.6
I understand and stop working to see what has been done and where it is leading to reviews solution and corrects any errors	27.1	31.5	30.4	3.3	6.1

the use of artificial intelligence virtual robotics as an effective learning strategy to help them to handle problem open problem effectively.

Table 5 described the mean and standard deviation for the seven items. Ability to communicate each other aspects ranged from 66 to 68. This shows that the opinion of respondents about the use of artificial intelligence virtual robotics as an effective learning strategy to felt confidence to share their idea, argument and opinion to acquiring information, critique and interpreting data.

Table 6 shows that the mean and standard deviation for the four items working with others to achieve a common goal or solution aspects ranged from 66 to 68. This shows that the opinion of respondents about the use of artificial intelligence virtual robotics as an effective learning strategy to felt happy, engage and motivated working in group or doing collaboration as teamwork in foreign language classroom.

Table 5 Students' perception on computational thinking skill of ability to communicate aspect in robotics class

Ability to communicate	SA	A	N	D	SD
I work with language learning by creating, comparing, and analyzing different ways of presenting information. This can involve organizing or ordering information in various ways	27.1	31.5	30.4	3.3	6.1
I learn English because I believe that abstract and formal communication skills can be made concrete and manipulated by using robotic technology during the learning process	24.9	31.5	33.1	6.6	1.7
My communication skills develop into more important and explicit activities during discussions in language learning	30.4	29.8	32.0	3.9	2.2
My communication skills develop in interpreting data by using robotic	29.8	26.5	31.5	6.1	3.3
I learn English I have more involved in the argument from evidence	28.7	23.2	35.4	7.2	3.9
My skills improved in acquiring, evaluating, and communicating information	26.5	26.5	28.7	11.6	4.4
Construct viable arguments and critique the reasoning of others language learning	30.4	33.7	25.4	4.4	3.9

Table 6 Students' perception on computational thinking skill of working with others to achieve a common goal or solution aspect in robotics class

Working with others to achieve a common goal or solution	SA	A	N	D	SD
I'm happy working in group or team in learning or in English class	26.0	24.9	37.0	7.2	2.8
I'm happy with my group or team in learning or in the class	20.5	14.9	17.2	4.3	2.6
I more engage in learning with collaboration as a team	20.5	14.2	19.2	3.6	2.3
I motivate in learning and confidence in group work	21.5	17.2	15.2	3.0	2.6

3.2 Discussion

Based on findings in Table 3, students believe that artificial intelligence virtual robotics can improve their computational thinking and confidence in dealing with complexity in a robotics learning environment. Through decomposition, students can

break down complex problems into manageable pieces and actively provide feedback. This helps them feel more comfortable expressing themselves in English. Developing a student's CT attitude requires instruction in self-confidence and resilience. It is important to increase individual motivation to learn CT skills and believe in one's ability to plan and carry out required learning tasks. Similar results found by [19, 20] when they asked children about their computational thinking and their knowledge. The concept of CT provides fresh motivation for individuals to adapt to the complexities of today's ever-evolving high-tech world.

This study uses virtual robots powered by artificial intelligence to teach computational thinking to students. They incorporate game elements such as scoring criteria to increase student engagement and motivation. The study results in Table 4 show that a student-centered framework that focuses on intrinsic motivation and boosts computational thinking skills is effective in fostering active involvement in learning. Table 4 displays study results [21, 22] students develop skills such as persistence in solving difficult questions, which is important for making predictions.

Our CT procedures aim to teach students about the importance of using problem-solving strategies in real-life situations similar with [23] study. By advocating for a methodical and sequential approach to problem solving, we hope to make students aware of the benefits of computational thinking. This skill is valuable for students at all levels as it helps develop various skills.

CT is a powerful problem-solving skill set that draws on computer science theory, providing new problem-solving approaches and understanding of human behavior. This innovative approach is considered cutting-edge for education and has potential to benefit the future generation. This study also highlights the importance of information literacy as a means of effective use of online resources and integration of technology to enhance cognitive processes of analysis, synthesis, and invention similarly with [24] findings.

This study shows that computational thinking can be beneficial for English classes as it helps improve communication, reading, and writing skills. Using a robotics-based classroom design, lecturers and undergraduate students can work together to develop solutions and products in English. By incorporating computational thinking skills into the curriculum, undergraduate students can be encouraged to draw parallels between interactions with robots and humans. This can help achieve educational goals and offer benefits for higher education lecturers.

Table 6 summarizes how CT is a valuable tool for teaching critical thinking and problem solving in formal education. CT helps educators guide the next generation by using modern technology to develop automated solutions. By providing a shared language, materials, and peer support, CT can improve communication and problem-solving skills. While computers are often associated with problem solving, it's important to remember that problem solving is a mental ability, not just a digital one (Table 2, 4, and 5). Humans have been solving problems without technology for a long time. This finding related like [25] stated the use of computers is often automatically associated with computational problem-solving procedures in our modern, highly digital society.

On findings show that undergraduates are happier, more satisfied, and more motivated when they work together to solve problems. They can learn computational thinking using virtual robotics and artificial intelligence. To be technologically skilled nowadays, one must be adept at different media formats, evaluate information, and communicate ideas through digital media.

4 Conclusions

Computational thinking is a crucial skill for all undergraduate students, and there are now tools available to help students develop this skill. Applying virtual robotics of artificial intelligence can improve computing abilities and encourage initiative, discovery, and growth in students. To fully incorporate computational thinking across disciplines, more research is needed. However, students in any field can benefit from this skill in both university and the workplace.

References

1. Wing JM (2014) Computational thinking benefits society. In: 40th anniversary blog of social issues in computing, 2014, vol 26
2. Li L (2022) Reskilling and upskilling the future-ready workforce for industry 4.0 and beyond. *Inf Syst Front* 1–16
3. Low EL (2023). Rethinking teacher education in pandemic times and beyond. *Educ Res Policy Prac* 1–12
4. Ruge G, Webber R, Kalutara P (2021) Constructing pedagogical alignment for a sustainable mindset of future-ready graduates. 44th, 595
5. Wing JM (2006) Computational thinking. *Commun ACM* 49:33–35
6. Komis V, Romero M, Misirli A (2017) A scenario-based approach for designing educational robotics activities for co-creative problem solving. In: *Educational robotics in the makers era 1*. Springer International Publishing, pp 158–169
7. Alimisis D, Moro M, Arlegui J, Pina A, Frangou S, Papanikolaou K (2007) Robotics and constructivism in education: the TERECOP project. *EuroLogo*
8. Çoban E, Korkmaz Ö (2021) An alternative approach for measuring computational thinking: performance-based platform. *Think Skills Creativ* 42:100929
9. International Society for Technology in Education (ISTE) (2014) ISTE standards administrators. Retrieved from <http://www.iste.org/standards>
10. Mannila L, Dagiene V, Demo B, Grgurina N, Mirolo C, Rolandsson L, Settle A (2014) Computational thinking in K-9 education. In: *Proceedings of the working group reports of the 2014 on innovation and technology in computer science education conference*, pp 1–29
11. Wong KWG, Ching CC, Mark KP, Tang JK, Lei CU, Cheung HY, Chui HL (2015) Impact of computational thinking through coding in K-12 education: a pilot study in Hong Kong. *General Stud* 85(88.01):2–08
12. Alam A, Mohanty A (2023) Foundation for the future of higher education or ‘misplaced optimism’? Being human in the age of artificial intelligence. In: *Innovations in intelligent computing and communication: first international conference, ICIICC 2022, Bhubaneswar, Odisha, India, December 16–17, 2022, Proceedings*. Springer International Publishing, Cham, pp 17–29

13. Yang FCO, Lai HM, Wang YW (2023) Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Comput Educ* 195:104721
14. Tedre M, Denning PJ (2016) The long quest for computational thinking. In: *Proceedings of the 16th Koli calling international conference on computing education research*, pp 120–129
15. Yadav A, Hong H, Stephenson C (2016) Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends* 1–4
16. Denis B, Hubert S (2001) Collaborative learning in an educational robotics environment. *Comput Hum Behav* 17(5–6):465–480
17. Muthmainnah OAJ, Al Mahdawi RS, Khalaf HA (2022) Adoption social media- movie based learning project (SMMBL) to engage students' online environment. *Educ Administration: Theor Pract* 28(01):22–36. <https://doi.org/10.17762/kuey.v28i01.321>
18. Abd MH, Allawi OW (2022) Cheating in E-learning from the perspective of lecturers within Iraqi universities. *Wasit J Comput Math Sci* 1(4):s
19. Jong MSY, Geng J, Chai CS, Lin PY (2020) Development and predictive validity of the computational thinking disposition questionnaire. *Sustainability* 12(11):4459
20. Czerkawski B (2013) Instructional design for computational thinking. In: *Society for information technology and teacher education international conference*. Association for the Advancement of Computing in Education (AACE), pp 10–17
21. Kafura D, Bart AC, Chowdhury B (2015) Design and preliminary results from a computational thinking course. In: *Proceedings of the 2015 ACM conference on innovation and technology in computer science education*, pp 63–68
22. Rich PJ, Langton MB (2016) Computational thinking: toward a unifying definition. *Competencies Teach Learn Educ Leader Dig Age: Papers CELDA* 2014:229–242
23. Alajlan H, Alebaikan R, Almassaad A (2023) Computational thinking in K–12 computer education: appropriate pedagogy. *Technol Pedagogy Educ* 1–13
24. Standl B (2017) Solving everyday challenges in a computational way of thinking. In: *Informatics in schools: focus on learning programming: 10th international conference on informatics in schools: situation, evolution, and perspectives, ISSEP 2017, Helsinki, Finland, November 13–15, 2017, proceedings* 10. Springer International Publishing, pp 180–191
25. Hunsaker E (2020) Computational thinking. In: *The K-12 educational technology handbook*

Software Change Prediction Model Using Ensemble Learning



Sanjay Patidar, Madhvan Sharma, and Himesh Mahabi

Abstract This research study has devised an ensemble learning strategy that employs a range of deep learning and machine learning classifiers to minimize the amount of time that developers spend detecting software defects. The process of collecting data initiates this empirical study. A dataset was created using the Understand Tool and will be subject to data under-sampling techniques and outlier detection. The dataset has a binary classification output, and the count of instances of positive and negative classes will remain balanced after applying these techniques. After this, the selected classifiers will be trained using the training dataset. Next, the built model will compare envisaged outcomes generated by these classifiers with actual outcomes of training data. Classifiers that accurately anticipate the outcomes would be assigned a data point. After this, each classifier is used to train a neural network with training independent variables as input and the saved data on whether a classifier accurately predicted an outcome at a given point or not as output variables. The predicting network will determine if a classifier will accurately classify a specific data point. Now for prediction of software change of recent points in dataset, it will be provided as an input for the prediction network and the selected classifier's aggregated result that correctly predicts the outcome would be recorded.

Keywords Prediction model · Prediction network · Ensemble learning · Software change prediction model · Software maintenance

S. Patidar · M. Sharma (✉) · H. Mahabi
Department of Software Engineering, Delhi Technological University, Delhi, India
e-mail: madhvan.sharma28@gmail.com

S. Patidar
e-mail: sanjaypatidar@dtu.ac.in

1 Introduction

Testing is an essential aspect throughout the entire software development life cycle, and its significance cannot be overlooked. However, due to its criticality, it can consume significant resources, necessitating the need to explore alternative approaches to lessen the time and money required during the testing phase.

This empirical study was conducted with a primary objective of finding new ways to optimize the testing process. The study employed a unique dataset generated by the Scitools Understand Tool, in which Object-Oriented metrics were utilized to analyze the various classes in code of various applications used in Android 10 and Android 11.

The dataset posed a significant problem of data imbalance (30.18% positive outcomes), which was successfully addressed by implementing outlier detection on the overrepresented class. This approach resulted in an improved comprehension of the dataset by the model and also eliminated a significant amount of noise from the data.

Then, the dataset was further balanced using an under-sampling technique to achieve the final dataset containing 50% positive outcomes and 50% negative outcomes.

We employed six classifiers in our model.

1. Artificial Neural Network (ANN)
2. Support Vector Machine (SVM)
3. Gaussian Naive Bayes (NB)
4. Decision Tree (DT)
5. Random Forest
6. K-Nearest Neighbors (KNN).

To evaluate and contrast the effectiveness of our model, we utilize the following measurements.

1. AUC Score
2. Precision.

There are three primary stages involved in implementing the ensemble [1] learning technique.

1. Data collection and data preprocessing
2. Training of base classifier
3. Implementation of prediction network
4. Aggregation of the outcomes.

2 Data Collection

2.1 Data Acquisition

The study began with data acquisition which involved collecting the source code of various Android applications. To initiate this process, the source code was downloaded for various Android applications from the Android Open Source Project repository [2, 3]. Code was obtained for two versions of Android, namely Android Q (version 10.0) and Red Velvet Cake (version 11.0). All CPP and Java files were downloaded, while Extensible Markup Language files were excluded from the research.

2.2 Extraction of Objective Oriented Metrics

The data collection process included a highly significant step, where CPP and Java files of both Android versions underwent analysis using the Scitools Understand Tool. During this step, several Object-Oriented metrics were extracted, including the following.

1. Average cyclomatic complexity
2. Number of immediate base classes
3. Coupling between objects
4. Number of immediate subclasses
5. Number of instance methods
6. Number of instance variables
7. Number of local (not inherited) methods
8. Number of methods, including inherited ones
9. Number of local (not inherited) public methods
10. Source lines of code
11. Depth of inheritance tree
12. Lack of cohesion in methods.

The metrics are utilized as the variables or attributes for the dataset, which are not influenced by any other factors and are used to analyze the outcome.

Singh et al. [4] have thoroughly explained their rationale for choosing these particular Object-Oriented metrics, while Malhotra et al. [5] also provide sound reasoning to support their choice of metrics.

2.3 Assessment of Changes Made Between Two Versions

After analyzing the Object-Oriented metrics of both versions, any differences were compared to identify the classes that had undergone changes between the two versions. If a class had the same Object-Oriented metrics in both versions, no change was recorded. However, if a class had different Object-Oriented metrics, it was identified as having undergone changes between the two versions, and the change was recorded. Afterward, the information was saved into a CSV file, signifying the completion of the dataset.

The data collection phase is now concluded and then the dataset was processed which involved balancing of data using outlier detection and under-sampling.

2.4 Dealing with Data Imbalance

Original set of data had a disproportionate dispersal of positive (change) outcome and negative (no change) outcome classes, with 30.43% and 69.57%, respectively, resulting in poor model performance due to the presence of noise and outliers in the dataset. However, when the negative outcome class was specifically subjected to outlier detection, it resulted in a more balanced distribution of 50% positive and 50% negative outcome classes. This led to enhancement of model performance [6] and more favorable results.

3 Training of Base Classifiers

The following subsections will provide a detailed and comprehensive elaboration of the most crucial aspect of the study.

3.1 Selection of Classifiers

The initial phase of this study involved deliberate picking of deep learning and machine learning [7] classifiers, namely Artificial Neural Network, Decision Tree, Support Vector Machine, K-Nearest Neighbor, Naive Bayes, and Random Forest.

These six classifiers were chosen specifically because they represent a diverse range of machine learning techniques, each with unique learning characteristics. By incorporating a range of classifiers, the results of the study are more broadly applicable and better represent the overall performance of machine learning models.

3.2 Training Base Classifiers

To proceed with the study, the training data (xtrain and ytrain) was utilized to train the classifiers.

Next, we used the six classifiers to predict the outcome of xtrain and then compared the output with ytrain. This step was aimed to identify which classifiers accurately predicted the outcome for a given data point. A binary variable was assigned a value of 1 if the classifier was able to make a correct prediction for a particular data point; otherwise, the variable was assigned a value of 0. In the future, these matrices will be referred to as comparison matrices for easy identification and reference.

4 Implementation of Prediction Network

In the following section, we will gain insights into the structure of the prediction network, which serves the purpose of identifying the classifiers that are capable of accurately predicting the aftermath of a new data point.

The network consisted of dense layers and batch normalization layers which were attached to enhance the performance of the model and to make it more faster, stable, and less prone to overfitting. Six networks were developed for the prediction network [8], one for each of the classifiers, with the independent variable being xtrain, and the comparison matrices serving as the dependent variable. These networks predicted whether a particular data point would generate an accurate output from the corresponding classifier (Fig. 1).

The prediction network dynamically selects which classifiers would be utilized in the accumulation process. Once the prediction network is trained, we can feed testing_data (xtest) and apply the aggregation technique to the output from the selected classifiers. Now aggregation techniques will be discussed in the coming section.

5 Aggregation of the Outcome

Previous research, including studies by Verma et al. [9] and De Luca et al. [10], have utilized various aggregation methodologies such as majority voting, weighted average, Bayesian model averaging (BMA).

Throughout this research, the use of different aggregation methodologies such as voting based on median, weighted average technique, or the majority-based voting would capitulate similar results as the outcome is binary. Therefore, we chose to use majority voting as it is a frequently used technique in ensemble learning models.

Once the prediction network stated all classifiers that were likely to produce accurate results, majority voting was used on the outcomes of these classifiers.

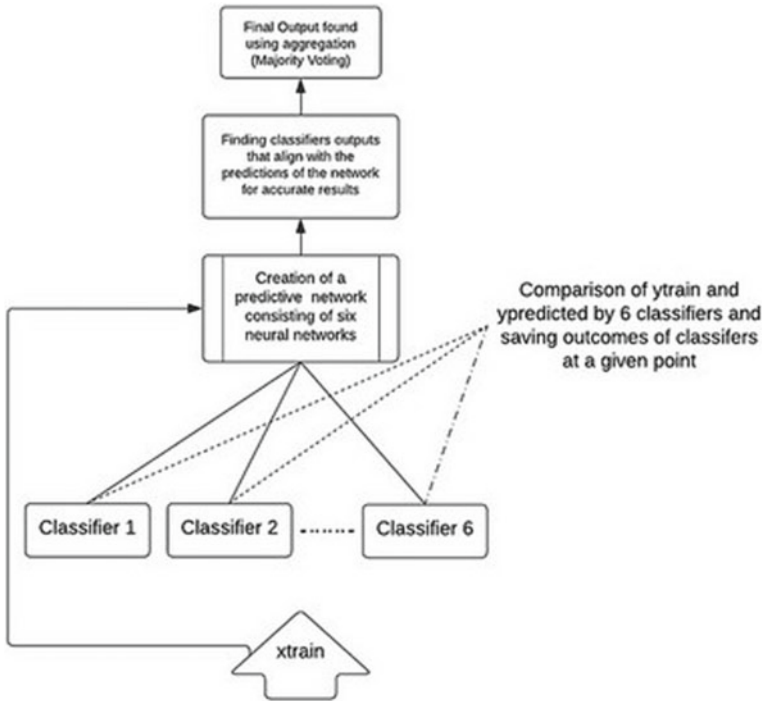


Fig. 1 Prediction network architecture of ensemble model

Majority voting is a straightforward technique where the class (change or no change) having majority votes is declared as the final outcome.

However, in cases where there is a tie between the classes, with an equal number of votes for each, resolving the tie can be challenging. The researchers [11] Margaritis et al. [12] leveraged the characteristics of the softmax function within a neural network to resolve ties or draw situations. But in this research, we directly rejected the outcome of the first recorded outcome, and thus, the other prevailing outcomes will always have a majority and break the tie.

6 Results and Analysis

Table 1 provides a clear representation of the performance of each classifier individually.

Table 1 Tabular representation of evaluation metrics of models

Classifier	AUC score	Precision
Ensemble	0.8918	0.8748
ANN	0.8376	0.7621
DT	0.8104	0.7774
Random Forest	0.8596	0.8381

6.1 AUC Score

The Receiver Operating Characteristic curve [13] is a performance metric that assesses the potential of the model to distinguish between classes. A higher AUC denotes that the model is preferable at correctly predicting positive instances as positive and negative instances as negative.

In this empirical research work, the ensemble model which was developed obtained an AUC score of 0.89, which was the highest compared to all other individual classifiers.

The Random Forest model achieved a 0.86 Area Under Curve score, which appears to be the closest any other machine learning algorithm came.

Other algorithms of machine learning could barely achieve accuracy which was close to 83%.

6.2 Precision Result

It [14] is a crucial benchmark for assessing the model performance.

It is used to estimate the proportion of true positive predictions out of all positive predictions made by the model, thus providing insight into the model's accuracy in correctly identifying positive instances. A higher precision score indicates a more precise model.

In this specific research, the model being evaluated attained an unprecedented precision score of 0.8748, thereby surpassing the Random Forest which was slightly lower by 0.0367.

The other classifiers evaluated had relatively lower precision scores, i.e., 0.7774 in case of Decision Tree (DT), whereas it was 0.8381 in case of Random Forest. This highlights the strength of the model's precision performance compared to other models (Fig. 2).

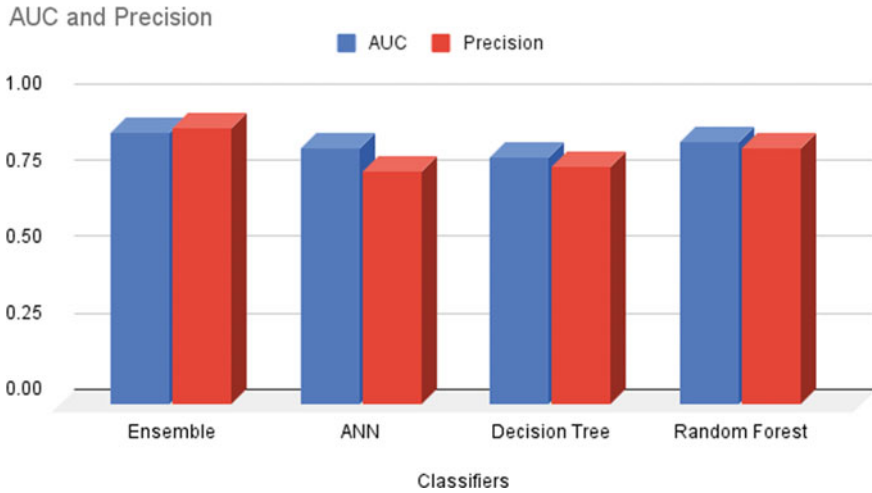


Fig. 2 Bar graph representation of comparison of evaluation metrics of ensemble and stand-alone models

7 Conclusion

We performed an empirical study aimed at developing a model to identify software that are likely to undergo changes, with the goal of reducing the time and money required during the testing phase of the software development life cycle.

Through this study, we were able to create an ensemble model that utilizes Object-Oriented metrics to accurately detect a class's proneness to change. This model proved to be superior to all other stand-alone models, as we achieved evaluation metrics that exceeded the stand-alone models used.

Additionally, our ensemble model dynamically selected the outcomes of base classifiers to be used for aggregation using majority voting, making our model unique among contemporary work in ensemble learning.

Our model's performance was assessed using precision and AUC score, which demonstrated superior performance compared to remaining discrete models.

References

1. Zhou ZH (2021) Ensemble learning. In: Machine learning springer. Singapore
2. Android Open Source Project repository. <https://github.com/aosp-mirror>
3. Google repository. <https://android.googlesource.com/platform/>
4. Aggarwal K, Singh Y, Kaur A, Malhotra R (2006) Empirical study of object-oriented metrics. *J Object Technol* 5:149–173. <https://doi.org/10.5381/jot.2006.5.8.a5>
5. Singh Y, Kaur A, Malhotra R (2010) Empirical validation of object-oriented metrics for predicting. Fault proneness models. *Software Q J* 18:3–35. <https://doi.org/10.1007/s11219-009-9079-6>

6. Yadav S, Bhole GP (2020) Handling imbalanced dataset classification in machine learning. In: 2020 IEEE Pune section international conference (PuneCon), Pune, India, 2020, pp 38–43. <https://doi.org/10.1109/PuneCon50868.2020.9362471>
7. Rahul, Kedia P, Sarangi S, Monika (2020) Analysis of machine learning models for malware detection. *J Discrete Math Sci Crypto* 23:395–407. <https://doi.org/10.1080/09720529.2020.1721870>
8. Lipnickas A, Korbicz J (2003) Adaptive selection of neural networks for a committee decision. In: Second IEEE international workshop on intelligent data acquisition and advanced computing systems: technology and applications, 2003. Proceedings, Lviv, Ukraine, 2003, pp 109–114. <https://doi.org/10.1109/IDAACS.2003.1249528>
9. Chandra TB, Verma K, Singh B, Jain D, Netam S (2020) Coronavirus Disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble. *Expert Syst Appl* 165:113909. <https://doi.org/10.1016/j.eswa.2020.113909>
10. De Luca G, Magnus J (2012) Bayesian model averaging and weighted-average least squares: equivariance, stability, and numerical issues. *The Stata J Prom Commun Stat Stata* 11:518–544. <https://doi.org/10.1177/1536867X1201100402>
11. Chaouiya C, Ourrad O, Lima R (2013) Majority rules with random tiebreaking in boolean gene regulatory networks. *PLoS ONE* 8
12. Kokkinos Y, Margaritis KG (2014) Breaking ties of plurality voting in ensembles of distributed neural network classifiers using soft max accumulations. *IFIP Adv Inf Commun Technol* 436. https://doi.org/10.1007/978-3-662-44654-6_2
13. Wu S, Flach P (2005) A scored AUC metric for classifier evaluation and selection
14. Anwyl-Irvine A, Dalmaijer ES, Hodges N et al (2021) Realistic precision and accuracy of online experiment platforms, web browsers, and devices. *Behav Res* 53:1407–1425

Discerning Monkeypox from Other Viruses of the Poxviridae Family in a Deep Learning Paradigm



Malti Bansal, Riyanshi Arora, Sai Keshari, and Sakshi Panchal

Abstract The world suffered a lot due to the Covid outbreak of 2019 which resulted into a pandemic and millions of people losing their lives and livelihoods as its repercussions. While the world was still recovering from its repercussions, the cases of monkeypox arose and were very evident in the US, Europe and Africa as well. The early detection of a disease plays a very vital role in curbing its spread. Foreseeing the Covid outbreak, in its early stages, its detection was very time-taking and hence late detection resulted in the spread of the disease. Therefore, we propose a CNN-based ensemble which exploits the feature extraction capabilities of VGG-16, MobileNet-50, Inception-V3 and ResNet-50 architectures. We thereby achieve a better ensemble accuracy of 90% using a large dataset. Along with the accuracy, we also aim at improving the recall, precision and f1-score in our ensemble learning method. We treat this problem to be a multiclass classification problem since detection of chickenpox, measles, cowpox, smallpox and healthy skin images can be often confusing and overlapping.

Keywords Accuracy · CNN · Deep learning · Detection · Ensemble · Image · Inception · Machine learning · MobileNet · Monkeypox · Skin · VGG

1 Introduction

1.1 History of Monkeypox

The emergence of the monkeypox virus as a disease in the year 2022, which was turned up by numerous countries, revealed a new threat to the world at a time when COVID-19 had already begun to have an impact. In 1990, around Central and West Africa, there were mere 50 cases of the monkeypox infection, whereas this number

M. Bansal (✉) · R. Arora · S. Keshari · S. Panchal
Department of Electronics and Communication Engineering, Delhi Technological University
(DTU), Delhi 110042, India
e-mail: maltibansal@gmail.com

had increased to as many as 5000 cases until the year 2020. Previously, it was believed that monkeypox mostly affected Africa. In 2022, even so, a large number of non-African countries in Europe as well as the US have confirmed detecting monkeypox cases [1]. There are currently 75 nations outside of Africa where there are verified occurrences of monkeypox, making it a significant global health concern. The zoonotic Orthopox virus is responsible for the monkeypox infection. It belongs to the “poxviridae” type family and has strong ties to both cowpox and smallpox. Although rodents and monkeys are still the main transmitters, cultural infection is still rather prevalent. In specific parts of Africa, which houses the majority proportion of tropical rainforests, a disease like monkeypox often affects a huge number of individuals. The transmission is primarily caused when the individual comes into a close contact [2] (be it physical or airborne through mucus, drops from respiratory tract) of any other animal, object or human counterpart which is said to be infected in the first place. Monkeypox infection was first discovered in 1958 while monkeys were under study in a laboratory at the three State Serum Institutes in Denmark, Copenhagen and South Africa [3]. It is likely that monkeypox has been present throughout sub-Saharan Africa since long before humans contracted the infection from an infected creature. After smallpox was finally eradicated in 1970, it became apparent that smallpox-like illnesses were still occurring in rural areas. This resulted in the identification of monkeypox as a separate disease. Monkeypox was revealed to be the source for a group of cases in the Midwest US in the summer of 2003. In 2003, US witnessed the initial wave of monkeypox, and it was traced back to the infection in prairie dogs which had been imported from sub-Saharan nations [4]. Since then, monkeypox has gained attention as a disease of worldwide public health concern (Fig. 1).

1.2 Need for Monkeypox Detection

Since May 2022, there have been findings of fresh cases of monkeypox virus in nations not earlier prone to the infection, with a simultaneous and consistent reporting from the list of endemic nations too. The CDC aka Center for Disease Control and Prevention had reported as many as 5783 cases as of July 1, 2022, which were documented in more than 53 countries [5]. Currently, the western hemisphere and parts of Europe are home to the majority of monkeypox cases. Many cases of monkeypox and groups of it have been documented often in both the virus-prone and non-prone countries spread across a wide variety of geographical zones. World Health Organization (WHO) conducted a study named “2022 Monkeypox Outbreak: Global Trends”, according to which it has been witnessed that there was a small (2.4%) rise in reported incidents from the week dated 24 October 24, 2022–October 20, 2022 to the one dated October 31, 2022–November 6, 2022 (week 43 to week 44) [6]. The bulk of cases (91%) documented internationally, according to the report, were in the Region of the Americas.



Fig. 1 Monkeypox skin lesion through multiple phases Reproduced from [1]

1.3 Machine Learning and AI for Monkeypox Detection and a Brief History for the Researches Using ML

It is evident that artificial intelligence (AI) along with machine learning (ML), its sub-domain, is growing in popularity across a wide spectrum of disciplines. With the special capabilities of ML, medical professionals can receive quick, accurate, and safe imaging solutions. These solutions have become widely recognized as useful decision-making aids. In 2019, Roy et al. employed various separation approaches to recognize various skin diseases to identify skin conditions [3]. In 2020, a deep learning-based study was performed using a tiny set of data consisting of COVID-19 patients (108-infected and 86-not infected), wherein 10 such distinct models were examined by Ardakani et al., the accuracy of which reached 99%, which was deemed commendable [7]. Recently, the year 2022 also witnessed a convolutional neural network (CNN)-based study, suggested by Sandeep et al. He proposed that a simpler convolutional network model can be used to diagnose melanoma, psoriasis, chickenpox and other skin-related infections and that utilizing the currently present *VGGNet*, around 71% of accuracy can be achieved through snapshot evaluation [7]. This recommended method, however, produces superior results with a typical precision of approximately 78%.

1.4 Proposed Method

We implement three different architectures of CNN, i.e., VGG-16, Inception-V3 and MobileNet-V2 after exploring various other models. Furthermore, we use hard voting technique to ensemble them for better recall, accuracy, f1-score and precision score. In the process, we also add some layers to the existing models to achieve better results in terms of accuracy. The workflow starts with image collection, preprocessing and augmentation followed by separation of dataset. Furthermore, the models are trained, their best accuracies are noted and ensemble is done using the hard voting method followed by testing and image recognition of all the six diseases of the poxviridae family. The paper presents an introduction of the history of monkeypox, need for its detection and significance of machine learning for the same. Followed by this, recent works in the field have been reviewed. Then we discuss the proposed methodology followed by dataset discussion and results that we have obtained after implementation.

2 Recent Works in the Field

As the cases of monkeypox started rising worldwide, and it became a virus variant of concern, many researchers approached toward the problem statement of detecting monkeypox using skin lesions in very different ways. However, still the literature and research in the field is quite scant. This section reviews the literature and works in the field. Reference [8] used eight models and their different architectures to classify the snapshots of skin blemishes of monkeypox, chickenpox and measles with healthy individuals. They used VGG, Inception, Xception, DenseNet, ResNet, IncepResNet and Efficient Net for solving the purpose. Then they used the best results, i.e., results of Xception and DenseNet and created an ensemble approach using voting methodology. Their best-case accuracy was that of 87.13%. They further used the Grad-CAM and LIME techniques for further explanation of the results. Recently, the year 2022 also witnessed a convolutional neural network (CNN)-based study, suggested by Sandeep et al. He proposed that a simpler convolutional network model can be used to diagnose melanoma, psoriasis, chickenpox and other skin-related infections and that utilizing the currently present *VGGNet*, around 71% of accuracy can be achieved through snapshot evaluation [7]. This recommended method, however, produces superior results with a typical precision of approximately 78%. The authors of [1] conducted two studies, where in study one, the dataset had images of chickenpox and monkeypox, while in study two the dataset had augmented images of diseases similar to monkeypox in terms of skin blemishes. They achieved a test correctness of 83 and 78% using VGG-16 architecture of CNN networks. They further used the LIME as explainable AI in order to study the true predictions of the model. Furthermore, Shams Nafisa Ali et al., Md. Tazuddin Ahmed et al. and their co-authors conducted

Table 1 Classification of original dataset

Disease	Training set	Validation set	Test set	Total
Chickenpox	348	38	45	431
Measles	147	21	31	199
Cowpox	124	19	22	165
Monkeypox	321	35	56	412
Smallpox	102	11	27	140
Healthy	98	17	41	156

a feasibility study for monkeypox detection using an application-based methodology. Apart from this, they also implemented VGG-16, ResNet-50, and Inception V3 models separately and then presented an ensemble approach of these models that gave an accuracy of 81.4%, 82.96%, 74.07% and 79.26%, respectively [9, 10]. Apart from this, authors such as Ameera S. Jaradat et al. and her co-authors, used a technology called DCGAN technology on their dataset [11]. It is basically a technique of data augmentation and it improved their accuracy score significantly when compared with others. Furthermore, their data was divided into four diseases, namely Monkeypox, normal healthy images, Scarlet fever and Roseola. They implemented it on five models, viz a viz., EfficientNet B3, VGG-16, VGG-19, MobileNetV2 and ResNet-50 [12]. Their worst-case accuracy was in the case of EfficientNet and the best-case accuracy came out to be that of MobileNetV2, which are 68.3% and 98.16%, respectively. However, the dataset used by them was quite small even after augmentation. Datasets play a very important role in determining the accuracy of the model. For a particular dataset, an accuracy of 70% can be considered as a good one, while the case might be opposite for some other dataset. Selen Gürbüz et al. and Galip Aydın et al. also did a similar kind of work but on pretrained models only [13]. They did not present a new approach for the same. Table 1 gives a comparison of the different models and techniques used by different authors.

3 Proposed Methodology

We implemented each model individually on our dataset and analyzed the outcomes determined by a number of factors such as correctness, loss, etc. After careful consideration, we found that MoileNetV2, InceptionV3 and VGG-16 are the best models for ensemble method.

3.1 Models Used

3.1.1 MobileNetV2

MobileNetV1 is convolutional neural network architecture by Google. It is majorly utilized in mobile vision and embedded vision. The rest other convolutions use standard convolution, whereas MobileNet uses convolutions that are depth-wise separable. MobileNets are majorly utilized in the Inception CNN (Fig. 2).

To be precise, it is used in the first initial layers of CNN and the purpose is to reduce computation in them. Mobile nets can prove to be useful for tasks such as object detection, landmark recognition, fine-grain classification, face attributes, etc. In MobileNetV2, we added several layers starting with a 2D convolution layer with activation function as “relu”, max pooling layer, flatten layer, dense layer having output length as 512 units, using “relu” activation function and another dense layer having output length as 6 units, using “softmax” activation function.

3.1.2 InceptionV3

The essential objective of InceptionV3 is to consume less registering power by adjusting the initiation structures from prior variants. When contrasted with VGGNet, networks of Inception have shown to be all the more genuinely compelling in wording, both how much factors the framework creates and the related expenses. While we make adjustments in such a network, it must be kept in notice that the benefits (statistical) are not lost. Because the new network’s efficiency is unclear, it is difficult to adjust an Inception network for varied use cases. Many network optimization solutions are offered in an InceptionV3 learning to alleviate the limits for simpler currently available offerings. Regularization, down-sampling, factorized CNNs and parallel processing computations are among the approaches used. In InceptionV3, we added four layers, starting with flatten layer, dense layer with output size as 1024 units, using “relu” activation function, dropout layer with frequency of 0.5 to drop nodes and dense layer with output size as 6 units, using “softmax” activation function.

3.1.3 VGG-16

In 2014, the Visual Geometry Group, commonly known as VGG, located at London produced impressive outputs in the ImageNet Challenge by developing VGG models, a kind of CNN architecture that was proposed by authors, namely Karen Simonyan and Andrew Zisserman. Their team developed VGG-16 [14, 15]. These thirteen convolutional layers make up the majority of the computational framework, five max pooling layers and three dense layers. It is referred to as VGG-16 since it has sixteen levels with adaptable value factors. Another model, namely VGG-19 is an

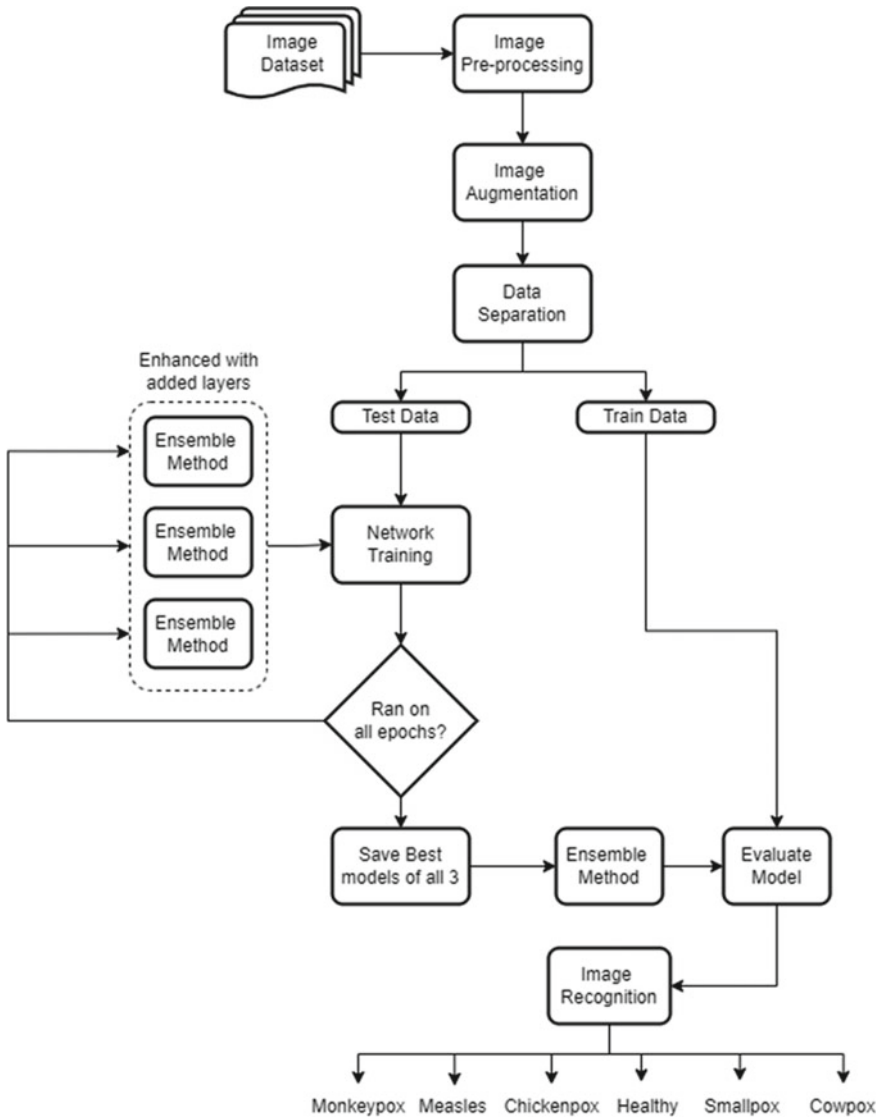


Fig. 2 Flow chart depicting the technique proposed

expanded variant of the aforementioned model, which has sixteen convolution layers, five max-pooling layers and three dense layers. In VGG-16, we added three layers, starting with flatten layer, dense layer having output length as 1024 units, using “relu” activation function and dense layer having output length as 6 units, using “softmax” activation function. The layers that follow are intended to gather characteristics, off

the source photos and categorize these. Once the model architecture has been defined, it needs to be compiled with an optimizer and a loss function.

3.2 Image Preprocessing, Augmentation and Data Separation

3.2.1 Image Gathering and Preprocessing

We have gathered a set of images from different sources explained thoroughly in data collection, after collecting a set of images for each of the following—monkeypox, smallpox, cowpox, measles, Chickenpox and healthy skin, image preprocessing was done. Preprocessing aims at strengthening the information contained in pictures by reducing unwelcome aberrations or enhancing specific visual properties that are important for subsequent computation and evaluation tasks. We used the image segmentation technique for the preprocessing. When we obtain the preprocessed image dataset for the folders of each type of outcome with number of files in them, we augment this data to generate a larger and variant set of images.

3.2.2 Image Augmentation

Using various algorithms or combinations of various preparation, including arbitrary tilt, phases, compression, turns, etc., image augmentation automatically generates pictures for training. We gently alter the initial picture to create another example, e.g., an image might be altered from its initial picture to generate a slightly more vibrant image, or the initial picture might be chopped, etc.

3.2.3 Data Separation

We select the training set and set the number of epochs for each model—VGG-16, MobileNetV2 and InceptionV3. Now, each model will run separately, and we will monitor loss and accuracy.

3.3 Basic Flow and Classification

The identical optimization and loss functions apply to all three models. The loss function gauges the extent to which the algorithm performs on the initial dataset, while the optimization tool serves to change the network's parameters during learning. We have used “Adam” optimization technique, which is a neural network model development optimization method that replaces random gradient descent. We have used “categorical_crossentropy” loss for a classification with more than one class algorithm

that includes more than two categories on the final result. One-hot class encoded data in a format of zeroes and ones are allocated to the result name. The resultant tag is transformed into categorized classification via the keras.Util to categorical technique, if it exists in numeric format. These three models also run the validation set of images simultaneously giving us accuracy and loss on both sets with every epoch. For achieving the ensemble of these three models, we picked out the best version of each model in terms of accuracy on validation set. After all epochs, we have ourselves the best versions of VGG-16, InceptionV3 and MobileNetV2 models which we can combine and produce a better ensemble model.

3.4 Ensemble

For ensemble approach, we used hard voting or the majority voting technique. Using this method, a simple majority vote determines the ensemble's ultimate conclusion. The ultimate forecast is the one with the highest likelihood that wins the majority of points out of all the ensemble models' predictions. When the categories are evenly distributed and the fundamental algorithms have comparable accuracy, hard voting is appropriate.

4 Dataset Used

We have accumulated our dataset from different sources since there is publicly verified dataset released by a major organization, comprising the images of diseases which have similar types of skin lesions. There are majorly five diseases which have similar looking skin defects which makes it hard to differentiate and correctly identify the disease, i.e., our problem statement. Those five diseases are monkeypox, cowpox, smallpox, measles and chickenpox (Fig. 3; Table 2).

5 Metrics Used for Evaluation of the Research

In order to evaluate a convolutional neural network, various metrics such as the precision, accuracy, f1-score, etc., are taken into account which would be conferred about in this section.

Accuracy is the percentage of correctly classified images. It is a basic metric and is calculated as

$$\frac{TN + TP}{TN + FP + TP + FN}$$

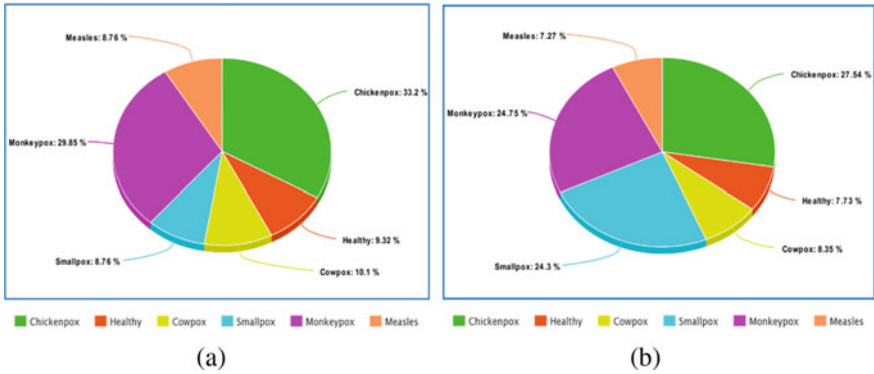


Fig. 3 Distribution of the six classes of disease/healthy images used for the classification. **a** Original dataset **b** Augmented dataset

Table 2. Classification of augmented dataset

Disease	Training set	Validation set	Test set	Total
Chickenpox	348	38	45	431
Measles	147	21	31	199
Cowpox	124	19	22	165
Monkeypox	321	35	56	412
Smallpox	102	11	27	140
Healthy	98	17	41	156

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

Precision is the percentage of accurate affirmative guesses. It measures how often the model is correct when it predicts positive. It is calculated as

$$\frac{TP}{TP + FP}$$

The recall is the ratio of genuine good forecasts to all real successes. It gauges how frequently the algorithm properly detects occurrences of positivity. It is calculated as

$$\frac{TP}{TP + FN}$$

A natural average of recall and precision is the f1-score. It provides an equilibrium among recall and precision. It is calculated as

Fig. 4 Confusion matrix used in the metrics for the proposed and implemented models

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

$$\frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Another metric is the area under the receiver operating characteristic (ROC) curve [16]. It assesses the network's capacity to differentiate among positive and negative classes [17]. Another very significant metric which plays an important role in model evaluation is the confusion matrix. The amount of accurate positive, accurate negative, accurate positive and accurate negative forecasts are displayed. It is useful to visualize how the model is performing and can help identify where the model is making errors (Fig. 4).

MSE also known as the mean squared error is a metric used to address regression issues. The mean of the proportional variations among the actual and projected values is what is measured. Then comes the MAE or the mean absolute error. Just like MSE, it is a metric used for regression problems. It calculates the mean gap among the expected and real numbers. The decision of assessment metric relies upon the main pressing concern, and it is fundamental to pick the suitable measurement to quantify the exhibition of a CNN model. Therefore, we use accuracy, precision, f1-score and recall for our use case and purpose. We also use confusion matrices and draw them for every model that we focus on while carrying out this research work.

6 Results

We carried out this research on a large, augmented dataset. We implemented four models: ResNet 50, VGG-16, InceptionV3 and MobileNetV2. The graphs shown in this section are depicting the validation accuracies and training accuracies for all the models implemented in our use case.

Figures 5 and 6 depict the training and validation accuracy of VGG-16 model as well as the training and validation loss for the same [18]. Logically, with increasing numbers of epochs, the accuracy should increase for training and validation and the loss should decrease, which is fairly evident in the resultant graphs depicted.

Fig. 5 Training and validation accuracy for VGG-16 model (x-axis: no. of epochs, y-axis: accuracy)

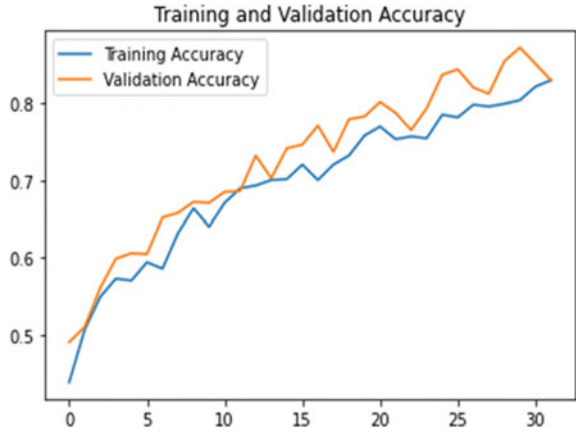
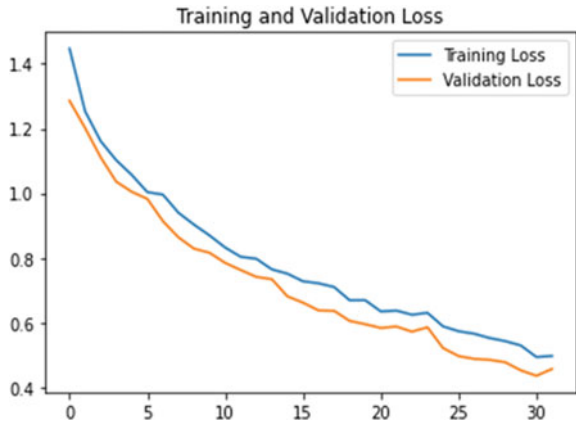


Fig. 6 Training and validation loss for VGG-16 model (x-axis: no. of epochs, y-axis: loss)



Figures 7 and 8 portray the preparation and approval accuracy of Inception model as well as the preparation and approval misfortune for the equivalent. Legitimately, with expanding quantities of ages, the accuracy ought to increment for preparing and approval and the misfortune ought to diminish, which is genuinely apparent in the resultant diagrams displayed beneath. Figures 9 and 10 illustrate the MobileNetV2 model’s training and validation accuracy in addition to its training and validation loss. Logically, as the number of epochs increases, the accuracy for training and validation should improve, while the loss should go down, as illustrated in the Figs. 9 and 10.

Furthermore, Table 3 gives the accuracy, precision, recall and f1-score in terms of percentage of the models implemented by us and the table also compares the pretrained models to the ensemble approach used by us.

Usually, researchers aim at improving only the correctness of the projected model when matched with pretrained models. [1, 19–22] But we aim at improving all the four metrics, i.e., the accuracy of the model/approach, its precision, recall as well as

Fig. 7 Training and validation accuracy for InceptionV3 model (x-axis: no. of epochs, y-axis: accuracy)

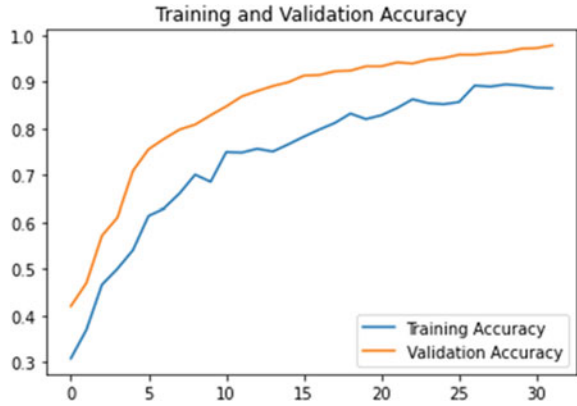


Fig. 8 Training and validation loss for InceptionV3 model (x-axis: no. of epochs, y-axis: loss)



Fig. 9 Training and validation accuracy for MobileNetV2 model (x-axis: no. of epochs, y-axis: accuracy)

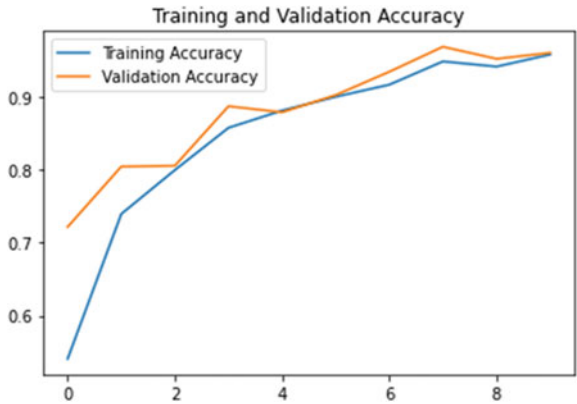


Fig. 10 Training and validation loss for MobileNetV2 model (x-axis: no. of epochs, y-axis: loss)



Table 3 Metric table of implemented models

Model	Metrics			
	Accuracy	Precision	Recall	F1-score
VGG-16	81	86	80	81
MobileNetV2	85	92	75	81
InceptionV3	85	92	80	85
Ensemble approach VGG-Inc-Mobile	90	95	85	89

the f1-score. Our ensemble approach of combining the pretrained models performed better than other three models implemented. The accuracy of the proposed ensemble is even better than the accuracy of the ensemble of some other models suggested/researched upon yet. Majority voting ensemble method is used by us in the paper. In the code, the predictions of four different models (VGG16, MobileNetV2, Inception and benchmark model) are combined using the mode function, which returns the most frequently occurring predicted class among the four models for each image. If the four models predict different classes, then MobileNet’s prediction is given high priority by using the mode function, thus giving it an advantage in the final prediction [17, 18, 23–25]. Therefore, the final prediction for each image is based on the most frequent prediction among the four models or MobileNet’s prediction if they all predict something different (Fig. 11).

Figures 12, 13, 14 and 15 are the generated confusion matrices for each of the models used by us along with that for the ensemble approach.

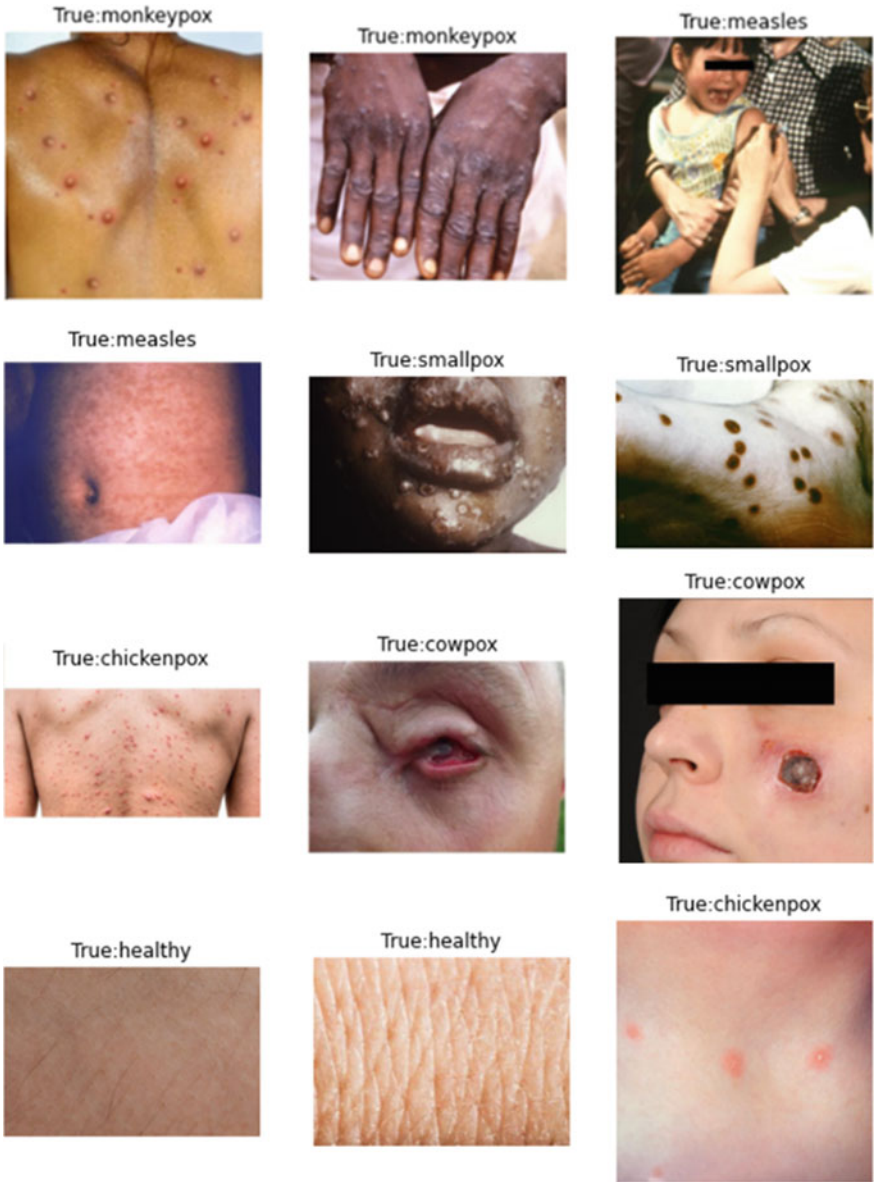


Fig. 11 Few correctly classified images from the wide range of our dataset

Fig. 12 Confusion matrix generated for MobileNetV2

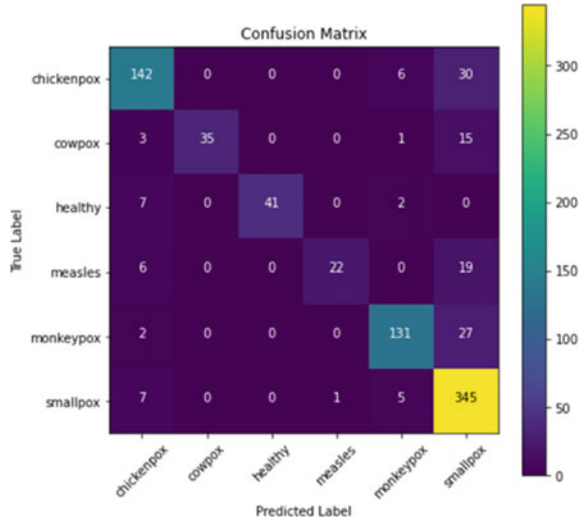


Fig. 13 Confusion matrix generated for VGG-16

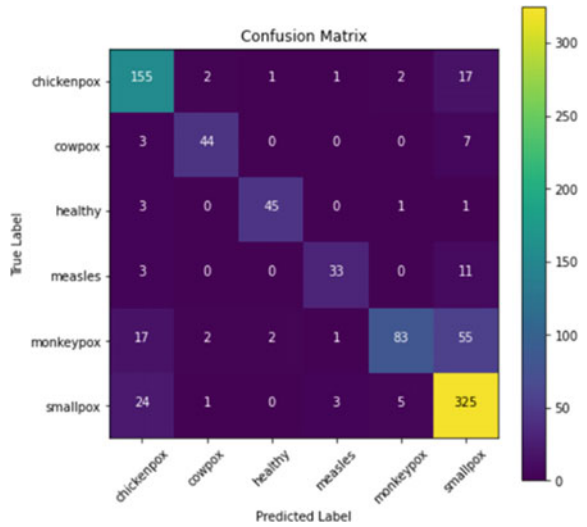


Fig. 14 Confusion matrix generated for InceptionV3

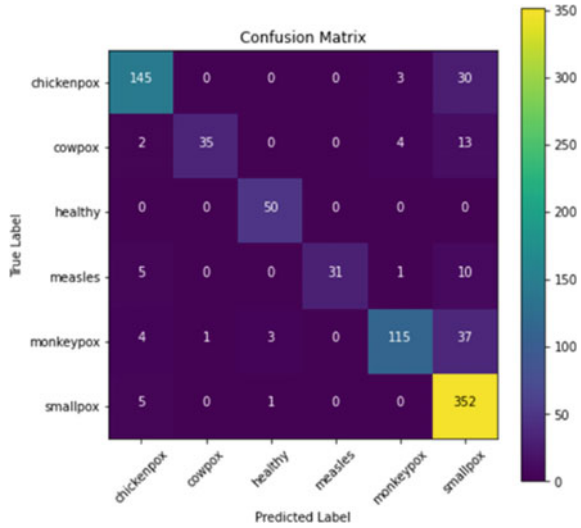
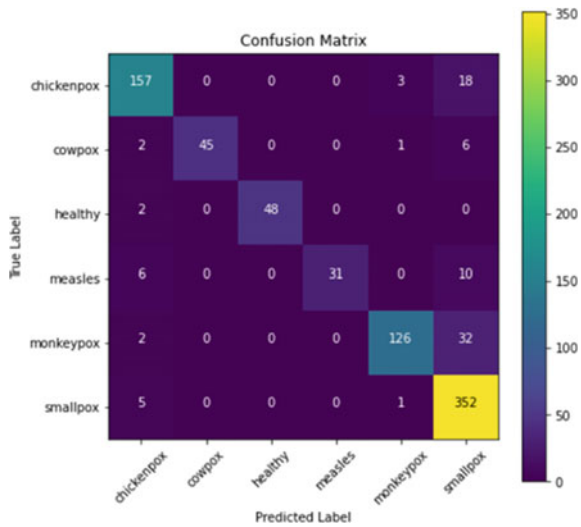


Fig. 15 Confusion matrix generated for ensemble approach



7 Conclusion

We firstly assessed the research made in the field of virology for diagnosis of the virus covered under the introduction section of the paper. We explored a new dataset and used image augmentation techniques in order to expand the dataset. We also explored different pretrained models and achieved a better accuracy than the previous works done. In this paper, we initially implement VGG-16, MobileNetV2 and InceptionV3. Then we propose an approach for achieving better accuracy than those three models

we have discussed in the paper. We use majority/hard voting technique for ensemble, and we give highest priority to MobileNetV2 model, followed by InceptionV3 and VGG-16. We achieve an accuracy better than that accuracy achieved by all three models separately and the accuracy is even better than most methods proposed in the past. We have achieved an ensemble method accuracy of 90% approximately and along with the accuracy, we have also taken into consideration, the improvement of precision, recall and f1-score [16, 26–32]. Predicting outbreaks of infectious diseases like monkeypox can be challenging, but deep learning techniques can provide a promising approach to improve accuracy and speed of prediction. Validating the model's performance on a range of datasets, including cross-validation and out-of-sample testing can help ensure its accuracy and generalizability. Model optimization techniques, such as hyperparameter tuning can improve model performance and reduce overfitting. Furthermore, there is also a scope of development of new models and applications to detect diseases of the poxviridae family based on skin lesions with more accuracy as the diseases caused by these viruses are mostly contagious and can result in further outbreaks and pandemics if not detected in early stages.

References

1. Ahsan MM, Uddin MR, Farjana M, Sakib A, Momin K, Luna Shahana (2022) Image data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16. <https://doi.org/10.48550/arXiv.2206.01862>
2. Moore M, Zahra F (2022) Monkeypox. <https://www.ncbi.nlm.nih.gov/books/NBK574519/>
3. McCollum AM, Damon IK (2014) Human monkeypox. *Clin Infect Dis* 58:260–267. <https://doi.org/10.1093/cid/cit703>
4. Okyay R, Bayrak E, Kaya E, Sahin A, Koçyiğit B, Tasdogan A, Avci A, Sumbul H (2022) Another epidemic in the shadow of Covid 19 pandemic: a review of Monkeypox. *Eurasian J Med Oncol* 6:95–99. <https://doi.org/10.14744/ejmo.2022.2022>
5. Kaler J, Hussain A, Flores G et al, Monkeypox: a comprehensive review of transmission, pathogenesis, and manifestation. *Cureus* 14(7):e26531. <https://doi.org/10.7759/cureus.26531>
6. 2022 Global Map & Case Count (2022) <https://www.cdc.gov/poxvirus/monkeypox/response/2022/world-map.html>
7. Ardakani AB, Kanafi AR, Rajendra Acharya U, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks. *Comput Biol Med* 121:103795. ISSN 0010-4825. <https://doi.org/10.1016/j.combiomed.2020.103795>
8. Mangena V, Khamparia A, Gupta D, Pande S, Tiwari P, Hossain MS (2021) Res-CovNet: an internet of medical health things driven COVID-19 framework using transfer learning. *Neural Comput Appl*
9. Sitaula C, Basnet A, Mainali A, Shahi T (2021) Deep learning-based methods for sentiment analysis on Nepali COVID-19-related tweets. *Comput Intell Neurosci*. <https://doi.org/10.1155/2021/2158184>
10. Glock K, Napier C, Gary T, Gupta V, Gigante J, Schaffner W, Wang Q (2021) Measles rash identification using transfer learning and deep convolutional neural networks, 3905–3910. <https://doi.org/10.1109/BigData52589.2021.9671333>
11. Tan M, Le Quoc (2019) EfficientNet: rethinking model scaling for convolutional neural networks

12. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. *IEEE Int Conf Comput Vis (ICCV)* 2017:618–626. <https://doi.org/10.1109/ICCV.2017.74>
13. Ribeiro MT, Singh S, Guestrin C (2016) Why should I trust you?: explaining the predictions of any classifier. In: *Proceedings of 78 Page 8 of 9 journal of medical systems* (2022) 46:78 1 3the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13–17, 2016, pp 1135–1144
14. Abdelhamid A, El-kenawy E-S, Khodadadi N, Mirjalili S, Khafaga D, Alharbi A, Ibrahim A, Eid M, Saber M (2022) Classification of Monkeypox images based on transfer learning and the AI-Biruni earth radius optimization algorithm 10:3614. <https://doi.org/10.3390/math10193614>
15. Ali SN, Ahmed T, Paul J, Jahan T, Sani SM, Noor N, Hasan T (2022) Monkeypox skin lesion detection using deep learning models: a feasibility study. <https://doi.org/10.48550/arXiv.2207.03342>
16. Reyna MA, Kiarashi Y, Elola A, Oliveira J, Renna F, Gu A, Perez Alday EA, Sadr N, Sharma A, Kpodonu J, Mattos S, Coimbra MT, Sameni R, Rad AB, Clifford GD (2022) Heart murmur detection from phonocardiogram recordings: the George B. moody physionet challenge 2022. <https://doi.org/10.1101/2022.08.11.22278688>
17. *Machine Learning and Knowledge Discovery in Databases* (2021) Lecture notes in computer science. <https://doi.org/10.1007/978-3-030-67658-2>
18. Nivetha MN, Moulya MC, Parveen MA, Shree MR, Ragupathy MS (2020) Blood content prediction using deep learning techniques. *Int J Innov Technol Explor Eng* 9(6):308–313. <https://doi.org/10.35940/ijitee.f3067.049620>
19. Mohan S, Thirumalai CS, Srivastava G (2019) Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access* 1–1. <https://doi.org/10.1109/ACCESS.2019.2923707>
20. Qin J, Chen L, Liu Y, Liu C, Feng C, Chen B (2020) A machine learning methodology for diagnosing chronic kidney disease. *IEEE Access* 8:20991–21002. <https://doi.org/10.1109/ACCESS.2019.2963053>
21. El-kenawy, E-S, Mirjalili S, Alassery F, Zhang Y-D, Eid M, El-Mashad S, Aloyaydi B, Ibrahim A, Abdelhamid A (2022) Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems. *IEEE Access* 1–1. <https://doi.org/10.1109/ACCESS.2022.3166901>
22. Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F (2009) ImageNet: a large-scale hierarchical image database. In: *IEEE conference on computer vision and pattern recognition*, pp 248–255
23. Chollet F (2017) Xception: deep learning with depth wise separable convolutions. 1800–1807. <https://doi.org/10.1109/CVPR.2017.195>
24. Dwivedi M, Tiwari RG, Ujjwal N (2022) Deep learning methods for early detection of Monkeypox skin lesion. In: *2022 8th international conference on signal processing and communication (ICSC)*, Noida, India, 2022, pp 343–348. <https://doi.org/10.1109/ICSC56524.2022.10009571>
25. Haque ME, Ahmed MR, Nila RS, Islam S (2022) Human Monkeypox disease detection using deep learning and attention mechanisms. In: *2022 25th international conference on computer and information technology (ICCIT)*, Cox's Bazar, Bangladesh, 2022, pp 1069–1073. <https://doi.org/10.1109/ICCIT57492.2022.10>
26. Shahyeez Ahamed B, Usha R, Sreenivasulu G (2022) A deep learning-based methodology for predicting monkey pox from skin sores. In: *2022 IEEE 2nd Mysore sub section international conference (MysuruCon)*, Mysuru, India, 2022, pp 1–6. <https://doi.org/10.1109/MysuruCon55714.2022.9972746>
27. Sharma S, Parmar M (2020) Heart diseases prediction using deep learning neural network model. *Int J Innov Technol Explor Eng* 9(3):2244–2248. <https://doi.org/10.35940/ijitee.c9009.019320>
28. Bansal M, Goyal A, Choudhary A (2022) A comparative analysis of K-nearest neighbour, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decis Anal J* 3:100071. ISSN 2772-6622. <https://doi.org/10.1016/j.dajour.2022.100071>, <https://www.sciencedirect.com/science/article/pii/S2772662222000261>

29. Bansal M, Priya (2020) Application layer protocols for internet of healthcare things (IoHT). In: 2020 Fourth international conference on inventive systems and control (ICISC), Coimbatore, India, 2020, pp 369–376. <https://doi.org/10.1109/ICISC47916.2020.9171092>
30. Bansal M, Sirpal V (2021) Fog computing-based internet of things and its applications in healthcare. *J Phys Conf Ser* 1916 012041
31. Bansal MP (2021) Performance comparison of MQTT and CoAP protocols in different simulation environments. In: Ranganathan G, Chen J, Rocha Á (eds) *Inventive communication and computational technologies*. Lecture notes in networks and systems, vol 145. Springer, Singapore. https://doi.org/10.1007/978-981-15-7345-3_47
32. Bansal MP (2022) Machine learning perspective in VLSI computer-aided design at different abstraction levels. In: Shakya S, Bestak R, Palanisamy R, Kamel KA (eds) *Mobile computing and sustainable informatics*. Lecture notes on data engineering and communications technologies, vol 68. Springer, Singapore. https://doi.org/10.1007/978-981-16-1866-6_6

An Exploratory Study to Classify Brain Tumor Using Convolutional Neural Networks



Manmeet Singh, Manav Misra, Jayesh Jain, Mayank Goel,
and Kumud Kundu

Abstract Brain tumors are created when unusual cells develop within the brain. These tumors can be divided into four groups, and some can be surgically removed while others gradually spread to neighboring tissues. In this study, we explore the classification of brain tumors using a convolutional neural network (CNN) and the concept of transfer learning. We also attempted to deploy the proposed model on a web application.

Keywords Brain tumor · Magnetic resonance imaging · Convolutional neural network · Machine learning

1 Introduction

1.1 Overview

Medical image processing has become an increasingly important area of research in recent years, with brain tumor classification being one of the most critical tasks. However, the manual analysis of multiple magnetic resonance imaging (MRI) pictures produced in a clinic can be a challenging and time-consuming operation, especially when dealing with large amounts of data. The extraction of tumor areas from MRI images can be challenging due to the visual variety and similarities between brain tumors and normal tissues that characterize brain cancers.

1.2 Motivation

Automated brain tumor classification using MRI scans can potentially improve the accuracy and efficiency of tumor diagnosis, leading to better patient outcomes.

M. Singh (✉) · M. Misra · J. Jain · M. Goel · K. Kundu
Inderprastha Engineering College, Sahibabad, Uttar Pradesh, India

Machine learning techniques can assist in this process by automatically identifying patterns in medical images that may not be easily detected by human analysis. This motivates the development and evaluation of machine learning models for brain tumor classification.

1.3 Organization of the Paper

The remainder of the paper is organized as follows: Sect. 2 introduces the research topic and provides a brief overview of the problem. Section 3 provides a comprehensive review of related work on brain tumor classification using CNN. Section 4 describes the methodology used in our study, including the dataset and preprocessing techniques applied, as well as an overview of the machine learning models used. Section 5 presents the results of our experiments and the performance of each model. Section 6 presents the comparative study with existing models. In Sect. 7, we summarize our main contributions and the implications of our study for the fields of medical image processing and brain tumor classification.

1.4 Main Contributions

In this research paper, we present a study of various machine learning models for brain tumor classification using MRI scans. Our contributions include the following:

- We experiment with several pretrained CNN architectures, including DenseNet121, InceptionV3, MobileNet, ResNet50V2, VGG16, and VGG19 to evaluate CNN models for brain tumor classification.
- We also explore the use of other machine learning models, including SVM, random forest, and gradient boosting, implemented using the Scikit-learn library.
- We containerize the web application using Docker, simplifying distribution and deployment and enabling easy use of the model without additional dependencies or configurations.

Our study provides insights into the effectiveness of various machine learning models for brain tumor classification using MRI scans. These insights can inform the development of automated diagnosis tools for critical tasks. The containerization of the web application using Docker also enhances the accessibility and usability of our model.

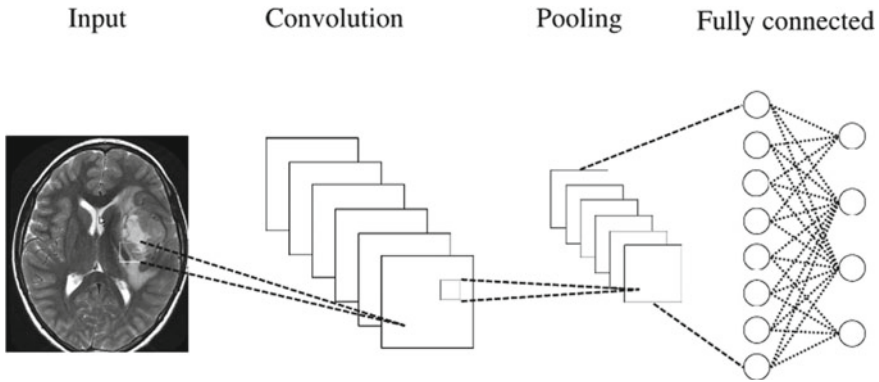


Fig. 1 CNN architecture

2 Background Study

2.1 Convolutional Neural Network

A neural network mimics the functioning of neurons in the human brain. Traditional neural networks process images pixel-by-pixel in low resolution, which is not ideal for image processing. Convolutional neural networks have neurons setup more like the frontal lobe, which in humans and other animals is where visual stimuli are processed. This arrangement allows the layers of neurons to cover the entire visual field, avoiding the problem of processing images piece by piece as in traditional neural networks.

A convolutional neural network (CNN) is a specialized type of artificial neural network designed for processing pixel data and is a powerful tool for image processing. This network processes pixel data using a mathematical operation called convolution, which is a specialized form of linear operation. Unlike traditional neural networks that use standard matrix multiplication in all layers, CNNs employ convolution in at least one of their layers (Fig. 1).

2.2 Brain Tumor

Brain tumors are abnormal growths that can form in the brain's tissues or spread to the brain from other parts of the body. They can cause a variety of symptoms, such as headaches, seizures, vision problems, memory loss, mood changes, and difficulty with movement or coordination. Early detection can lead to more successful treatment outcomes and a higher likelihood of survival.

Treatment options for brain tumors vary depending on the type, size, and location of the tumor. Common treatments include surgery, radiation therapy, and

chemotherapy, often used in combination. Patients may also benefit from supportive therapies, such as physical therapy, speech therapy, and occupational therapy to manage symptoms and improve their quality of life. Increasing awareness about brain tumors and their symptoms, as well as the importance of early detection, can help save more lives. Providing effective treatments is also crucial in improving the quality of life for those affected by this condition.

2.3 Magnetic Resonance Imaging

MRI is a powerful tool for identifying brain tumors due to its ability to visualize the effects of tissues inside the human body without the interference of bone artifacts. Various MRI sequences can target different diseased areas, and combining multiple imaging modalities enables a comprehensive assessment of the tumor's size, shape, number, and location.

Additionally, MRI can help distinguish peritumoral edema from cerebrospinal fluid and facilitate visualization of the high spinal tumor core. It also provides valuable information about the degree of surrounding cerebral edema, ventricular compression, and displacement of brain tissue. Overall, MRI is a highly beneficial technique for identifying neurological illnesses.

In clinical practice, MRI is commonly used in combination with other diagnostic tools to accurately diagnose and stage brain tumors. This allows healthcare professionals to determine the most appropriate treatment plan for each patient. The non-invasive nature of MRI makes it a safer and less stressful option than other diagnostic techniques, such as biopsy or surgery. As technology advances, MRI continues to evolve, with newer techniques such as diffusion tensor imaging (DTI) and functional MRI (fMRI) allowing for more precise and detailed imaging of the brain.

3 Literature Survey

Our proposed methodology is based on a literature survey that we concluded using various search engines and publication websites. We used keywords such as “brain tumor”, “MRI”, “CNN models”, and “tumor detection” to initiate our search.

The use of machine learning and artificial intelligence has been a popular approach for diagnosing brain tumors from MRI scans, as demonstrated in previous studies with CNN models and specific segmentation strategies, resulting in over 90% accuracy [1–5].

In contrast to models like 3D segmentation method (3DSM), shape constrained automatic segmentation (SCAS), support vector machine (SVM), and wavelet-based texture classification (WBTC), which have been used for brain tumor detection, CNN-based tumor detection models consistently achieve higher accuracy [6].

Image classification, segmentation, and selection are the critical criteria for any model to be successful. Studies have shown that uncropped images provide better results compared with cropped and segmented lesions [7].

The treatment of brain tumors highly depends on the knowledge and experience of the physician. Hence, the development of automated brain tumor detection is vital to assist radiologists and surgeons. The ELM-LRF method used in the study is efficient, has a short training period, and achieves a high classification accuracy of 97.18%. The study suggests that the proposed method is effective in computer-aided brain tumor detection, and the ELM-LRF structure is an essential tool for biomedical image processing applications [8].

Compared with artificial neural network (ANN), deep neural network (DNN), SVM, and random forest classifiers, CNN models have been consistently shown to achieve much higher accuracy in numerous studies [9, 10].

Optimizing hyperparameters for training CNN layers using techniques like nonlinear Levy Chaotic Moth Flame optimizer (NLCMFO) has been shown to achieve 97.4% [11]. In addition, using the ensemble learning model and VGG16, which integrates CNN, can boost accuracy to 98.15 and 98.5%, respectively [12].

In addition to previous studies, [13] proposed an efficient deep learning-based system using CNN and salp swarm algorithm (SSA). It achieved 99.1% accuracy for brain tumor detection on the Kaggle brain tumor dataset.

A novel parallel deep convolutional neural network (PDCNN) topology [14] is proposed for brain tumor classification, achieving high accuracy on three different MRI datasets by extracting both global and local features.

The adaptive Gabor convolutional neural network (AG-CNN) model demonstrated high accuracy and low computational cost for brain tumor classification in MRI images [15], making it a promising approach for real-world clinical settings.

Several classification techniques have been explored, including EfficientNetB1, which achieved the best results with a training and validation accuracy of 87.67% and 89.55%, respectively [16]. Other models like VGG16 and VGG19 have also shown competitive accuracy [17, 18].

Overall, CNN models with optimized hyperparameters and appropriate image selection and segmentation techniques have shown promising results for brain tumor detection from MRI scans using machine learning and artificial intelligence. The studies reviewed in this literature survey demonstrate that deep learning-based systems using CNN and other advanced techniques have great potential to improve the accuracy of brain tumor diagnosis, which can have significant implications for clinical practice and patient outcomes.

4 Proposed Methodology

In this project, we have two main objectives, the first half deals with the classification of brain tumors, and in the second half the web application is containerized using Docker (Fig. 2).

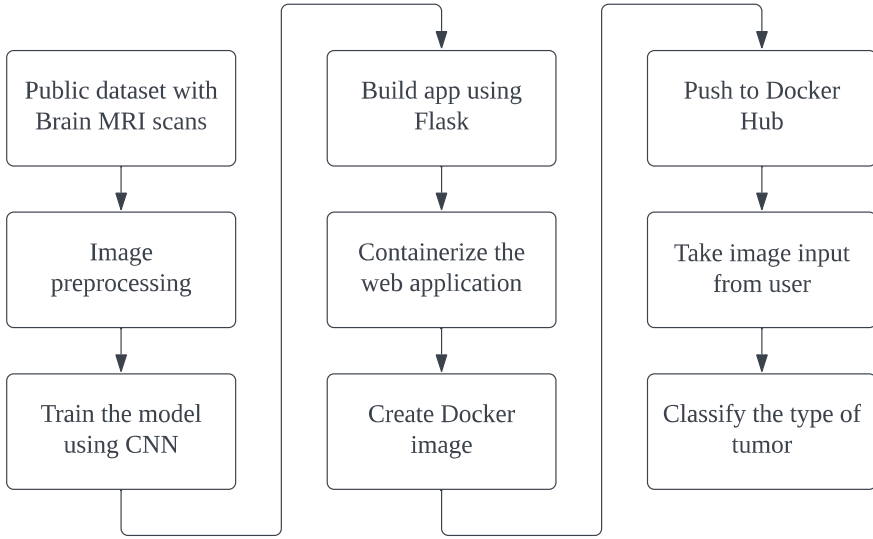


Fig. 2 Flowchart of the proposed methodology

4.1 Dataset

The dataset, which includes brain MRIs, was obtained from Kaggle [19]. It contains a total of 3264 jpg-formatted images, divided into training and testing folders, each containing four sub-folders: glioma tumors, meningioma tumors, pituitary tumors, and no tumors.

4.2 Image Preprocessing

During the preprocessing stage, the images were converted to grayscale and resized to 224×224 . Data augmentation techniques were applied to increase the size of the dataset, including horizontal and vertical flipping.

4.3 Model Training

A CNN model was built to classify the images into four categories: gliomas, meningiomas, pituitary tumors, and no tumors. The model incorporated convolution, pooling, flattening, and dense layers. To train the model, several pretrained architectures such as DenseNet121, InceptionV3, MobileNet, ResNet50V2, VGG16, and VGG19 were used. In addition, classical machine learning models like support vector

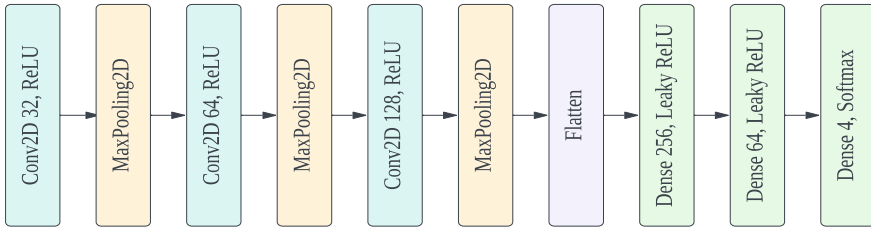


Fig.3 CNN architecture used for training

machine, random forest, and gradient boosting were employed to assess the accuracy of the model (Fig. 3).

4.4 Model Deployment

Containerizing the web application using Docker not only simplifies the distribution and deployment of the application, but also enables anyone to easily use the model by downloading the Docker image and running it on their local system or server without having to worry about installing any additional dependencies or configurations.

5 Results

To perform the analysis, we created Python code using various libraries including OS, NumPy, Pandas, Keras, and TensorFlow. Table 1 summarizes the hyperparameter values used to train our CNN model. We set the initial learning rate to 0.001, and if there was no improvement in the validation accuracy after four epochs, we reduced the learning rate by a factor of 2. Categorical cross entropy was used as the loss function. Additionally, we established a minimum learning rate of 0.0001 to prevent it from becoming excessively small. A diminished learning rate can lead to slow convergence and longer training times.

After training our CNN architecture, we achieved a remarkable training accuracy rate of 99.65%. This result suggests that our model has effectively learned from the training data and can accurately predict the presence of brain tumors in MRI images.

Table 1 Hyperparameters values

Hyperparameter	Value
Epochs	10
Batch size	32
Image dimensions	224, 224

However, we observed that classical machine learning models outperformed the CNNs during testing. Our support vector machine (SVM) model achieved a validation accuracy rate of 87.14%.

6 Comparative Study

In addition to comparing our CNN model with other pretrained models such as DenseNet121, InceptionV3, MobileNet, ResNet50V2, VGG16, and VGG19, we also evaluated the performance of our model against three classical machine learning algorithms: support vector machine, random forest classifier, and gradient boosting classifier. Figure 4 displays these results (Fig. 5).

Our proposed CNN model outperformed the existing models in the comparative analysis. When utilizing transfer learning on the VGG16 architecture for brain tumor detection, our model achieved an accuracy of 99%, surpassing the reported accuracy of 98.5% [12]. This indicates the promising performance of both models, with our model demonstrating a slight advantage in terms of higher accuracy. The improved performance can be attributed to our model's effective capture of subtle nuances and patterns in the data (Fig. 6).

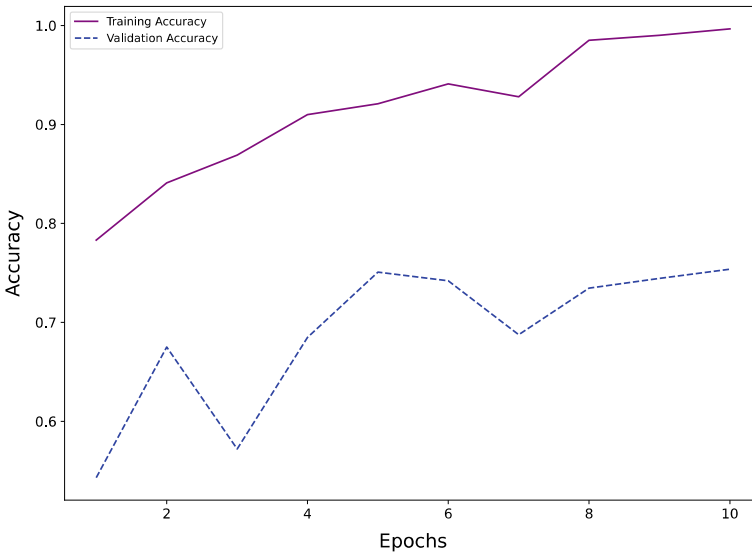


Fig. 4 Training and validation accuracy of the proposed CNN model

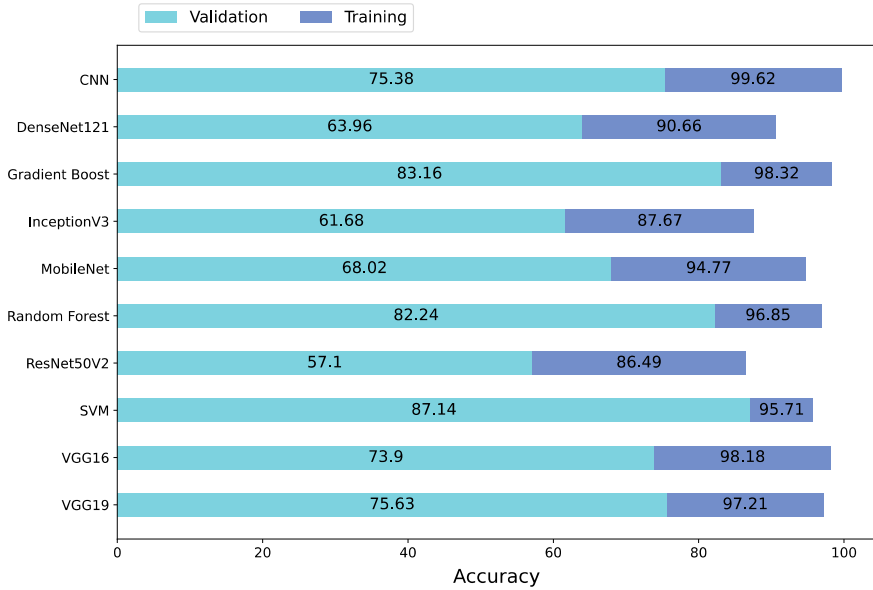


Fig. 5 Performance analysis of CNN model and other pretrained models

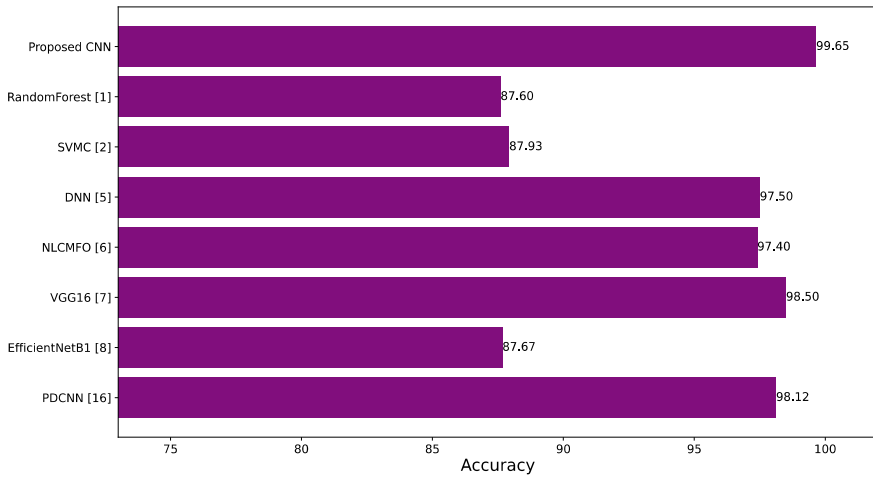


Fig. 6 Comparative performance analysis with existing models

7 Conclusion

In conclusion, this study proposed the use of convolutional neural networks for detecting brain tumors from MRI images. The optimized CNN architecture achieved a remarkable training accuracy rate of 99.65%, demonstrating its ability to accurately

predict the presence of brain tumors. However, limitations of the study include the limited size and scope of the dataset, which may impact the generalizability of the results. The study also did not consider additional factors such as age, gender, and medical history of patients, which could influence tumor classification.

While classical machine learning models, particularly the SVM, outperformed the CNNs during testing, the choice of the machine learning algorithm depends on the specific problem and dataset. Exploring different models and comparing their performances are beneficial in selecting the most appropriate one.

Overall, the findings highlight the effectiveness of convolutional neural networks in detecting brain tumors from MRI images. Future research should address the study's limitations by incorporating larger and more diverse datasets and considering additional patient factors.

We have added Docker to our web application for easier containerization and improved portability. The web application can be accessed by a few commands. Docker eliminates the need for extra configuration, making deployment hassle-free. It also isolates the application for enhanced security and reduces conflicts with other software. Anyone can access the web application by executing these commands.

```
docker pull manavmisra2/ml_ops
docker run -it -name tumor_detector -p 8888:8888 manavmisra2/ml_ops
```

References

1. Lamrani D, Cherradi B, El Gannour O, Bouqentar MA, Bahatti L (2022) Brain tumor detection using MRI images and convolutional neural network. *Int J Adv Comput Sci Appl* 13(7). <https://doi.org/10.14569/IJACSA.2022.0130755>
2. Singh P, Diwedi AK (2022) Convolutional neural network system for brain tumor detection. *Dogo Rangsang Res J* 9(1)
3. Vankdoth R, Hameed MA (2022) Brain tumor MRI images identification and classification based on the recurrent convolutional neural network
4. Khaled Abd-Ellah M, Ismail Awad A, Khalaf AAM, Hamed HFA (2019) A review on brain tumor diagnosis from MRI images: practical implications, key achievements, and lessons learned. *Magn Reson Imaging*. <https://doi.org/10.1016/j.mri.2019.05.028>
5. Pereira S, Pinto A, Alves V, Silva CA (2016) Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans Med Imaging* 35(5):1240–1251. <https://doi.org/10.1109/TMI.2016.2538465>
6. Kiruthiga T (2022) Convolution neural network based brain tumor detection using efficient classification technique. *ICTACT J Data Sci Mach Learn* 3(2)
7. Alquraan H, Alqudah AM, Abu Qasmieh I, Alqudah A, Al-Sharu W (2019) Brain tumor classification using deep learning technique—a comparison between cropped, uncropped, and segmented lesion images with different sizes. *Int J Adv Trends. Comput Sci and Eng* 8(6)
8. Ari A, Hanbay D (2018) Deep learning-based brain tumor classification and detection system. *Turkish J Electric Eng Comput Sci* 26(5). <https://doi.org/10.3906/elk-1801-8>
9. Bindu JH, Nikhil K, Rao KR, Viswas Y, Charan PS (2022) Computer aided diagnosis for brain tumor detection using VGG-16 and CNN models. *Gradiva Rev J* 8(7)
10. Seetha J, Raja SS (2018) Brain tumor classification using convolutional neural networks. *Biomed Pharmacol J* 11(3). <https://doi.org/10.13005/bpj/1511>

11. Dehkordi AA, Hashemi M, Neshrat M, Mirjalili S, Sadiq AS (2022) Brain tumor detection and classification using a new evolutionary convolutional neural network. <https://doi.org/10.48550/arXiv.2204.12297>
12. Younis A, Qiang L, Nyatega CO, Adamu MJ, Kawuwa HB (2022) Brain tumor analysis using deep learning and VGG-16 ensembling learning approaches. *J Appl Sci* 12(14). <https://doi.org/10.3390/app12147282>
13. Rahman T, Islam MS (2023) MRI brain tumor detection and classification using parallel deep convolutional neural networks. *Measur Sens* 26. <https://doi.org/10.1016/j.measen.2023.100694>
14. Saurav S, Sharma A, Saini R, Singh S (2023) An attention-guided convolutional neural network for automated classification of brain tumor from MRI. *Neural Comput Appl* 35:2541–2560. <https://doi.org/10.1007/s00521-022-07742-z>
15. Alyami J, Rehman A, Almutairi F, Fayyaz AM, Roy S, Saba T, Alkhourim A (2023) Tumor localization and classification from MRI of brain using deep convolutional neural network and salp swarm algorithm. *Cogn Comput*. <https://doi.org/10.1007/s12559-022-10096-2>
16. Filatov D, Yar GN (2022) Brain tumor diagnosis and classification via pre-trained convolutional neural networks. <https://doi.org/10.1101/2022.07.18.22277779>
17. Alsaif H, Guesmi R, Alshammari BM, Hamrouni T, Guesmi T, Alzamil A, Belguesmi L (2022) A novel data augmentation-based brain tumor detection using convolutional neural network. *Appl Sci* 12(8)
18. Ul Hoque S (2022) Performance comparison between VGG16 and VGG19 deep learning method with CNN for brain tumor detection. *Int Res J Mod Eng Technol Sci* 4(7)
19. Bhuvaji S, Kadam A, Bhumkar P, Dedge S, Kanchan S (2020) Brain tumor classification (MRI). *Kaggle*. <https://doi.org/10.34740/KAGGLE/DSV/1183165>

Skin Cancer Detection with Edge Devices Using YOLOv7 Deep CNN



Dhruba Datta , Harsh Prakash, and Priya Singh 

Abstract Skin cancer, the most common type of cancer, develops when abnormal skin cells proliferate erratically. Melanoma, Squamous Cell Carcinoma and Basal Cell Carcinoma are the three most prevalent kinds of skin cancer. Regular skin self-examinations for changes or anomalies are essential since skin cancer can be effectively treated when found early. In recent years, deep learning algorithms like You Only Look Once (YOLO) have demonstrated remarkable results in object detection tasks, including detecting skin lesions. This study investigates the utilisation of YOLOv7-tiny for skin cancer detection from dermoscopic images using edge computing devices. A method is proposed that combines transfer learning and data augmentation techniques to enhance the YOLOv7 algorithm's accuracy for skin lesion detection on edge computing devices. The experiments reveal that the proposed framework achieves an impressive 82.1% accuracy in detecting skin lesions. Additionally, an extensive evaluation is performed using a sizeable dermoscopic image dataset, demonstrating its potential for clinical application in early skin cancer detection. This study contributes to the growing field of computer-aided skin cancer diagnosis and has the potential to enhance patient outcomes while lowering skin cancer-related medical expenses.

Keywords Deep learning · Skin cancer · Medical image processing · Skin lesion · Edge computing · Learning systems

1 Introduction

In the current decade, skin cancer is the most common type of cancer [1]. Skin cancer occurs more frequently than the sum of all cancers. As stated by the World Health Organisation, around a third of all cancers are skin cancers, resulting in the prevalence of skin cancer being predicted to rise over time [2]. There are several

D. Datta · H. Prakash · P. Singh (✉)
Delhi Technological University, Bawana Road, Rohini, New Delhi, India
e-mail: priya.singh.academia@gmail.com

types of skin cancer, such as melanoma, intraepithelial carcinoma, squamous cell carcinoma, and more [3]. Skin cancer has a mortality rate of up to 75% [4].

In many areas, including speech synthesis, object tracking, object detection, image classification, and natural language processing, Deep Learning (DL) techniques have demonstrated promising results [5]. DL has gained popularity in medical imaging due to its greater capability. By using trainable filters and pooling operations, Convolutional Neural Networks (CNNs), a form of DL approach, automatically extract a variety of complex high-level properties from unprocessed input images [6, 7].

In the fields of healthcare and medicine, timely identification of illnesses is crucial for effective treatment planning. Early diagnosis plays a vital role in the success of treatment and the implementation of mitigation measures, ultimately impacting the chances of survival.

Previously, the primary procedure used by clinicians to diagnose most cancers, a biopsy was primarily employed to detect melanoma. Only a biopsy can definitively confirm the diagnosis of cancer, even when other tests may suggest that it is present [8]. The procedure is often carried out in a clinic, and it can take up to a week to get the findings that reveal whether a tumour is benign or malignant. Edge computing devices can be highly helpful in reducing waiting times and increasing the effectiveness of diagnosis and treatment recommendations in places with scarce medical resources, such as rural areas. Our contributions in this paper are:

- To provide a cost-effective and user-friendly pre-screening tool for individuals in underdeveloped regions to detect skin cancer.
- The YOLOv7-tiny model is evaluated for its performance in predicting skin cancer, with the goal of making it applicable for use on edge computing devices like the Raspberry Pi.
- The proposed tool aims to minimise the need for people to undertake expensive and time-consuming trips to hospitals for medical consultations and only identify individuals with a high risk of cancer, thus avoiding overburdening medical professionals.

The following describes how the document is organised: In Sect. 2, the related and relevant state-of-art techniques available in the literature are discussed. Subsequently, in Sect. 3, the overall methodology adopted in the present work is recommended. Later in Sect. 4, the outcomes are presented. Finally, Sect. 5 presents the results and suggests future scope.

2 Related Work

This section has reviewed the many deep learning methods that researchers have used to work on skin cancer diagnosis.

Nie et al. [1] explain that DCNNs, particularly YOLO algorithms, are effective in improving the accuracy of melanoma diagnosis due to the similarity between benign and malignant dermoscopic images. YOLO has been successful in lightweight

systems, with a mean average precision (mAP) exceeding 0.82 with a training batch of only 200 images. YOLOv2 achieved an accuracy of approximately 83% in a study with a short dataset, which was the best performance. However, larger datasets and more melanoma classes are needed for further improvement.

Banerjee et al. [2] used YOLO to suggest a two-phase segmentation method. On different datasets, a Jac score of up to 88.64% was attained, demonstrating increased segmentation and classification precision. Further validation calls for more study and larger datasets.

Unver et al. [3] proposed pipeline segments for skin lesions using GrabCut and YOLO. It can be utilised as a dimension-independent segmentation strategy and shows a 90% sensitivity rate on the ISBI 2017 dataset. For the categorization of melanoma, YOLOv3 is advised.

Wei et al. [4] proposed that the skin cancer recognition model improves performance by using a feature discrimination network with two feature extraction modules. On the ISBI 2016 dataset, a lightweight U-Net-based semantic segmentation model performs better than DL-based methods.

Chhatlani et al. [9] use machine learning and image processing algorithms to detect melanoma skin cancer. The suggested methodology uses the YOLOv5 model to produce results with an average precision of 89% that are comparable to those of traditional biopsy techniques. The finished product is a web application that uses effective visualisation methods to completely forecast melanoma. The model can be improved to function even more smoothly with future development and optimisation.

Vaidya et al. [10] use the YOLOv7 model trained on the PlantDoc dataset to create a digital solution for early plant disease identification. With a compact size of 75.1 MB and a quick detection time of 6.8 ms, the model achieves a mean average precision of 71%. Future studies may involve increasing the dataset size and enhancing the image quality.

Hasya et al. [11] seven different forms of skin tumours are classified by a skin cancer detection system employing YOLOv3 with 96% absolute accuracy and 80% real-time accuracy. DL holds the potential for early diagnosis in medicine.

As far as we know, the suggested methodology has not been applied in any studies that have attempted to address this problem statement.

3 Methodology

This section elaborates on the research's methodology. Included in it will be an explanation of the dataset used, the transfer learning strategy, and the experimental setting used.

Table 1 Collected data samples

Dataset	Training	Validation
Melanoma	455	130
Benign	515	130
<i>Total sample</i>	970	260

Table 2 Augmentation applied on the dataset

Image augmentation	Settings applied
Auto-orient	Applied
90° Rotate	Clockwise, counter-clockwise, upside down
Resize	Stretch to 640 × 640
Grayscale	Apply to 10% of images
Flip	Horizontal, vertical

3.1 Data Description

A popular dataset for Skin cancer detection namely the International Skin Imaging Collaboration (ISIC) dataset used to evaluate the performance of the suggested model is obtained from Kaggle.¹ Additionally, ISIC community sponsors yearly skin lesion competitions to motivate scientists to enhance the performance of Computer-Aided Diagnostic algorithms and spread awareness of skin cancer. From a total of 2750 photographs in the 2017 segmentation challenge category, the author chose 1200 images to create balanced classes that are evenly split between benign and melanoma instances according to Table 1.

Roboflow² was used as a manual labelling tool to ensure that the photos were correctly labelled. Additionally, the platform offers a wide variety of data augmentation techniques that may be utilised to enhance the dataset and performance of the model. As seen in Table 2, a larger and more varied dataset was created using Roboflow’s built-in data augmentation features for training of YOLO model. The dependability of the model and accuracy can be considerably increased as a result, which will ultimately increase the capacity of the model to identify skin cancer in dermoscopic images.

¹ www.kaggle.com.

² www.roboflow.com.

3.2 Transfer Learning Approach

The YOLO algorithm is a system for instantaneous object detection that can identify numerous items in a picture using just one neural network. A deep CNN, the foundation of YOLO architecture, divides the incoming image into a grid of cells and then anticipates the bounding boxes and class probabilities for each cell. Anchor boxes, which are established fixed sizes and shapes that aid the network in becoming increasingly accurate at anticipating bounding boxes, are used by the YOLO network to predict bounding boxes. In order to concatenate feature maps from older layers to later ones in the YOLO architecture, which aids the network in detecting objects at various scales and resolutions. YOLO is a strong and effective object detection system that is frequently used in many different applications, such as image and video surveillance, self-driving automobiles, and medical picture analysis.

Due to the YOLOv7-tiny model’s higher performance on edge computing devices with the constrained processor and memory resources, we chose to employ it for our research. In order to be more effective for edge computing, the YOLOv7-tiny model is a version of the original YOLOv7 model with fewer layers and parameters. Like other YOLO models, YOLOv7-tiny recognises and categorises items in real time using deep CNN. Due to its reduced size and fewer layers, the YOLOv7-tiny object detection model performed better on the edge computing device used in this study. It is a reliable and effective model that consistently produces excellent results for a variety of real-world applications.

As seen in Fig. 1, the architecture of YOLOv7-tiny is made up of a backbone of convolutional layers that feature extraction from the source image. A detection head then forecasts the bounding boxes and class probabilities of the objects from image using these features. The use of a single detection head that predicts boxes at various scales, as opposed to utilising separate heads for each scale, is a significant distinction between YOLOv7-tiny and other YOLO models. The model’s efficiency is improved and the number of parameters is decreased as a result. The adoption of

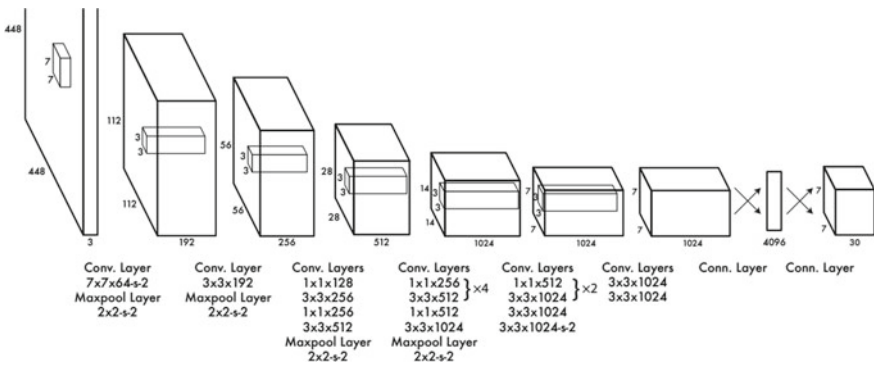


Fig. 1 YOLO architecture with convolutional layers taken from [12]

Table 3 Model’s training parameters

Training parameters	Values
Epochs	400
Momentum	0.9
Batch size	64
Decay	0.0005
Learning rate	0.0026

a unique neck design, which collects data from several scales to increase detection accuracy, distinguishes YOLOv7-tiny from prior YOLO models. This is accomplished by combining features from several network layers using a feature pyramid network. Since YOLOv7-tiny performs more quickly than other YOLO models, for real-time object detection applications, it is a good option. Due to its fewer layers, it might not perform as well on some detection tasks, such as detecting small objects. Because it provided a decent mix between speed and precision and was thus a viable model for our intended application, we decided to work with YOLOv7-tiny.

3.3 *Experimental Setup*

The MS COCO dataset served as the YOLOv7 model’s pre-training data and is now being trained on the author’s dataset using transfer learning. The Tesla P-100 GPU made available by Kaggle is used for training together with the hyperparameters listed in Table 3.

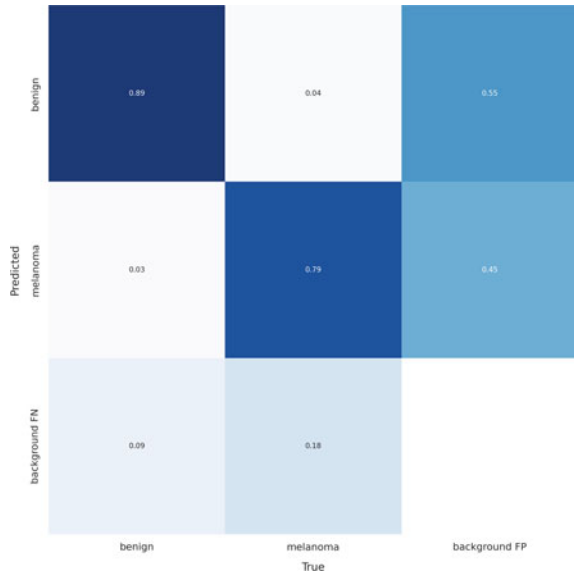
4 **Result and Discussion**

The findings of this study are explained in this section, as well as how different factors were used to gauge how well the research model performed. As demonstrated in Table 4, the YOLOv7-tiny model trained in this study obtains mAP of 0.821, accuracy of 0.8, and recall of 0.8521.

Table 4 Evaluation metrics for the model

Evaluation metrics	Values
Precision	0.8
Recall	0.8521
mAP	0.821

Fig. 2 Classification results using confusion matrix



80% of all positive events that the model predicted were correctly identified, according to the precision value of 0.8. This indicates that the model’s optimistic forecasts are generally accurate to a high degree. The recall value of 0.8521 shows that 85.21% of the actual positive events were picked up by the model. This indicates that the model is fairly sensitive to picking up on positive events. The object detection model’s overall performance is shown by the mAP value of 0.821. It is determined as average of precision values at various recall thresholds, taking into account both the accuracy and recall values. A model that performs better has a higher mAP value. Figure 2 illustrates the classification results in the form of a confusion matrix.

The mean average precision (mAP) is a commonly used evaluation measure in computer vision for tasks such as object detection, localization, and classification. Object detection algorithms, segmentation systems, and information retrieval methods are often assessed based on their performance using the mAP metric. Localization involves determining the precise location of an object using parameters such as bounding box coordinates, while classification involves identifying the object based on its characteristics (such as whether it is a dog or a cat). Figure 3 shows the mAP curve for the model in our study.

By having high precision and recall values and an overall mAP score of 0.821, these results indicate that the object detection model did a good job of recognising positive cases. When compared with the previous YOLO series model, despite its small size, it produces a fairly comparable accuracy of 82.1% as compared to previously done researches [2]. Despite current progress, there is room for improvement in skin cancer detection through the use of edge computing devices. To achieve more accurate results, it is essential to refine deep convolutional neural networks

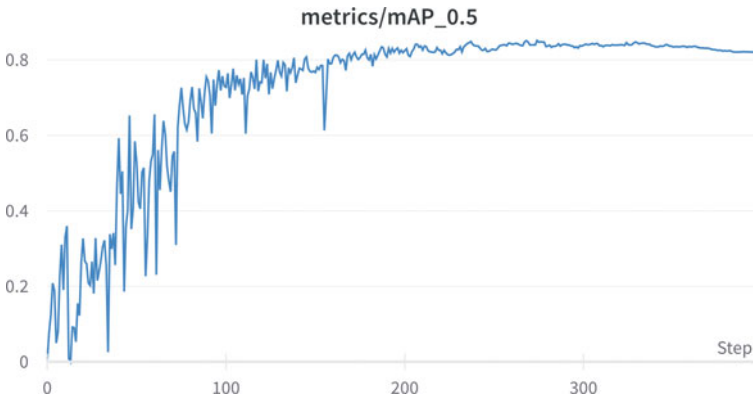


Fig. 3 Model mAP curve

(DCNNs) and increase the number of melanoma categories included in datasets. This study highlights the importance of these improvements, as they can lead to better classification outcomes.

5 Conclusion

This research focuses on developing an accurate skin cancer detection method using edge computing devices by leveraging the latest YOLOv7 model, a member of the YOLO series of single-shot object detection models. ISIC dataset is utilised to train YOLOv7 model, yielding superior results compared to its predecessors for the same task. The model achieved a significant precision of 80% recall of 85.2%, and mAP of 81.2%. The use of the ISIC dataset enables real-world deployment of the model in regions with limited access to healthcare facilities. Nonetheless, there is potential for enhancement in future models and datasets to deliver more effective skin cancer detection solutions using edge computing devices. This study emphasises the necessity for refining the DCNN network and expanding datasets to include more melanoma categories, which will lead to improved classification outcomes.

References

1. Universitatea de Medicina si Farmacie "Gr. T. Popa" Iasi, Institute of Electrical and Electronics Engineers and IEEE Engineering in Medicine and Biology Society. Romania Chapter (n.d.) 2019 E-health and bioengineering conference (EHB): EHB 2019-7-th edn. Iasi, Romania, 21–23 Nov 2019
2. Banerjee S, Singh SK, Chakraborty A, Das A, Bag R (2020) Melanoma diagnosis using deep learning and fuzzy logic. *Diagnostics* 10(8). <https://doi.org/10.3390/diagnostics10080577>

3. Ünver HM, Ayan E (2019) Skin lesion segmentation in dermoscopic images with combination of YOLO and grabcut algorithm. *Diagnostics* 9(3). <https://doi.org/10.3390/diagnostics9030072>
4. Wei L, Ding K, Hu H (2020) Automatic skin cancer detection in dermoscopy images based on ensemble lightweight deep learning network. *IEEE Access* 8:99633–99647. <https://doi.org/10.1109/ACCESS.2020.2997710>
5. Albahar MA (2019) Skin lesion classification using convolutional neural network with novel regularizer. *IEEE Access* 7:38306–38313. <https://doi.org/10.1109/ACCESS.2019.2906241>
6. Nasr-Esfahani E, Samavi S, Karimi N, Soroushmehr S, Hossein Jafari M, Ward K, Najarian K (2016) Melanoma detection by analysis of clinical images using convolutional neural network. https://doi.org/10.0/Linux-x86_64
7. Malhotra R, Singh P (2023) Recent advances in deep learning models: a systematic literature review—multimedia tools and applications. Springer. <https://doi.org/10.1007/s11042-023-15295-z>
8. Singh P, Kumar M, Bhatia A (2022) A comparative analysis of deep learning algorithms for skin cancer detection. In: 2022 6th international conference on intelligent computing and control systems (ICICCS). <https://doi.org/10.1109/iciccs53718.2022.9788197>
9. Chhatlani J, Mahajan T, Rijhwani R, Bansode A, Bhatia G (2022) DermaGenics—early detection of melanoma using YOLOv5 deep convolutional neural networks. In: 2022 IEEE Delhi section conference, DELCON 2022. <https://doi.org/10.1109/DELCON54057.2022.9753227>
10. Vaidya S, Kavthekar S, Joshi A (2023) Leveraging YOLOv7 for plant disease detection. In: 2023 international conference on innovative trends in information technology. ICITIIT 2023. <https://doi.org/10.1109/ICITIIT57246.2023.10068590>
11. Hasya HF, Nuha HH, Abdurrohman M (2021) Real time-based skin cancer detection system using convolutional neural network and YOLO. In: Proceedings—2021 4th international conference on computer and informatics engineering: IT-based digital industrial innovation for the welfare of society, IC2IE 2021, 152–157. <https://doi.org/10.1109/IC2IE53219.2021.9649224>
12. Redmon J, Divvala S, Girshick R, Farhadi A (2015) You only look once: unified, real-time object detection. <http://arxiv.org/abs/1506.02640>

Effective Image Captioning Using Multi-layer LSTM with Attention Mechanism



Japnit Singh, Kishan Kumar Garg, and Arahant Panwar

Abstract Image captioning has become one of the more pressing problems in machine learning with a lot of headway being made in generating descriptive image captions in the English language. This is due to the ease of availability of the English dataset but that should not restrict any work in native languages; therefore to overcome this problem, we have used the Flickr8K dataset with Google Cloud Translator as done in Ankit Rathi et al. to obtain Hindi captions. In this study, we proposed an image captioning model that utilizes InceptionV3 as a feature extractor and a stacked LSTM model with Bahdanau attention to generate captions for given images. The performance of the proposed model was evaluated on the Flickr8k Hindi dataset using the BLEU metric. The results showed a BLEU-1 score of 0.64294, BLEU-2 score of 0.480917, BLEU-3 score of 0.36554, and a BLEU-4 score of 0.2191, which is considered as the most significant score.

Keywords Flickr8K Hindi dataset · InceptionV3 · LSTM · Bahdanau attention

1 Introduction

One of the challenging problems in the areas of computer vision and artificial intelligence is the generation of meaningful textual descriptions of images. Although a lot of research has been done on the topic, there are few works on image captioning in languages aside from English due to a lack of datasets in other languages. Our motivation to work in this field was to contribute toward bridging language barriers for Hindi-speaking audiences. There has not been extensive research done in image captioning in non-English languages. In Hindi-speaking countries like India, it may also create new chances for image captioning in industries like journalism, social media, and e-commerce.

J. Singh (✉) · K. K. Garg · A. Panwar
Delhi Technological University, Delhi Bawana Road, Rohini, New Delhi 110042, India
e-mail: japnit2012@gmail.com
URL: <http://dtu.ac.in>

The use of an encoder–decoder architecture is one of the finest methods for producing image captions. The following is a summary of the paper’s main contributions:

- Feature extraction is performed by InceptionV3 pre-trained on ImageNet.
- The text is preprocessed using steps such as tokenization, text cleaning, vocabulary creation, and padding and truncation.
- Multi-layer LSTM with Bahadanau attention mechanism is used to generate captions.

Combinations of varying types of CNN and LSTM models have been implemented in the past [1–3]. Inspired by their approach, we too have used a CNN-based encoder (InceptionV3) but with an attention mechanism. It is one of the guiding concepts of deep learning, and it is used to correctly distinguish between the relative relevance of distinct areas in an image’s characteristics so that the model can focus on those parts that have a bigger effect on the classification results and pay less attention to the other regions.

In this study, with Bahdanau’s attention, we offer a novel method for captioning Hindi images utilizing inception and multi-layered LSTM. The model’s ability to concentrate on various aspects of the image while creating captions is made possible by the employment of attention mechanisms, which raises the caliber of the captions. In order to improve the pre-trained inception model on Hindi picture datasets, we additionally use transfer learning techniques.

2 Related Works

This section discusses contemporary picture captioning as well as the various research approaches. Vinyals et al. [4] presented a generative model that has used machine translation’s recent advancements and can be used to produce real-world phrases that describe images. The target sentence is created by reading the source and being converted to a vector form by a “decoder” RNN. Deshpande et al. [5] have proposed to first predict a meaningful summary of the image and then generate the description based on it. Since the development of captions should be based on their summary, they have employed part-of-speech (POS) summaries. Their technique produces captions by first (a) extrapolating several summaries from the image and then (b) forecasting captions based on each summary. Rathi et al. [3] created a new version of the Flickr8K dataset by converting the English captions to their Hindi counterparts by using the Google Cloud Translator API and suggested an encoder–decoder model where CNN and RNN-LSTM are used as encoders to encode text data and images, respectively, and linguistic neural network model has been used in the decoder. Mishra et al. [6] suggested a novel approach for image captioning in Hindi language, based on encoder–decoder deep learning architecture with an “attention mechanism”. The proposed architecture employs scaling in CNNs to achieve higher accuracy than the accuracy achieved in the existing work. Mishra et al. [7] proposed to manually create

a dataset from well-known MSCOCO dataset by translating the existing dataset in order to provide captions in Hindi language. He encoded the input image using ResNet-101, then used the GRU to decode it into the suitable description. Poddar et al. [2] proposed a CNN-LSTM model with multiple layers for automatically identifying items in photos and producing appropriate descriptions for them. In order to recognize objects using deep learning techniques, it uses models that are based on transfer learning.

3 Proposed Methodology

We describe our approach in this section for caption generation of an image using an encoder–decoder-based architecture. The architecture involves two main components: an encoder that extracts features from the input image and a decoder that generates a sequence of words based on the encoded features. Our focus is on maximizing the correctness of the captions by exploring different decoder models, such as the LSTM network.

3.1 Encoder–Decoder Architecture

We look into architectures based on encoders and decoders for captioning images. An image is provided, which maximizes the accuracy of the caption descriptions. Figure 1 illustrates the details of the encoder–decoder architecture we employed in our study. The construction of encoder–decoder model involves three main stages.

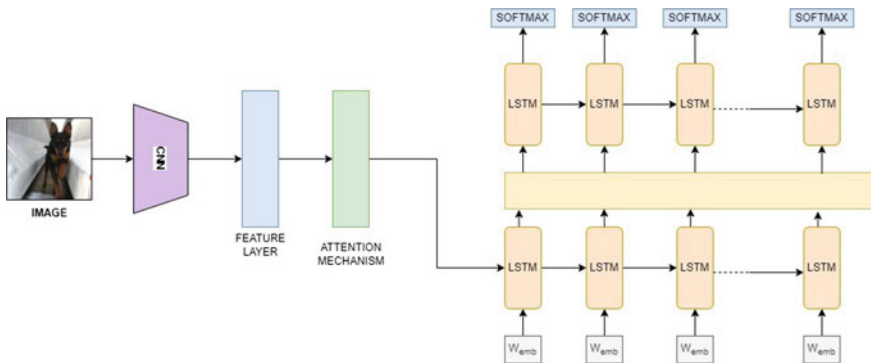


Fig. 1 Encoder–decoder architecture

3.2 Feature Extraction

We adopt the convolutional neural network (CNN), InceptionV3 pre-trained on ImageNet database. InceptionV3 was developed by Google in 2015 [8] to improve accuracy and efficiency for image classification tasks. It has 42 layers and offers better accuracy and computational efficiency than its predecessors through the use of advanced architectural features and training techniques. Its pre-training on ImageNet enables transfer learning, making it a valuable tool for various image processing tasks.

3.3 Text Preprocessing

In image captioning, text preprocessing is the process of organizing and preparing the textual information that will be utilized to produce image captions. In order to ensure that the resulting captions are correct and meaningful, text preparation is essential in image captioning. Without adequate text preprocessing, the machine learning algorithms could have trouble deciphering the text's content and producing captions that make sense, yielding subpar results. Text preparation usually entails the following steps:

- **Tokenization:** The initial stage of text preparation entails tokenizing or cutting the caption into smaller bits. Usually, this is accomplished by breaking the text down into words; however, character-based tokenization and sub-word tokenization are also viable options.
- **Text cleaning:** Unwanted letters, punctuation, or special symbols may be present in the caption text. To make the caption text simple to process, some extraneous parts are removed through text cleaning.
- **Vocabulary creation:** The remaining unique words or tokens are all added to the vocabulary or dictionary, together with an index that lists each word in the vocabulary, to complete the vocabulary. As a result, the vocabulary words and their related indices are mapped one to one.
- **Padding and truncation:** To ensure that each input sequence to a machine learning model has the same length, padding and truncation are text preprocessing techniques utilized. Truncation removes sequence segments that are too long, whereas padding appends zeros or a particular character at the end of a sequence.

3.4 Attention Mechanism

The proposed encoder–decoder model makes use of Bahdanau attention. [9] proposed this architecture that learns joint alignment and translation. Due to the linear combination of encoder states and decoder states that it conducts, it is also referred to as additive attention. The context vector is computed as follows:

$$c_t = \sum_{i=1}^N \alpha_{ti} v_i, \quad (1)$$

the weight α_{ti} assigned to each feature v_i can be calculated as:

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{i=1}^N \exp(e_{ti})}, \quad (2)$$

where

$$e_{t,i} = f(h_{t,i}). \quad (3)$$

3.5 Language Modeling

For our study, we have explored the performance of various decoder models through an ablation study, where we systematically remove certain components from the model to evaluate their impact on the overall performance. Based on our analysis, we have determined that using LSTM as the decoder architecture for generating image captions is the best approach. The LSTM network generates captions in a stepwise manner by producing a single word at each time step. This process involves taking into account a context vector, the preceding hidden state, and the words generated in the previous steps. Because a shallow LSTM network has a limited ability to grasp long-term dependencies in the input sequences, it might not be adequate to capture the complexity of language. To address this issue, we have used multi-layer LSTM networks, such as stacked LSTMs. These networks can help address the issue of limited capacity by allowing for more complex and abstract representations of language by building on the learned representations of previous layers. By stacking multiple LSTMs, the network can capture more complex dependencies in the input sequence and generate more accurate and meaningful captions.

4 Experimental Setup

This section outlines the approach used to develop the dataset and assess the effectiveness of the proposed methodology.

4.1 Dataset

Our models were trained on the uncleaned five-sentence sample. The Hindi version of the Flickr8k dataset contains five Hindi language captions for each image. The dataset is comprised of 8000 images for training and 1000 images for validation and testing purposes.

4.2 Evaluation Metric

As a measurement tool, we have employed Bilingual Evaluation Understudy Score (BLEU), which compares generated captions to reference captions to determine how comparable they are. It was first put forth by Papineni [10].

4.3 Hyperparameters Used

Features are extracted using the InceptionV3 model and converted into 64×2048 feature vectors from images with a size of 229×229 . A dropout rate of 0.4 is applied to avoid overfitting, and an embedding layer with 300 neurons is used. The model is trained with a fixed batch size of 64 and optimized using the Adam optimizer with a learning rate of $3e - 4$. Softmax cross-entropy is employed as the loss function. To improve performance, multiple layers of LSTM were stacked, and it was found that the best BLEU score was achieved with three layers.

5 Results and Discussion

Table 1 summarizes the results of comparing the performance of our proposed image captioning model with existing methods on the Flickr8k Hindi dataset. The models are identified as Inception-LSTM- k , where k represents the number of LSTM layers used in the model. The results indicate the proposed image captioning model with three layers of LSTM outperformed the other models, achieving the highest BLEU scores across all metrics. Figure 2 represents the captions generated by our proposed model on images 1–8 of our randomly chosen validation set.

Specifically, the model obtained the BLEU-1 score of 0.6429, BLEU-2 score of 0.4809, BLEU-3 score of 0.3655, and BLEU-4 score of 0.2191 indicating its ability to generate more accurate and diverse captions.

Additionally, the model was compared to hybrid CNN-LSTM [2] model and VGG16-LSTM model [3]. The hybrid CNN-LSTM model [2] achieved lower scores

Table 1 Comparison with baselines

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Inception-LSTM-1	0.594435	0.440589	0.337126	0.197693
Inception-LSTM-2	0.60664	0.44176	0.32856	0.18831
Inception-LSTM-3	0.64294	0.480917	0.36554	0.21907
Hybrid CNN-LSTM	0.55698	0.35914	–	–
VGG16-LSTM	0.5844	0.4	0.27	0.12



(a) एक आदमी और एक आदमी पानी में एक नाव में हैं।

A man and another man are in the water in a boat.



(b) एक बच्चा एक छोटे बच्चे के साथ खेल रहा है।

A child is playing with a small child.



(c) दो कुत्ते एक लाल गेंद को पकड़ने के लिए एक साथ खेल रहे हैं।

Two dogs are playing together to catch a red ball.



(d) एक लड़का एक चट्टान पर चढ़ रहा है।

A boy is climbing a rock.



(e) एक कुत्ता एक बाड़ के रास्ते पर चलता है।

A dog walks on a road beside a pond.



(f) एक आदमी पानी में कूद रहा है।

A man is jumping in the water.

Fig. 2 Image description: images with corresponding captions generated by our proposed model. These are images 1–6 of our randomly chosen validation set

across all metrics, indicating that the proposed model is more effective at generating captions. The VGG16-LSTM model [3] achieved competitive scores but still performed slightly lower than the proposed model, indicating that the attention mechanism in the proposed model helps to focus on important image features and generate more accurate captions.

Overall, the results suggest that the proposed CNN-LSTM model with attention mechanism is an effective approach for image captioning tasks and can generate high-quality captions compared to other models. Further improvements can be made by exploring other architectures or incorporating additional features to improve the model's performance.

6 Conclusion

The objective of the research was to propose a novel approach for Hindi image captioning using InceptionV3 and multi-layered LSTM with Bahdanau attention. The proposed model aimed to generate captions for images in Hindi language. To achieve the objective of generating meaningful captions for images in Hindi language, we used inception for feature extraction from images, multi-layered LSTM with Bahdanau attention for caption generation, and transfer learning techniques to fine-tune the pre-trained InceptionV3 model on the Flickr8k Hindi dataset. This dataset contains images with captions in Hindi language, which made it suitable for our experiments. The proposed model was evaluated using BLEU scores. The model obtained BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores of 0.6429, 0.4809, 0.3655, and 0.2191, respectively, which indicate that the model is able to generate captions in Hindi language with reasonable accuracy. A limitation of the proposed model is that it could potentially benefit from larger dataset and more advanced algorithms to improve its performance. However, an advantage of the current model is that it requires low computation time, making it a viable option for applications with limited computing resources. Additionally, the proposed model can be extended to generate captions in other languages, thus making it a valuable tool for cross-lingual image captioning.

References

1. Laskar SR, Singh RP, Pakray P, Bandy-Opadhyay S. English to Hindi multimodal neural machine translation and Hindi image captioning. In: The proceedings of 6th workshop of Asian translation
2. Poddar AK, Rani R (2023) Hybrid architecture using CNN and LSTM for image captioning in Hindi language. Architecture using CNN and LSTM for image caption in Hindi language. *Procedia Comput Sci* 218
3. Rathi A (2020) Deep learning approach for image captioning in Hindi language. In: 2020 international conference on computer, electrical & communication engineering (ICCECE)

4. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. [arXiv:1411.4555](https://arxiv.org/abs/1411.4555)
5. Deshpande A, Aneja J, Wang L, Schwing A, Forsyth D (2019) Fast, diverse and accurate image captioning guided by part-of-speech. [arXiv:1805.12589](https://arxiv.org/abs/1805.12589)
6. Mishra SK, Saha S, Bhattacharyya P (2021) A scaled encoder decoder network for image captioning in Hindi. In: Proceedings of the 18th international conference on natural language processing (ICON)
7. Mishra SK, Dhir R, Saha S, Bhattacharyya P (2021) A Hindi image caption generation framework using deep learning. *ACM Trans Asian Low-Resour Lang Inf Process* 20(2):1–19
8. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the inception architecture for computer vision. [arXiv:1512.00567](https://arxiv.org/abs/1512.00567)
9. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. [arXiv:1409.0473](https://arxiv.org/abs/1409.0473)
10. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp 311–318

A Hybrid Approach for Sentiment Analysis Using Game Theory in Word Sense Disambiguation



Aryan Singhania, Harsh Gupta, and Minni Jain

Abstract This study explores the use of Evolutionary Game Theory (EGT) for the task of sentiment analysis. The proposed approach involves the use of EGT concepts to disambiguate the particular sense of a word and analyze the context in which it is used. Methods involving Evolutionary Game Theory are employed to learn the associations between different words and synsets. Each word is treated as a player and its synset space as its strategy space. The model aims to find the Nash Equilibria to correctly disambiguate all tokens. The SentiWordNet lexicon is used to identify the sentiment of each sentence. The effectiveness of the proposed approach is evaluated using labeled twitter datasets in which WSD (EGT)-wup-lesk-word2vec variation showed an accuracy of 80.6%. The results demonstrate the efficacy of using pre-trained word embeddings and the potential of WSD for sentiment analysis.

Keywords EGT · WSD · Sentiment Analysis · Game Theory

1 Introduction

Because of the recent rise in online and social media contact, sentiment analysis has become essential. Finding the sense, sentiment, and attitude behind the text in a particular sentence is what is meant by this. Sentiment analysis is essential in a variety of domains, including social media analysis, performance analysis, and the early detection of critical problems. However, currently employed methodologies include count-based metrics or simple sentiment analysis. Existing methods have issues with dealing with word ambiguity, where one word can have numerous meanings, accuracy due to limited data, and performance across domains. Using concepts from

A. Singhania (✉) · H. Gupta · M. Jain
Department of Computer Engineering, Delhi Technological University, Delhi, India
e-mail: aryansinghania_2k19co088@dtu.ac.in

H. Gupta
e-mail: harshgupta_2k19co149@dtu.ac.in

evolutionary game theory, we have employed word sense disambiguation to identify the proper sense of the word being used in the phrase.

In this paper, we propose a novel approach for sentiment analysis using word sense disambiguation. Our approach leverages techniques from evolutionary game theory to disambiguate ambiguous words in a given text, and then uses the resulting disambiguated words to perform sentiment analysis. We hypothesize that incorporating WSD into sentiment analysis can improve its accuracy and effectiveness by addressing the issue of word ambiguity. Our work aims to advance the field of sentiment analysis by addressing one of its key challenges through the integration of evolutionary game theory techniques. We believe that our proposed approach has the potential to improve the accuracy and effectiveness of sentiment analysis in practical applications such as social media monitoring and product review analysis.

This research paper makes the following contributions:

1. It identifies the effectiveness of a WSD approach for sentiment analysis using evolutionary game theory.
2. It adds to the existing literature by evaluating the impact of using different word similarity matrices.
3. It showcases the effectiveness of SentiWordNet for creating unsupervised models for sentiment analysis.
4. It highlights the benefits of using word embedding along with N-gram databases for solving NLP problems.
5. It emphasizes the future research potential of using game theory for sentiment analysis tasks.

The rest of this paper is organized as follows: in Sect. 2, we review related work on sentiment analysis and word sense disambiguation. In Sect. 3, we discuss the prerequisites required to implement the approach. In Sect. 4, we describe our proposed approach in detail, including the WSD algorithm used and the sentiment analysis technique employed. In Sect. 5, we present experimental results and compare our approach to existing sentiment analysis methods. Finally, in Sect. 6, we conclude the paper and discuss potential future directions for research.

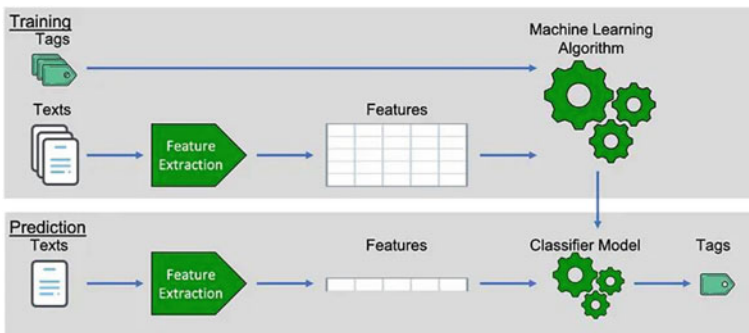
2 Related Works

For the purpose of analyzing a company's performance and establishing its place in the market, sentiment analysis is becoming increasingly crucial. Sentiment analysis, often known as opinion mining, is the process of determining the attitude, opinion, and feelings included inside a sentence. Sentiment analysis can be performed manually, in which each text is evaluated and categorized as positive or negative. However, this will be a huge waste of time, money, and labor. In light of NLP, we therefore view sentiment analysis as a fundamental categorization problem. Since it can be difficult to categorize the texts within the sentences, performing sentiment analysis presents a significant problem. It can be very challenging to identify which sense is

being used in a given line and whether it is positive or negative because every text has a variety of senses.

We will review various approaches [1] that exist and have been implemented for sentiment analysis and discuss their advantages and limitations. We also provide an overview of some open challenges and future directions in this area.

In machine learning models [2], some of the texts and their corresponding sense or tag are given to the algorithm for the training purpose so that the model can predict when new texts will be provided. Figure below represents the working of machine learning algorithms. In supervised machine learning algorithms, the data is labeled which means it would be affected by personal opinion and subjectivity. Naive Bayes [3] is one of the most commonly used algorithms for sentiment analysis.



Naive Bayes [3] simply tells us the probability whether the word used has a positive effect or negative effect. It makes predictions based on the training data and prior knowledge provided to it. To extract features from the text, we need to perform preprocessing that requires representation of data in numerical or vector form [4]. As we have to incorporate a wide feature space, a document term matrix (DTM) is constructed that also helps in improving model performance. The conditional feature probabilities are represented by the formula.

$$\hat{P}(w_i|c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)}$$

where $|V|$ represents number of unique words. Features here will contribute equally throughout the sentence as it is assumed that they are independent of each other. The basic Naive Bayes DTM model generates a score of 66% accuracy.

Apart from Naive Bayes, we can use algorithms like linear regression that plots the relation among words and senses on a straight line and support vector machine algorithm that is useful when data is complex. The drawback of DTM model that matrix formed becomes sparse having thousands of dimensions can be improved using deep learning.

Deep learning [5] uses artificial neural networks to solve complex problems in a manner humans do. As we represent the text in numeric form, we need to map the respective word with some similar numeric vector [4]. This is done using word embedding. Word embedding [5] models such as Word2Vec [6] and GloVe, represent words as vectors [4] that are learned through a neural network to predict their context in a corpus. The Word2Vec approach [6] generally has 2 architectures: Continuous Bag of Words (CBOW) and Skip-gram. The main difference between the two comes from how they use input and output layers of the model, respectively. Using word embedding, we produce vectors having fewer dimensions than the sparse matrix in the DTM model, therefore helping us with dimensionality reduction. However word embedding methods are highly expensive as compared to other methods.

Rule-based models [7], on the other hand, are easily implemented and understood and are comparatively cheaper. Knowledge-based approach [7, 8] majorly used Valence Aware Dictionary and sEntiment Reasoner (VADER) tool for providing sentiment scores. We can produce a compound score that takes into consideration the polarity and intensity of the text sentiments. The scores produced are normalized and adjusted according to the rules so that they fall between -1 (negative) to $+1$ (positive). These methods have comprehensive linguistic rules and powerful lexicon [9]. We can produce a score of upto 72% accuracy using VADER. There are several difficulties in rule-based models like it is very time consuming, it might produce endless combinations of features.

Determining the correct sense of the word also poses a major challenge. For classifying the text sentiments as $+$ effect or $-$ effect, we need to find the correct sense being used and the nature of that particular sense. There are approaches like a dictionary-based approach, considering the sense that occurs the majority number of times, senses depending on co-occurring words. In this paper, we have used word sense disambiguation [10] for determining the correct sense of the word being used in the sentence.

Word sense disambiguation [10, 11] helps us to disambiguate the ambiguous words and is applied in various fields like sentiment analysis [12], opinion mining, hate speech detection, and many more. There are several types of algorithms that can be used to solve the WSD. They include supervised algorithms [13], unsupervised algorithms, and semi-supervised algorithms. In the past, it has been demonstrated that supervised machine learning techniques are more accurate than unsupervised ones at disambiguation. The drawback of supervised approaches is that each phrase that needs to be disambiguated needs to have training data that has been manually annotated. However, manual annotation is a costly, challenging, and time-consuming technique that cannot be used on a big scale. Cluster-based algorithms (measuring the entropy) can be used for unsupervised learning as dictionary sense training data is not required here. Due to their independence from human labor, unsupervised approaches offer enormous potential to break the knowledge acquisition bottleneck. Semi-supervised algorithms are in demand now-a-days as they can use both labeled as well as unlabeled datasets.

We can use dictionary-based methods that use encoded knowledge of words and senses. They don't need any pre-defined dictionary or corpus and work on the

database structures and its properties. The baseline accuracy for this is around 28% that always chooses the most frequently used sense.

Here our aim is to optimize a model using both knowledge-based approach [11] and graph-based approach. We will be making a graph where nodes only represent the words, not the senses. We will try to find the interaction of every pair of words and then calculate the similarity among them. Unit interactions result in the emergence of properties, which in our instance can be seen as meanings according to the problem specification. Our model uses a distributional approach to a similarity measure to weight the relationships between words, giving words with close relationships more weight overall.

3 Prerequisites

3.1 *Evolutionary Game Theory*

Evolutionary game theory is a mathematical framework for simulating the strategic interactions among members of a population. Our approach simulates the interactions of various word senses as a game in which each sense is represented by a player. Based on the results of their interactions, the players modify their methods over time, and successful strategies spread more widely across the population. The use of evolutionary game theory in our method allows for a more effective approach to identifying the correct synset associated with each word allowing for a greater accuracy on the sentiment analysis task.

3.2 *N-Grams*

An N-gram is a sequence of N words that appear together in a text. It is used to simulate how likely different word sequences are in a language. The quantity of context recorded and the data sparsity is determined by the value of n selected. We use precomputed N-grams to increase the accuracy of the word similarity measure.

3.3 *SentiWordNet*

The SentiWordNet lexical database is an extension over the pre-existing WordNet database which annotates each synset with a sentiment score. We use SentiWordNet as the knowledge-base for our model, which we use to derive the set of sentiment-tagged synsets associated with each word.

4 Proposed Method

We propose a knowledge-based algorithm for sentiment analysis which makes use of word sense disambiguation and game theory. Our approach seeks to solve some of the shortcomings of previous methods by providing a more thorough and interpretable method of modeling text semantics. We describe a more effective strategy for predicting the proper interpretation of an ambiguous word using Evolutionary Game Theory and the semantic information acquired by N-grams. We then make use of these results to identify the sentiment of each sentence using a sentiment-tagged knowledge-base.

1. Parse and preprocess each sentence in the dataset. Each sentence must be lemmatized, and POS-tagged in order to correctly identify the set of synsets related to the word in its current context.
2. Remove monosemous words and stop words
3. Create the word similarity matrix using the N-gram dataset.
4. Use the lexical knowledge-base to create a set of all possible senses related to the identified POS for each polysemous word. Each word represents a player, and each cell in the matrix represents the probability that the two players interact.
5. Compute the sense-similarity matrix and ensure that the similarity values in each cell are unique. This can be done by using an augmented similarity measure which comprises the results of multiple other measures.
6. Simulate games between all words and apply replicator equations for each game to compute the Nash Equilibrium for that particular game and update the mixed strategy for the players. Repeat these steps until an equilibrium is reached.
7. Assign the synset with the highest probability to each of the polysemous words and use the sentiment values to identify the positive/negative score associated with each word.
8. Calculate the net positivity score of the sentence using the individual positivity scores of each word to classify it into positive, negative, or neutral classes.

4.1 *Constructing Word Similarity Matrix*

We take into account the co-occurrence and collocation of two words in order to calculate the similarity between them. To obtain the co-occurrence and collocation information, we use the Tagged and Cleaned Wikipedia N-gram dataset. The dataset consists of N-grams ($1 \leq n \leq 5$) created from all of the text on Wikipedia. We make use of 1-g, 2-g, and 5-g to obtain the total occurrences, collocation data, and co-occurrence data, respectively. The similarity matrix can be further augmented by using other sources of information which also take into account the syntactic structure of the sentence and the syntactic co-dependence of two words.

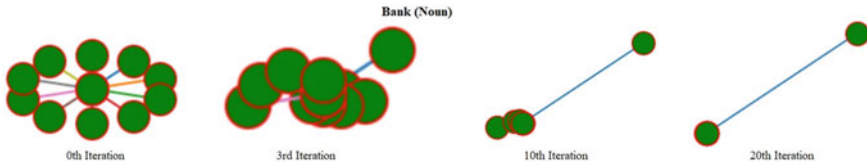


Fig. 1 Identification of the correct synset for “bank” in the sentence “My car passed by the river bank.” Initially, all synsets are equally probable. By the 10th iteration of the games, a particular synset (bank.n.01: sloping land (especially the slope beside a body of water)) is clearly favored by the model. The Nash Equilibrium is reached by the 20th iteration

4.2 Constructing Sense-Similarity Matrix

SentiWordNet is used to get the set of all possible sense tagged synsets associated with each word. A sense-similarity matrix is created in which the semantic similarity between each pair of senses is stored. The semantic similarity between two word senses can be calculated in multiple ways, such as using the depth of the first common WordNet ancestor or using WordNet is-a and has-a relationships and the information content associated with WordNet concepts.

We make use of the Wu-Palmer similarity measure to find the similarity between two synsets. The wup-similarity measure uses the depth of the Least Common Subsumer (the first common ancestor) to calculate the similarity between two WordNet nodes according to the following formula:

$$\text{Wu - Palmer} = 2 * \frac{\text{depth}(\text{lcs}(s1, s2))}{(\text{depth}(s1) + \text{depth}(s2))}$$

The Wu-Palmer similarity measure fails to give a useful result in case the two synsets have no common ancestor node. To offset this problem, we augment the similarity measure with a modified form of the adapted lesk measure and Word2Vec sentence similarity measures to calculate the similarity between synsets. The modified lesk measure uses the words in the synset definition and the definitions of its related hypernyms, hyponyms, holonyms, and meronyms to calculate the similarity between two synsets, while the Word2Vec measure uses the word embeddings from a pre-trained Word2Vec model to calculate the sentence similarity. The results from the similarity measures are normalized and a weighted average over the three measures is taken for each pair of synsets to obtain the sense-similarity matrix (Fig. 1).

4.3 Simulating Games

For each word, a synset represents a possible strategy that it can play. The strategy space for a player thus consists of the set of all possible synsets that can be assigned to it. The strategy-payoffs are initialized with an equal probability. In each iteration, the

players play a game with each of their neighbors and reevaluate their strategy-payoffs. The algorithm is terminated once the Nash Equilibrium is reached.

4.4 *Assigning Sentiment Classes*

The sentiment score, S , for a document is assigned using the sentiment scores, $s_{i(\text{postneglobj})}$, assigned to each disambiguated synset in the sentence. The total sentiment score for a sentence is calculated as:

$$S = \Sigma(s_{i(\text{pos})} - s_{i(\text{neg})})$$

The document is assigned a positive label if $S > 0$, a negative label if $S < 0$, and a neutral label otherwise.

5 Experimentation and Results

5.1 *Dataset*

The Twitter sentiment analysis dataset, which consists of 162,980 unique tweets classified into 3 categories (positive, negative, and neutral), is utilized for testing the model. Each tweet has been tokenized, lemmatized, and tagged before being consumed by the model.

5.2 *Results*

The model was evaluated on the Twitter sentiment dataset. Each document was assigned a class using the sentiment scores associated with it. Results were obtained using multiple sense-similarity measures (Table 1).

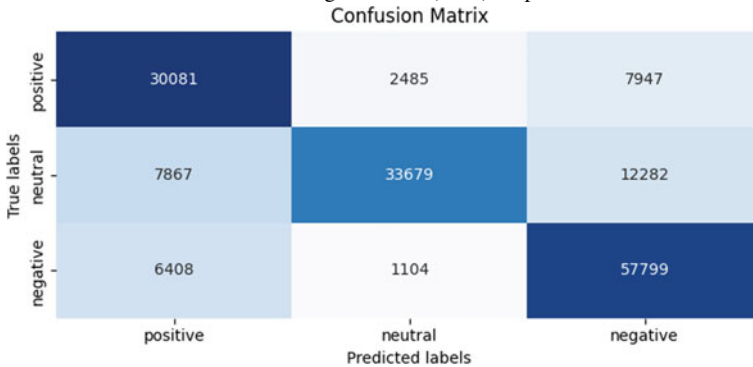
The adapted lesk measure is able to alleviate the issues with missing similarity scores when using only wup-similarity to a very small extent. Augmenting the sense-similarity scores with the Word2Vec similarity scores boosted the model's accuracy significantly (Table 2).

Table 1 Comparison with sentiment analysis models using SentiWordNet

Sense-similarity measure	Accuracy (%)
Weighted SentiWordNet scores [14]	61
Word relevance (SVM) [15]	77.68
WSD (EGT) + wup-similarity	75.94
WSD (EGT) + wup-similarity + adapted lesk	76.12
WSD (EGT) + wup-similarity + adapted lesk + Word2Vec	80.60

Bold means the proposed methodology result.

Table 2 Confusion matrix for model using the WSD (EGT)-wup-lesk-word2vec measure



6 Conclusion and Future Work

This study proposed a knowledge-based sentiment analysis model using SentiWordNet, Word Sense Disambiguation and concepts from Evolutionary Game Theory, for the Twitter sentiment analysis dataset. The findings demonstrated that the Game Theoretic model performed well in the sentiment analysis task. The efficacy of using pre-trained word embeddings (Word2Vec) was also demonstrated. Performance can possibly be increased even further by modifying the model to operate with larger knowledge bases and N-gram datasets, as well as employing more robust similarity metrics.

References

1. Devika MD, Sunitha C, Ganesh A (2016) Sentiment analysis: a comparative study on different approaches. Proc Comput Sci 87:44–49. <https://doi.org/10.1016/j.procs.2016.05.124>
2. Kawade D, Oza K (2017) Sentiment analysis: machine learning approach. Int J Eng Technol 9:2183–2186. <https://doi.org/10.21817/ijet/2017/v9i3/1709030151>

3. Mathapati P, Shahapurkar A, Hanabaratti K (2017) Sentiment analysis using Naïve Bayes algorithm. *Int J Comput Sci Eng* 5:75–77. <https://doi.org/10.26438/ijcse/v5i7.7577>
4. Mikolov T, Corrado GS, Chen K, Dean J (2013) Efficient estimation of word representations in vector space, pp 1–12
5. Wang C, Nulty P, Lillis D (2020) A comparative study on word embeddings in deep learning for text classification: 37–46. <https://doi.org/10.1145/3443279.3443304>
6. Al-Saqa S, Awajan A (2019) The use of Word2vec model in sentiment analysis: a survey. <https://doi.org/10.1145/3388218.3388229>
7. Cambria E, Schuller B, Liu B, Wang H, Havasi C (2013) Knowledge-Based approaches to concept-level sentiment analysis. *Intell Syst* 28:12–14. <https://doi.org/10.1109/MIS.2013.45>
8. Vizcarra J, Kozaki K, Torres-Ruiz M, Quintero R (2020) Knowledge-based sentiment analysis and visualization on social networks. *New Gener Comput* 39. <https://doi.org/10.1007/s00354-020-00103-1>
9. Asghar M, Kundi F, Khan A, Ahmad S (2014) Lexicon-based sentiment analysis in the social web. *J Basic Appl Sci Res* 4:238–248
10. Tripodi R, Pelillo M (2017) A game-theoretic approach to word sense disambiguation. *Comput Linguist* 43(1):31–70. https://doi.org/10.1162/COLI_a_00274
11. Agirre E, de Lacalle OL, Soroa A (2014) Random walks for knowledge-based word sense disambiguation. *Comput Linguist* 40(1):57–84
12. Rentoumi V, Giannakopoulos G, Karkaletsis V, Vouros GA (2009) Sentiment analysis of figurative language using a word sense disambiguation approach. In: Proceedings of the international conference RANLP-2009. Association for Computational Linguistics, Borovets, Bulgaria, pp 370–375
13. Zhong Z, Ng HT (2010) It makes sense: a wide-coverage word sense disambiguation system for free text. In: Proceedings of the ACL 2010 system demonstrations. Association for Computational Linguistics, Uppsala, Sweden, pp 78–83
14. Cernian A, Sgarciu V, Martin B (2015) Sentiment analysis from product reviews using SentiWordNet as lexical resource. In: 2015 7th international conference on electronics, computers and artificial intelligence (ECAI), Bucharest, Romania, pp WE-15-WE-18. <https://doi.org/10.1109/ECAI.2015.7301224>
15. Hung C, Lin H-K (2013) Using objective words in SentiWordNet to improve word-of-mouth sentiment classification. *IEEE Intell Syst* 28(2):47–54. <https://doi.org/10.1109/MIS.2013.1>

A Sequential News Capture and Summarization Model (SNCSM)



Narayan Jee Jha, Rishav Sinha, Sameer Kumar, and Trasha Gupta

Abstract Automatically extracting specific information of interest from daily news articles is a crucial and critical challenge. Despite the abundance of work on domain-specific news summarization, most existing methods do not selectively capture and summarize important news only. To bridge this gap, we propose a Sequential News Capture and Summarization Model (SNCSM) that can automate the process of capturing domain-specific important news in addition to the task of news summarization. Carried out in two phases, this study developed SNCSM and tested the model accuracy using a weighted accuracy technique on a Civil Services Examination (CSE) dataset. Phase-I dealt with the construction of dataset and Phase-II involved the model development and model accuracy testing. CSE syllabus, previous years QnA, and news articles were web-scraped from the Internet. Tokenization of scraped data using spaCy, data pre-processing, feature engineering, and clustering were performed to construct the dataset. Word-similarity and ML models were implemented to capture and predict the importance (weight) of tokens, respectively, hence news articles. RMSE and $R2$ score along with other two metrics were used to evaluate model performances. MLP regression resulted the least RMSE of 0.311 and highest $R2$ score of 0.537. However, average token weight predicted by the models on the Prediction Day (P-Day, 01.01.2022) dataset were used for the calculation of importance of news articles. Highly weighted P-Day ($\leq \mu + 0.25 \times \sigma$) news articles (25%) were summarized using the BERT Extractive Summarizer (BERT-ES) with the mean ROUGE-L $F1$ -score of 0.475. The SNCSM achieved a weighted accuracy, linear combination of $R2$ score and ROUGE-L $F1$ -score, of 0.566. However, further study is needed to enhance the model performance.

Keywords Tokenization · Word-similarity · BERT · spaCy · k -means clustering · ML models · News summarization

N. J. Jha · R. Sinha (✉) · S. Kumar · T. Gupta
Department of Applied Mathematics, Delhi Technological University, New Delhi, India
e-mail: kumarrishavsinha@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
A. Swaroop et al. (eds.), *Proceedings of Data Analytics and Management*, Lecture Notes in Networks and Systems 787, https://doi.org/10.1007/978-981-99-6550-2_8

85

1 Introduction

Two fundamental problems in the field of Natural Language Processing (NLP) are the problems of text summarization and document similarity.

With the exponential growth of the volume of data, technologies like text summarization have become very essential. Its feature of generating shorter versions of a longer text while keeping its original meaning intact has enabled its audience to get insights of a long text in a short time. Its application ranges from the summarization of social media posts [8] to the summarization of research papers [23] and legal documents [2]. Broadly, text summarization is of two types—(i) extractive, where a subset of important sentences is extracted from the source text, and (ii) abstractive, where a model rephrases the main idea of the source text. This study focuses on the extractive summarization of news articles, an important application of text summarization known as news summarization.

The second problem, document similarity involves measuring the degree of similarity between two or more documents based on their content and structure. Document similarity has its application in many different domains like search engines [10], recommendation systems [5], and plagiarism detection systems [21].

CSE is a highly competitive examination in India which requires wide knowledge of current affairs related to government policies, international relations, economics, geopolitics, environment, administration, and related fields. THE HINDU (TH) [11] newspaper is the main source of current affairs followed by the CSE aspirants. Everyday reading of a complete newspaper takes hours of valuable time of aspirants. And the problem of identifying, analyzing, and summarizing important news is highly time-consuming. Using SNCSM we can increase the efficiency and effectiveness of the exam preparation.

This paper exploited the power of document similarity and machine learning algorithms to capture and predict important news, respectively. For the tasks of keywords extraction from the TH news articles and keywords similarity calculation, this study used spaCy, an advanced NLP library written in Python and Cython and developed by Matthew Honnibal et al. [13]. And for news summarization, this study used BERT-ES developed by Derek Miller [7].

The rest of the paper is structured as follows: Sect. 2 discusses the related works, Sect. 3 presents the tools and technologies used, Sect. 4 explains the preparation of dataset, Sect. 5 explains the methodology used for the development of SNCSM, Sect. 6 concludes the result of the study, Sect. 7 concludes this study, and Sect. 8 mentions all the references.

2 Literature Review

While the roots of document similarity and text summarization problems can be traced back to the 1950s and 1960s, this section focuses on recent papers that relate to the present paper.

Chantrapornchai et al. [4] presented a methodology to extract information from tourism data. The dataset was constructed by scraping TripAdvisor, Traveloka, and Hotels.com. Three datasets namely Hotels, Shopping and Tourism, and Restaurant were created. Two NLP tools, BERT, and spaCy were used for the construction of models to perform named Entity Recognition and text classification tasks. There was a minor performance advantage of BERT over spaCy for both the tasks.

Singh et al. [22] identified top news articles on news websites and measured the similarity between two same news in English and Hindi languages. Top news was extracted from Google's news feed and Google Translate was used to translate Hindi news into English. To calculate the news similarity authors used Cosine similarity, Jaccard similarity, and Euclidean distance measures. While all three methods yielded good results, cosine similarity resulted highest accuracy, recall, and F -measure scores of 0.8125, 1.0, and 0.7692, respectively.

Proposed by Priyadharshan et al. [16] described a methodology to summarize Tamil sports news. This paper began with the gathering of Tamil sports news from the Internet and went through the tokenization, feature extraction, feature enhancement, and summary generation phases. The feature matrix was constructed using various features including term-frequency, inverse-document-frequency, number of named entities, and sentence position. To improve the summary quality, this paper used Restricted Boltzmann Machine to recompute the values of the feature matrix.

Batra et al. [1] published a paper that used the "COVID-19 Public Media Dataset" to compare the performances of BERT, GPT-2, XLNet, BART, and T5 models for the task of news summarization. Results were evaluated through ROGUE scores. BERT outperformed other models with the ROUGE-2 score of 0.354 and ROUGE-L score of 0.364. The authors incorporated BERT into their web application called "CoVShorts" to summarize COVID-19 news articles.

Sindhu et al. [21] presented a methodology for detecting plagiarized documents in Malayalam language. Tokenization based on TF-IDF, stop-word removal, lemmatization, and POS tagging were performed on the suspicious and possible source documents to meet the needs of comparison. The Vector Space Model was used to detect plagiarized documents at sentence level. For similarity computations, Jaccard, Cosine, and Dice, similarity coefficients were used. A Probabilistic Neural Network was used to combine the similarity scores and to classify whether the suspicious documents were plagiarized or not.

3 Tools and Technologies

In this study, a range of tools and technologies were used. These tools were carefully selected to meet the specific needs of this study.

1. spaCy: Published under the MIT license, spaCy uses Thinc (ML library) as its backend. In model building, it performed two major tasks, keywords extraction and word similarity calculation.

2. Selenium: Developed by Huggins and aimed at browser automation, Selenium was used to scrape the syllabus [24] and QnA data from the Internet.
3. Puppeteer: Developed by Google, Puppeteer is a Node.js library which provides an API to control Chrome/Chromium [3] over the DevTools Protocol. It was used to web-scrape the news articles' Uniform Resource Locator (URLs).
4. Cheerio JS: Used for the task of news articles scraping from URLs, Cheerio is a fast, flexible, and easy to use tool for parsing HTML, and XML data. It was developed by Muller et al.
5. BERT: Introduced by Google, BERT is a family of masked-language that was used to run extractive summarization of the predicted important news articles.

4 Data Description and Pre-processing

4.1 Data Gathering

The following subsections describe the sources of data and the techniques used to collect them.

1. Syllabus: The detailed syllabus for the CSE-MAIN exam was web-scraped from the official site of the UPSC by using Selenium and was stored in syllabus.txt file. Figure 1 is a sample figure of scraped data.
2. QnA: Similarly, QnA data for the years 2017–2021 were web-scraped from various authentic websites dedicated to the preparation of CSE-MAIN and was stored in qna_yyyy.txt files. Figure 2 is a sample Figure of QnA scraped data.
3. Newspaper: As the study focuses on news summarization, the daily news was scraped from the TH website from the year 2016 to 2021. Using Puppeteer, URLs of news articles were web-scraped and stored in url_ddmmyyyy.txt files. Using Cheerio, news articles were web-scraped from the stored URLs and were stored in news_ddmmyyyy.txt files. The sample figure is presented in Fig. 3. Only news tagged as “World”, “Business”, “National”, “India”, “Science”, “Technology”, “Society”, “Profile”, “Editorial”, “Economy”, “Event”, “History and Culture”, “Education”, or “Sci-Tech” were gathered. A total of 147007 and an average of 67 news articles per day were collected. The sample figure is presented in Fig. 4.

4.2 Tokenization

In this study, tokens labeled as “PERSON”, “GPE”, “NORP”, “FAC”, “LOC”, “EVENT”, “WORK_OF_ART”, “LANGUAGE”, “LAW”, “ORG”, and “PRODUCT” were considered, while tokens labeled as “DATE”, “TIME”, “PERCENT”,

```

Indian Culture - Salient aspects of Art Forms, Literature and Architecture from ancient to modern times
Indian Art Forms
Indian Paintings
Mural Paintings
Miniature Paintings
Mughal, Rajput, Pahari Paintings

```

Fig. 1 Syllabus

```

How do you justify the view that the level of excellence of the Gupta numismatic art is not at all noticeable in later times?
Gupta coins are among the most remarkably detailed coins from ancient India featuring exquisite artistic details. They stand out from coins crafted during the reign of other dynasties on following counts:
Achievement of remarkable craftsmanship was evident by the fineness of the variety of images carved on both faces of the coin and its smooth and even edges. Detailed carvings ranged from Chandragupta riding a horse to Samudragupta playing a Veena and the Goddess Lakshmi to a sacrificial horse for Ashvamedha ceremony and so on.
Apart from these detailed images, there were inscriptions as well, often adding details of the image inscribed on the coin.
Also, most of the important kings of Gupta dynasty are now believed to have had multiple coin-types during the course of their reign, in which older designs were dropped and newer motifs were adopted.
Scholars have pointed out that such designs were made possible by use of clay-molds by skilled mint-masters.
In the post-Gupta period, not only the quality of gold coins fell, but also the numbers of gold coins being issued dropped drastically. Due to systemic economic distresses, town-based artisans producing good quality coins were forced to migrate to the countryside causing a decline of craft production and a decay of townships. Without urban centres and foreign trade, and with increasing decentralisation of political power which had resulted from the urban to rural migration of artisans, the excellence of Gupta numismatic art could not be sustained or recreated in the times that followed.

```

Fig. 2 qna_2017

```

https://www.thehindu.com/news/international/Turkey%E2%80%99s-dangerous-war-on-Kurdish-militants/article13975225.ece
https://www.thehindu.com/news/international/Gun-friendly-Texas-becomes-more-trigger-happy/article13975223.ece
https://www.thehindu.com/news/international/Pak.-Army-chief-confirms-death-sentence-to-9/article13975221.ece
https://www.thehindu.com/business/Economy/RBI-tells-banks-to-replace-defective-1000-rupee-notes/article13975209.ece
https://www.thehindu.com/business/Economy/Government-extends-tax-residency-rule-deadline/article13975207.ece
https://www.thehindu.com/business/Economy/China%E2%80%99s-December-factory-activity-shrinks/article13975206.ece
https://www.thehindu.com/news/national/India-Pakistan-exchange-list-of-nuke-installations-prisoners/article13975185.ece
https://www.thehindu.com/news/international/Explosion-rocks-Kabul/article13975172.ece
https://www.thehindu.com/news/international/2016-will-be-a-game-changer-for-UK-Cameron/article13975168.ece
https://www.thehindu.com/news/international/Two-killed-several-injured-in-Tel-Aviv-bar-attack-police/article13975181.ece

```

Fig. 3 qna_2017

```

China's December factory activity shrinks
China looked set for a soggy start to 2016 after activity in the manufacturing sector contracted for a fifth straight month in December, suggesting the government may have to step up policy support to avert a sharper slowdown, while China's services sector ended 2015 on a strong note, the economy still looked set to grow at its slowest pace in a quarter of a century despite a raft of policy easing steps, including repeated interest rate cuts, in the past year or so. The world's second-largest economy faces persistent risks this year as leaders have pledged to push so-called "supply-side reforms" to reduce excess factory capacity and high debt levels. The official manufacturing Purchasing Managers' Index (PMI) stood at 49.7 in December, in line with expectations of economists polled by Reuters and up only fractionally from November. A reading below 50 suggests a contraction in activity while a higher one indicates an expansion. Still, economists seemed to find some comfort that there were no signs of a sharper deterioration which has been feared by global investors. The slight uptick up in the manufacturing PMI "suggests that (economic) growth momentum is stabilising somewhat ... however, the sector is still facing strong headwinds," said Zhou Hao, China economist at Commerzbank in Singapore. "In order to facilitate the destocking and deleveraging process, monetary policy will remain accommodative and the fiscal policy will be more proactive." Weak demand from at home and abroad has weighed on China's factories, exacerbating the problem of excess capacity and forcing them to cut prices of their goods, eating into their profits and adding to deflationary pressures in the economy. Total new orders - a proxy for both domestic and foreign demand - rose to 50.2 in December from November's 49.8, the PMI survey showed. But export orders shrank for the 15th straight month, albeit at a less severe pace. The sub-index inched up to 47.5 from November's 46.4. The National Bureau of Statistics (NBS) said that although oil prices were very low at present, cash at the end of the year was tight for factories, putting relatively large pressure on manufacturers. A challenging 2016 China's economic growth is expected to cool from 7.3 per cent in 2014 to 6.9 per cent in 2015, the central bank said in a recent work paper, its slowest pace in 25 years. It said growth could ease further to 6.8 per cent in 2016. Indeed, China could run its biggest budget deficit in half a century this year as leaders turn to more government spending to arrest the slowdown in the economy, policy advisers say, after disappointing returns from a year of policy easing. The PBOC has cut interest rates six times since November 2014 and reduced banks' reserve requirement ratios (RRR), on the amount of cash that banks must set aside as reserves. The government has also stepped up spending on infrastructure projects and eased restrictions on home buying to boost the sluggish property market. The central bank is widely expected to cut interest rates and banks' reserve requirement ratios further this year. A similar official survey on the services sector showed activity there quickened in December, again helping to ease fears of a hard landing for the economy this year. The services sector has been the lone bright spot in the economy in the last few years, helping to offset prolonged weakness in the vast manufacturing sector, though financial markets tend to focus more closely on factory readings. The official non-manufacturing Purchasing Managers' Index (PMI) rose to 54.4, from November's 53.6, according to the NBS. The services sector has accounted for the bigger part of China's economic output for at least two years. A private gauge of Chinese manufacturing called Markit PMI, which focuses more on small-to-medium-sized private firms, will be released on January four. China is set to release fourth quarter and full-year GDP data on January 19.

```

Fig. 4 News articles

“MONEY”, “QUANTITY”, “ORDINAL”, and “CARDINAL” were excluded in the tokenization [15] process.

1. Syllabus: spaCy was used to perform tokenization on syllabus.txt file. A total of 400 tokens were extracted with “Type” as 2nd feature. And, extracted data was stored in DictSyll.csv file. Figure 5 is a sample Figure of DictSyll.csv.
2. QnA: Again, spaCy was used to perform tokenization of all five qna_YYYY.txt files which yielded a total of 2666 tokens. A dataset consisting of three attributes,

	A	B
1	Keyword	Type
2	Kushana	PERSON
3	Satavahanas	PERSON
4	Chandragupta	PERSON
5	Sufi	NORP
6	Urbanisation	LAW
7	Women and Women's Organization	ORG
8	Naxalism	EVENT
9	Industrial Revolution, world wars	EVENT
10	India Movement	EVENT
11	Nationalist Upsurge Post-World War II	EVENT
12	Wavell Plan	EVENT
13	Indian National Army	ORG
14	Independence	EVENT
15	Cabinet Mission	EVENT
16	Subhash Chandra Bose	PERSON
17	United States	GPE
18	Civil War	EVENT
19	Rajputs	NORP

Fig. 5 DictSyll

	A	B	C
1	Keyword	Frequency	Type
2	Gupta	5	PERSON
3	India	190	GPE
4	Chandragupta	1	PERSON
5	Samudragupta	1	PERSON
6	Veena	1	PRODUCT
7	the Goddess Lakshmi	1	WORK_OF_ART
8	Ashvamedha	1	PERSON
9	Mughal Empire	2	GPE
10	Mughal	3	NORP
11	Bengal	4	GPE
12	Avadh	1	GPE
13	Hyderabad	3	GPE
14	Mysore	1	GPE
15	the Mughal Empire	2	GPE
16	Marathas	3	PERSON
17	Nadir Shah	1	PERSON
18	Persia	1	GPE
19	Afghan	1	NORP
20	Ahmad Shah Abdali	1	PERSON

Fig. 6 DictQnA_2017

namely “Keywords”, “Frequency”, and “Type”, has been gathered and was stored in DictQnA_YYYY.csv files. DictQnA_2017 sample data is attached in Fig. 6.

3. Newspaper: Once again, spaCy was used to perform tokenization of news_ddmmyyyy.txt files. On an average, 900 tokens/day were extracted. We collected two attributes, “Date” and “Keywords + Frequency”, in DictNews_YYYY.csv files. A sample data is presented in Fig. 7.

In addition of DictNews_YYYY.csv, keywords with their respective category were stored in DictType_YYYY.csv files. Figure 8 is a sample of DictType_2017.csv.

Date	Keyword_n_frequency
15-6-2016	{'Ayurvedic', [3], 'Maharashtra', [2], 'Pune', [2], 'Balaji Tambe', [1], 'Pre-Natal Diagnostic Techniques', [1], 'Judicial', [1], 'Sangamner', [2], 'Ahmednagar', [1], 'Tambe', [5], 'Asurve
16-6-2016	{'British', [4], 'EU', [15], 'United Kingdom', [1], 'Parliament for Bately and Spen', [1], 'Jo Cox', [3], 'Birstall', [1], 'Leeds', [1], 'West Yorkshire', [1], 'Britain', [10], 'The European Union
17-6-2016	{'Iraq', [2], 'Falujah', [5], 'Islamic', [9], 'Iraqi', [1], 'State', [20], 'Abdulwahab al-Saadi', [1], 'AFP_Raed Shaker Jawdat', [1], 'Mosul', [1], 'The Supreme Court', [1], 'Special Investigati
18-6-2016	{'Subramanian Swamy', [4], 'Chidambaram', [1], 'Jaitley', [6], 'Rajan', [40], 'RBI', [30], 'Raghuram Rajan', [9], 'Chicago', [2], 'BIP', [26], 'The Reserve Bank', [2], 'The Government of I
19-6-2016	{'Arun Jaitley', [1], 'Congress', [11], 'Amarinder Singh', [2], 'The Am Aadmi Party', [1], 'Delhi', [2], 'BIP', [14], 'Amarinder', [3], 'The Enforcement Directorate and Income Tax departm
20-6-2016	{'Mombasa', [1], 'Kabul', [9], 'Afghan', [6], 'Reuters', [1], 'Nepales', [3], 'AFP', [2], 'Justice ministry', [1], 'Ghazni', [1], 'Turkey', [2], 'Palmyra', [1], 'Damascus', [6], 'Khat Linear', [1]
21-6-2016	{'US', [5], 'India', [84], 'NSG', [45], 'The Nuclear Supplier Group', [1], 'Seoul', [9], 'State Department', [1], 'John Kirby', [1], 'Chinese', [5], 'Beijing', [1], 'Kirby', [2], 'PM Modi's', [1],
22-6-2016	{'UK', [5], 'BIO Agency', [4], 'BIO', [6], 'India', [24], 'Tech Mahindra', [5], 'Europe', [1], 'US', [6], 'Digital Change', [1], 'Mumbai', [2], 'Tech Mahindra's', [1], 'Digital Transformator
23-6-2016	{'Narendra Modi', [9], 'The Shiv Sena', [2], 'Pakistan', [36], 'Modi', [11], 'Shivaasan', [1], 'Sena', [1], 'Saamana', [The Sena', [1], 'BSP', [4], 'Maharashtra', [1], 'India', [89], 'Army', [
24-6-2016	{'Census', [2], 'Nitish Kumar', [n]nBihar', [1], 'Nitish Kumar', [1], 'The Census of India', [1], 'Socio Economic Caste Census', [1], 'ST', [1], 'Kumar', [3], 'Patna', [2], 'The Asian Developme
25-6-2016	{'Russian', [7], 'Putin', [7], 'Chinese', [7], 'Vladimir Putin', [1], 'China', [21], 'Russia', [12], 'Beijing', [6], 'U Keqiang', [1], 'Fu', [1], 'The Great Hall of the People', [1], 'Communism', [1]
26-6-2016	{'AI', [1], 'Ken Forbus', [1], 'Northwestern University', [1], 'U.S.', [42], 'The Structure Mapping Engine', [1], 'Cognitive Science', [1], 'SMF', [5], 'CogSketch', [1], 'Dr Forbus', [1], 'Dedf
27-6-2016	{'Christian', [1], 'Pakistan', [23], 'Muslim', [2], 'Haji Park Tajpora', [1], 'Lahore', [2], 'Badaf', [3], 'Wasef Nasser', [1], 'Muhammad Naveed', [1], 'Samra', [2], 'Shagufa', [1], 'Naseer
28-6-2016	{'Arun Jaitley', [2], 'Jaitley', [1], 'Commerce', [1], 'Nirmala Sitharaman', [1], 'Coaf', [1], 'Piyush Goyal', [1], 'Dharmendra Pradhan', [1], 'Titendra Singh', [1], 'The Income Tax Departm
29-6-2016	{'The Gujarat High Court', [1], 'post-Godhra', [1], 'Viramgam', [1], 'Ahmedabad', [3], 'V5', [1], 'High Court', [4], 'Harsha Devani', [1], 'Biren Vaishnav', [1], 'Gujarat High Court', [1], 'L
30-6-2016	{'Pakistan', [20], 'India', [89], 'Mumbai', [6], 'LeT', [1], 'Indian', [18], 'Pakistan Foreign Office', [1], 'Nafees Zakaria', [1], 'Zakaria', [2], 'Pakistan', [1], 'Lashkar-e-Taiba', [2], 'Lakhyi
1/7/2016	{'Apple', [3], 'Phones', [1], 'US', [3], 'California', [1], 'Adele', [1], 'selfie', [2], 'Dhaka', [28], 'Islamic', [9], 'updates.10.40', [1], 'Bangladesh', [2], 'Tuhin Mohammad Masud', [1], 'I

Fig. 7 DictNews_2017

16	Hindu @# NORP
17	Kairana @# PERSON
18	Uttar Pradesh's Shamli @# ORG
19	BSP @# ORG
20	Mayawati @# PERSON
21	BJP @# ORG
22	Muslims @# NORP
23	Hindus @# NORP
24	Rajya Sabha @# ORG
25	MLC @# ORG

Fig. 8 DictType_2017

4.3 Data Pre-processing

An important semi-automated step, in model development was data pre-processing to handle irregularities and errors.

1. Syllabus and QnA: Some tokens were given wrong type which was corrected manually. This step resulted DictSyll.csv + DictQnA.csv = DatasetY.csv.
2. Newspaper: For CSE-MAIN 2017, Dataset_2017 of unique tokens of the past 500 days (an arbitrary number) from the date of examination was constructed from DictNews_yyyy.csv and DictType_yyyy.csv files. Other features were “Category”, “FreqDay1”, “FreqDay2”, ..., “FreqDay500”, and “TotalFreq”. Sample is shown in Fig. 9.

A total of 105,967 unique tokens constituted the dataset which was later transformed into 8298 tokens by dropping all the tokens having “TotalFreq” < 10. Similar procedure was performed for CSE-MAIN 2018, 2019, 2020, and 2021 which generated datasets having same features and 7940, 7968, 10,568, 10,769 unique tokens, respectively. The datasets obtained was stored in Dataset_yyyy.csv files. While csv file writing, spaCy was unable to detect some of the special characters which made tokens irrelevant. All such irrelevant tokens were identified and dropped. Sometimes, different letter-case of same word were counted inaccurately which was solved by bringing the tokens to a common case, and adding the frequencies of duplicates, if any.

Keywords	Type	15-6-2016	16-6-2016	17-6-2016	18-6-2016	19-6-2016	20-6-2016	21-6-2016	22-6-2016	23-6-2016	24-6-2016	25-6-2016	26-6-2016	27-6-2016	28-6-2016	29-6-2016	30-6-2016	1/7/2016
INDIA	GPE	41	90	74	50	35	67	84	104	89	165	30	93	92	43	44	89	37
INDIAN	NORP	19	37	21	41	15	17	19	24	10	27	13	7	15	5	23	18	16
US	GPE	14	43	29	25	14	8	18	9	21	11	11	42	27	7	9	25	22
TRUMP	PERSON	34	21	3	18	0	0	1	12	0	5	26	0	0	4	14	3	1
CHINA	GPE	5	42	3	2	36	17	19	19	46	57	21	14	39	17	9	12	21
STATE	ORG	28	10	20	33	7	2	5	6	7	0	6	16	15	8	14	23	16
BIP	ORG	7	19	17	26	14	4	0	16	4	0	4	21	14	10	17	8	12
CONGRES	ORG	8	27	20	19	11	2	4	11	5	0	9	14	15	8	8	20	3
PAKISTAN	GPE	3	21	1	23	12	11	27	21	1	3	11	4	3	3	1	1	31
HINDU	NORP	15	20	13	13	10	14	1	13	10	10	3	9	8	17	17	14	20
KASHMIR	LOC	6	3	1	2	0	0	0	7	2	0	2	2	3	4	4	12	0
CHINESE	NORP	2	18	2	0	15	11	5	5	14	12	7	2	9	14	1	1	4

Fig. 9 Dataset_2017

4.4 Making of Final Dataset

The steps of construction of final dataset is metioned in this subsection.

1. **Weighted Frequency Calculation:** This study focused on the news articles of 500 days prior to the day of yearly CSE-MAIN. Concept of weighted frequency was introduced to assign higher weight to the recent news and lower weight to the older news. The idea of weighted frequency was implemented through the sine function, since

$$\int_0^{\pi/2} \sin(x) dx = 1$$

as shown in Fig. 10. The shaded area was divided into 500 parts and a list of 500 corresponding constants were produced by the formula:

$$A_i \equiv \int_{\frac{(i-1)\times\pi}{1000}}^{\frac{i\times\pi}{1000}} \sin(x) dx$$

where, $i = 1, 2, \dots, 500$ Then, the constants and frequencies of Dataset_yyyy.csv were multiplied correspondingly to calculate date-wise weighted frequency. Sum of date-wise weighted frequency introduced a new feature, “Weighted-Frequency”, into the Dataset_yyyy.csv files. The files were later renamed as DatasetX_yyyy.csv.

2. **Weight Computation:** A new feature named “Weight” was introduced into the DatasetX_yyyy.csv with the help of spaCy’s word similarity feature. “Weight” computation was achieved by mapping the tokens from DatasetX_yyyy.csv to the tokens from DatasetY.csv. “Weight” computation for all the tokens of DatasetX_yyyy.csv was done by using the following two formulas:

$$WS(K_i) = \max(\text{similarity}_{(\forall K_j \in \text{DatasetY.csv})}(K_i, K_j))$$

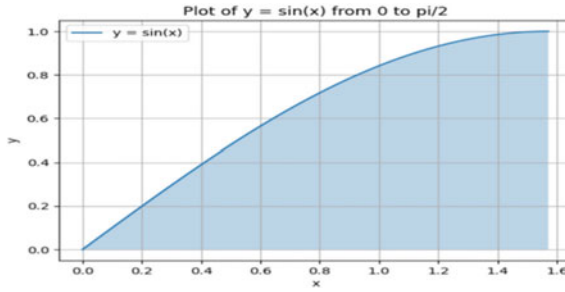


Fig. 10 Integration of $\sin(x)$ over 0 to $\pi/2$

$$\text{Weight}(K_i) = \text{WS}(K_i) \times \text{DatasetY}[\text{Frequency}][K_i]$$

where, similarity (K_i, K_j) was computed by spaCy and $\max()$ is the maximum function.

Since DictSyll.csv did not have the “Frequency” feature, the average categorical frequency of DictQnA.csv were used for the “Weight” computations. Figure 11 is a sample of dataset after above computations.

3. Merging of Datasets: All five datasets (DatasetX_2017 to DatasetX_2021) were merged into a single MergedDataset.csv file. For the words present in more than one dataset, average pooling was done for the “WeightedFrequency” and “Weight” attributes. A new attribute named “MergeCount”, count of datasets in which a word was present, was added (Fig. 12).

Merging process resulted 18,080 unique tokens whose sample Figure is attached below.

4. Clustering: Clustering was used to divide the population into collection of objects based on similarity and dissimilarity between them. Specifically, k -Means method [9] for $K = 2$ to $K = 9$ was used to perform the clustering. Along with it, we used elbow and Silhouette coefficient [19] methods to get optimal value of K . Plots shown in Fig. 13 were received. Best silhouette value was at $K = 3$.

After visualization, we found that some organizations got higher “Weight” than expected. This happened because of token having “INDIA” in their name and due to the working of spaCy to find similarity between tokens, got them higher “Weight”. This was corrected manually to avoid the outlier data. Lastly, plots shown in Fig. 14 were received. Best silhouette value was at $K = 4$.

Therefore, taking $K = 4$ as optimal K , a new attribute named “Clusters” was introduced into the MergedDataset.csv which was renamed as FinalDataset.csv, as shown in Fig. 15.

Keyword	Category	Wt. Frequency	Weight
INDIA	GPE	64.14040975	190
US	GPE	33.37486162	1
INDIAN	NORP	23.87266663	59
TRUMP	PERSON	21.6571627	1.443070229
CHINA	GPE	19.29504958	17
STATE	ORG	14.87365958	5
BIP	ORG	14.49628807	2.033752911
CONGRESS	ORG	13.94957393	5.63490673
PAKISTAN	GPE	12.04731663	2
HINDU	NORP	8.49640712	2.1937094
CHINESE	NORP	8.341275527	0.868087879
THE SUPREME COURT	ORG	8.146995138	11
MODI	PERSON	7.643345108	2.442350122
DELHI	GPE	7.1787741	2

Fig. 11 Dataset_2017_new

Keyword	Category	Count	Wt. Frequency	Weight
10 DOWNING STREET	FAC	3	0.014397747	0.410609152
5TH FLEET	ORG	3	0.01606538	0.334946521
A CONSTITUTION BENCH	ORG	5	0.120759641	0.861047361
A HIGH COURT	ORG	5	0.02977064	0.826965448
A LOK SABHA	PERSON	4	0.02553925	0.847602796
A MEMORANDUM OF UNDERSTANDING MOU	EVENT	1	0.00591048	0.173102457
A MINISTRY	ORG	1	0.004003317	0.127825485
A MONEY BILL	PRODUCT	3	0.027422608	0.430270929
A NOBEL PRIZE	WORK_OF_ART	1	0.005469904	0.138384893
A PUBLIC INTEREST LITIGATION	ORG	2	0.009252509	0.353309431
A UNIFORM CIVIL CODE	ORG	1	0.002306379	0.153888166
A320	LAW	3	0.02556955	0.329551498
AADHAAR	PRODUCT	5	2.389468058	1.244747746
AADHAAR ACT	LAW	3	0.0455391	1.573246359
AADHAAR CARD	PERSON	1	0.005188124	0.171206146
AADHAAR PAN	ORG	2	0.00805603	0.440693199
AADHAAR	PRODUCT	5	0.087972789	1.07574234

Fig. 12 MergedDataset

5 Methodology: SNCSM

5.1 Data Analysis and Visualization

FinalDataset.csv consisted of 18,080 rows and 6 columns out of which 3 columns (“MergeCount”, “WeightedFrequency”, and “Weight”) were numeric and other 3 columns (“Keyword”, “Category”, and “Clusters”) were categorical in nature. Keyword “INDIA” had the highest value of “WeightedFrequency” (74.71) and “Weight” (192.6).

99.96% data points were present in cluster A which made the attribute of no use for the model development. 86.46% data points belonged to three major categories, “ORG”, “PERSON”, and “GPE” as shown in Fig. 16.

8109 data points had “MergeCount” = 1 and 3253 data points had “MergeCount” = 5. Figure 17 is a histogram visualizing “MergeCount”.

Figure 18 is a scatterplot between “WeightedFrequency” and “Weight”. All the data points lie in 1st quadrant only and are condensed near the origin. For the outliers removal, observable in Fig. 18, this study considered data points having:

$$0 < \text{Weight} < \mu_{\text{Weight}} + 2 \times \sigma_{\text{Weight}}, \text{ and}$$

$$\text{WF} < \mu_{\text{WF}} + 2 \times \sigma_{\text{WF}}$$

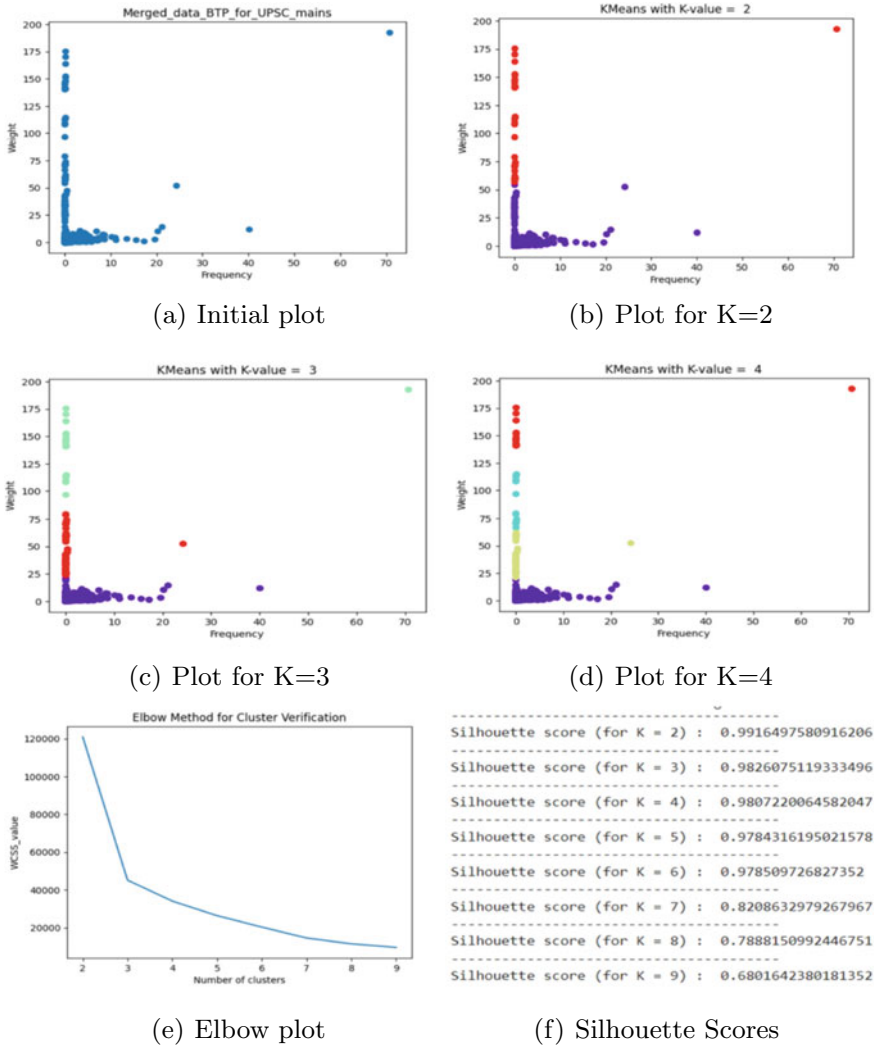
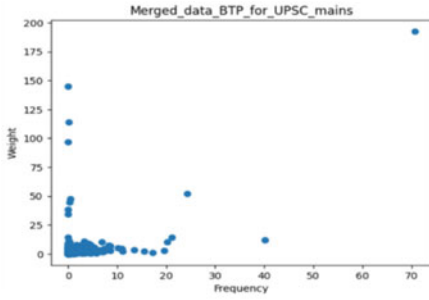


Fig. 13 Old clustering

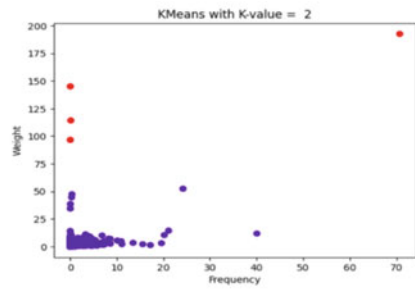
where WF stands for “WeightedFrequency”. On applying the above condition, 7.97% data points got eliminated and the dataset was left with 16,638 data points.

Figure 19 presents the heatmap of the correlation matrix after outliers’ removal. Pearson Correlation coefficient of “Weight” with “WeightedFrequency” and “Merge-Count” were 0.44 and 0.71, respectively.

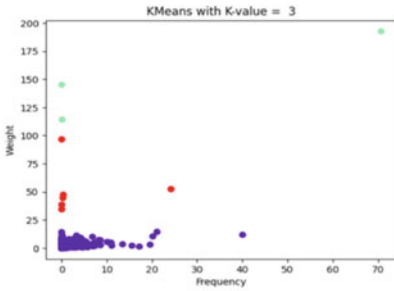
Next steps included categorical data handling, splitting of dataset and feature scaling. One-Hot Encoding was done for the “Category” feature, while other categorical features were dropped as they were not important for the model. Next, dataset was



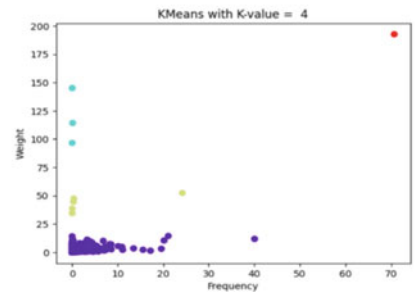
(a) Initial plot



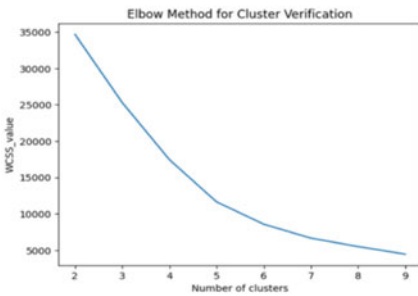
(b) Plot for K=2



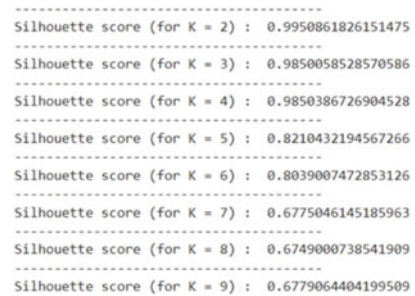
(c) Plot for K=3



(d) Plot for K=4



(e) Elbow plot



(f) Silhouette Scores

Fig. 14 New clustering

partitioned into two datasets, training (80%) and testing (20%), using train_test_split function of the scikit-learn library. At the end, feature scaling was performed through StandardScaler function of the scikit-learn library which finalized the dataset for the ML models.

1	Keyword	Category	Count	Wt.	Frequency	Weight	Clusters
2	INDIA	GPE	5	70.71491677		192.6	D
3	DELOITTE INDIA	ORG	5	0.062971955		0.994897	A
4	EY INDIA	ORG	5	0.052135512		0.689564	A
5	AMAZON INDIA	ORG	5	0.059613484		1.806735	A
6	COAL INDIA	PRODUCT	5	0.07780332		4.059687	A
7	MARUTI SUZUKI INDIA	ORG	5	0.062332298		0.900201	A
8	INDIA GATE	FAC	5	0.069760044		2.093748	A
9	AIRASIA INDIA	ORG	4	0.031246412		1.996454	A
10	HINDUSTAN	ORG	5	0.02862035	145.1041511		B
11	PNV INDIA	ORG	4	0.04149944		1.445757	A
12	INDIGO	ORG	5	0.15366909		1.218675	A
13	INDIA POST	ORG	5	0.035294581		2.769657	A
14	INDIA DIPAK MISRA	PERSON	3	0.143831978		0.497748	A
15	US INDIA	GPE	5	0.127474209	114.241119		B
16	AUDI INDIA	ORG	4	0.020301942		0.356467	A
17	QUIT INDIA	EVENT	3	0.033201105		0.887682	A
18	SWARAJ INDIA	EVENT	3	0.014491615		0.457662	A
19	NORTHEAST INDIA	GPE	5	0.022694342	96.62081636		B
20	GRANULES INDIA	PRODUCT	2	0.010735881		0.340164	A

Fig. 15 FinalDataset

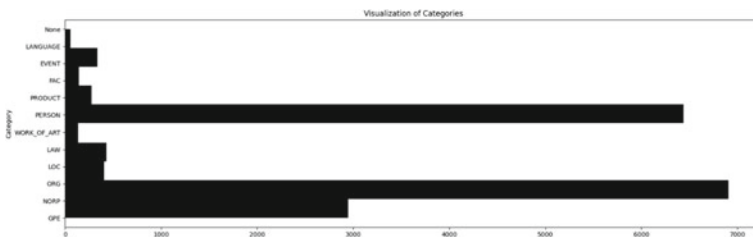


Fig. 16 Histogram visualizing categories

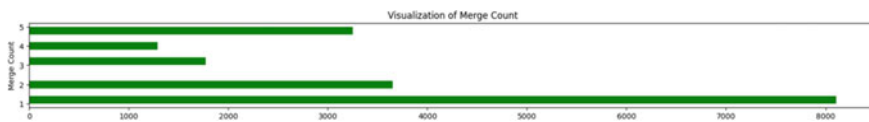


Fig. 17 Histogram visualizing MergeCounts

Fig. 18 A scatterplot between WeightedFrequency and weight



Fig. 19 Heatmap of correlation matrix

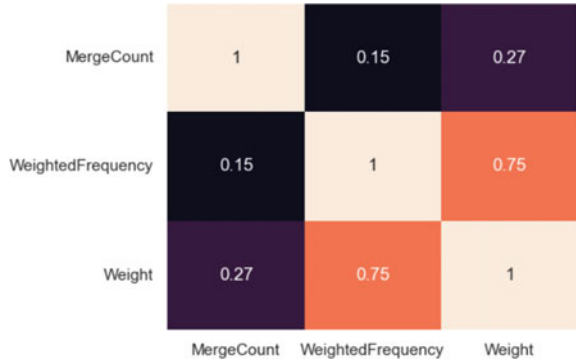


Table 1 MAE and RMSE

Models	MAE	RMSE
LR	0.188	0.313
MLP	0.188	0.311
LSVR	0.174	0.325
RFR	0.218	0.366
DTR	0.251	0.429

5.2 ML Models

After feature scaling, final dataset was made up of “Weight” as the output feature, to be predicted, and input features namely “Category” (One-Hot Encoded), “Merge-Count”, and “WeightedFrequency”.

30 ML and Deep Learning (DL) models were implemented, trained, and tested on the dataset finalized in the last subsection. Based on R -squared score, models were categorized into 4 categories (R^2 score = 0.53 (Cat1), 0.49 (Cat2), 0.36 (Cat3), and 0.12 (Cat4)).

Finally, five models, LR and MLPR [18] from Cat1, LSVR [20] from Cat2, RFR [12] from Cat3, and DTR [17] from Cat4, were selected.

Table 1 lists the results of MAE and RMSE on the testing data used for the evaluation of the selected five models. Similarly, Table 2 lists the results of R^2 score and adjusted R^2 score.

5.3 News Capturing

A separate dataset was created for the P-Day by following the same methodology describe in this section. “Cluster” feature was not considered because the feature was

Table 2 R^2 score and adjusted R^2 score

Models	R^2 score	Adj. R^2 score
LR	0.531	0.529
MLP	0.537	0.535
LSVR	0.495	0.493
RFR	0.361	0.358
DTR	0.121	0.117

Keyword	Category	Count	Wt. Frequency
INDIA	GPE	6	71.99486734
US	GPE	6	41.31109271
INDIAN	NORP	6	24.02913802
STATE	ORG	6	20.69870933
CHINA	GPE	6	20.33885553
COVID 19	ORG	3	19.87569398
CONGRESS	ORG	6	19.42511198
TRUMP	PERSON	6	15.66050574
BJP	ORG	6	15.29824066

Fig. 20 P-Day dataset

Keyword	Category	Count	Wt. Frequency	LRWeight	DTRWeight	RFRWeight	LSVRWeight	MLPWeight	AverageWeight
INDIA	GPE	6	71.99486734	2.643281672	0.899317875	0.919689215	2.548329729	2.266598534	1.855434025
US	GPE	6	41.31109271	1.926479913	0.899317875	0.919689215	1.786121835	1.639912998	1.434304367
INDIAN	NORP	6	24.02913802	1.522714001	0.924696553	1.033931411	1.377482128	1.431572043	1.258079227
STATE	ORG	6	20.69870933	1.32914876	0.528738956	0.526322229	1.276124164	1.083695405	0.948805923
CHINA	GPE	6	20.33885553	1.436548779	0.899317875	0.919689215	1.265155789	1.273429267	1.15828185
COVID 19	ORG	3	19.87569398	0.76613949	0.185697158	0.151562845	0.743148853	0.60267613	0.498493887
CONGRESS	ORG	6	19.42511198	1.299996131	0.528738956	0.526322229	1.244487034	1.058616123	0.911512147
TRUMP	PERSON	6	15.66050574	1.152733858	0.397459521	0.437845758	1.140316103	0.841444683	0.793959985
BJP	ORG	6	15.29824066	1.202988745	0.528738956	0.526322229	1.141972466	0.982817844	0.87656806

Fig. 21 Prediction by the models

removed as discussed in the Data Analysis and Visualization subsection. Figure 20 is a sample of the P-Day dataset.

Five ML models trained and tested in the last subsection were used to predict “Weight” attribute of the P-Day dataset. Figure 21 presents the weight predicted by the models and the average of the predicted weights, named as “AverageWeight”.

Weight of a news article was calculated by adding weights of tokens, as shown in Fig. 23, extracted from the article for the construction of P-Day dataset. Since “WeightedFrequency” of a token was calculated by using its total daily frequency, therefore the predicted “AverageWeight” was allocated to news articles in proportion to the token’s frequency in the articles. A new dataset named Articles.csv was constructed having five attributes namely “Keywords”, “Frequency”, “ArticleNo”, “PDayFreq”, and “Weight” which is presented in Fig. 22.

1	Keyword	Frequency	ArticleNo	PDayFreq	Weight
2	Taiwan	3	1	17	0.133557908
3	China	2	1	44	0.052674008
4	Tsai Ing-wen	1	1	1	0.712259508
5	Beijing	1	1	5	0.196968725
6	Tsai	4	1	4	0.816533727
7	New Year's	1	1	9	0.100790206
8	Chinese	1	1	20	0.054285841
9	Xi Jinping	1	1	2	0.588903432
10	New Year	1	1	9	0.100790206
11	the Taiwan Strait	1	1	1	0.577149738
12	European Union	1	2	1	0.933211868
13	EU	2	2	6	0.233249513
14	Commission	1	2	1	0.809364772

Fig. 22 Article

Fig. 23 ArticleWeight

1	ArticleNo	WeightScore
2	Article 1	3.333913299
3	Article 2	1.975826153
4	Article 3	7.2413641
5	Article 4	3.819533977
6	Article 5	0.479869121
7	Article 6	2.79979403
8	Article 7	2.961689342
9	Article 8	10.73614315
10	Article 9	0.810186816
11	Article 10	5.144086544
12	Article 11	3.546783041
13	Article 12	0.904959624
14	Article 13	4.772656775
15	Article 14	20.27703434
16	Article 15	2.604635206
17	Article 16	2.877535741

5.4 News Summarization

Highly weighted news articles, as calculated in previous subsection, were summarized using BERT-ES. The articles with:

$$\text{WeightScore} \leq \mu_{\text{WeightScore}} + (0.25 \times \sigma_{\text{WeightScore}})$$

were considered important. After applying the above condition, 17 articles were captured as important from the P-Day, which was 25% of the total articles. ROUGE-L [14] *F1*-Score was used to evaluate the summarization of all the 17 articles. ROUGE-L *F1*-Score ranged from 0.387 to 0.703 with the mean value of 0.475.

6 SNCSM Accuracy

The model we developed is a sequential model of two stages, important news capturing and summarization. To calculate the final accuracy, given by the following formula, of our model we used a weighted accuracy calculation technique that used the accuracy of both the stages. For the accuracy of 1st stage, we considered average R^2 score [6] of our models and for the accuracy of 2nd stage we considered average ROUGE-L *F1*-Score.

$$\text{Accuracy} = \left(\text{Weight of 1st Stage} \times \frac{(\text{Average } R^2 \text{ Score})}{(\text{Ideal } R^2 \text{ Score})} \right) + \left(\text{Weight of 2nd Stage} \times \frac{(\text{Average ROUGE-L } F1\text{-Score})}{(\text{Ideal ROUGE-L } F1\text{-Score})} \right)$$

Considering both the stages equally weighted, taking ideal R^2 score = 0.9 and ideal Rouge-L $F1$ -score = 0.7, we got the final accuracy of our model = 0.566.

7 Conclusion

This paper presented a methodology used for the development of SNCSM and computed the accuracy score of the model on the CSE dataset.

First, we constructed our dataset by crawling the TH website and other sources. Then a series of various tasks including pre-processing and clustering were performed to construct the dataset.

Five ML models, LR, MLPR, LSVR, RFR, and DTR, were trained and tested on the dataset. MAE, RMSE, R^2 score and adjusted R^2 score were used to evaluate the models. Average of the “Weights” predicted by the models were used for the development of our SNCSM.

Highly weighted top 17 articles were summarized using the BERT-ES with the average ROUGE-L $F1$ -Score of 0.475. Finally, we calculated the weighted accuracy of our model by combining the average R^2 score of ML models and average ROUGE-L $F1$ -Score of news summarization which resulted the weighted accuracy of 0.566.

References

1. Batra H, Jain A, Bisht G, Srivastava K, Bharadwaj M, Bajaj D, Bharti U (2021) Covshorts: news summarization application based on deep NLP transformers for SARS-CoV-2. In: 2021 9th international conference on reliability, Infocom technologies and optimization (trends and future directions) (ICRITO). IEEE, pp 1–6
2. Bhattacharya P, Poddar S, Rudra K, Ghosh K, Ghosh S (2021) Incorporating domain knowledge for extractive summarization of legal case documents. In: Proceedings of the eighteenth international conference on artificial intelligence and law, pp 22–31
3. Bidelman E (2019) Getting started with headless chrome. <https://developers.google.com/web/updates/2017/04/headless-chrome>
4. Chantrapornchai Chantana, Tunsakul Aphisit (2021) Information extraction on tourism domain using spacy and BERT. ECTI Trans Comput Inf Technol 15(1):108–122
5. Chen C-M, Tsai M-F, Liu J-Y, Yang Y-H (2013) Music recommendation based on multiple contextual similarity information. In: 2013 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT), vol 1. IEEE, pp 65–72
6. Chicco D, Warrens MJ, Jurman G (2021) The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci 7:e623

7. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
8. Gillani M, Ilyas MU, Saleh S, Alowibdi JS, Aljohani N, Alotaibi FS (2017) Post summarization of microblogs of sporting events. In: Proceedings of the 26th international conference on World Wide Web companion, pp 59–68
9. Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. *J R Stat Soc Ser C (Appl Stat)* 28(1):100–108
10. Haveliwala TH, Gionis A, Klein D, Indyk P (2002) Evaluating strategies for similarity search on the web. In: Proceedings of the 11th international conference on World Wide Web, pp 432–442
11. The Hindu. Newspaper data source. <https://www.thehindu.com/archive/>
12. Ho TK (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol 1. IEEE, pp 278–282
13. Matthew Honnibal (2015). Introducing spacy. <https://explosion.ai/blog/introducing-spacy>
14. Lin C-Y (2004) Rouge: a package for automatic evaluation of summaries. In: Text summarization branches out, pp 74–81
15. Loper E, Bird S (2002) NLTK: the natural language toolkit. arXiv preprint [https://arxiv.org/cs/0205028](https://arxiv.org/abs/cs/0205028)
16. Priyadharshan T, Sumathipala S (2018) Text summarization for Tamil online sports news using NLP. In: 2018 3rd international conference on information technology research (ICITR). IEEE, pp 1–5
17. Quinlan J (1986) Induction of decision trees. *Mach Learn*
18. Frank R (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386
19. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 20:53–65
20. Schölkopf B, Luo Z, Vovk V (2013) Empirical inference: Festschrift in honor of Vladimir N. Vapnik. Springer Science & Business Media
21. Sindhu L, Idicula SM (2017) Plagiarism detection in Malayalam language text using a composition of similarity measures. In: Proceedings of the 9th international conference on machine learning and computing, pp 456–460
22. Singh R, Singh S (2021) Text similarity measures in news articles by vector space model using NLP. *J Inst Eng (India) Ser B* 102:329–338
23. Song G, Wang Y (2020) A hybrid model for medical paper summarization based on covid-19 open research dataset. In: 2020 4th international conference on computer science and artificial intelligence, pp 52–56
24. UPSC. Syllabus data source. https://www.upsc.gov.in/sites/default/files/Engl_CSP_2017.pdf

A Novel Explainable Artificial Intelligence-Based Deep Reinforcement Learning for Secured Smart City Applications



Vandana Sharma, Tamizharasi Seetharaman, K Mohammed Essam, and Ahmed Alkhayat

Abstract In recent times, smart cities have acquired tremendous transition toward sustainable development. However, with growing advancements, there comes numerous security challenges. With the rapid technological advancements, there comes greater connectivity between devices giving rise to a plethora of data and security constraints. Since data is continuously generated and transmitted across the smart city applications, preserving security measures has become a vital factor. Conventionally, intrusion detection systems are widely used to monitor and preserve the security parameters across smart cities. One major challenge with the traditional approaches is that most of the security methods described are supervised classifier and they require high-quality labels. But, however, real-time applications mainly deals with unlabeled data. Further, it lacks in explainability of the predictive model. In order to overcome these constraints, in this paper, we define an novel explainable deep reinforcement learning techniques for securing smart city applications. The proposed approach actively prevents various security threats across smart cities and the experimental results provide improved explainability, stability, security, and efficiency measures.

Keywords Deep reinforcement learning · Explainable artificial intelligence · Smart city · Security

V. Sharma

Amity Institute of Information Technology, Amity University, Noida Campus, New Delhi, India

T. Seetharaman (✉)

Department of CSE, School of Engineering and Technology, CMR University, Bangalore, India

e-mail: gst30091993@gmail.com

K. Mohammed Essam

Department of AIML, Acharya Institute of Technology, Bangalore, India

A. Alkhayat

College of Technical Engineering, The Islamic University, Najaf, Iraq

1 Introduction

The advent of smart cities offers more sophisticated and sustainable life to all the urban residents. The prime objective here is to improve the people quality of life with greater technological innovation and automation. In general, smart cities make use of sensors, communication networks, and IoT devices to collect and transfer data across various applications. Data forms the prime source of smart city applications through which they operate in an efficient manner [1]. Though smart cities seem to be a greater concept, they come with their individual set of challenges. This is because of the reason that smart city applications are connected to larger networks, which sometimes impose greater vulnerabilities over the user data. At the same time, there is no denial that smart cities form the inevitable part of the future world [2].

Smart cities need smart security features for the number of reasons. First and foremost is that the people need to feel assured of the fact that their data remains safe and private. Next is the use of smart devices which is growing exponentially, which ultimately enables the data exchange at a large scale. Further, the rapid connectivity between the smart devices significantly increases the possibility of data breaches and security issues. In addition, the intelligent objects across the connected smart city networks continuously transfer the data to cloud environment for efficient processing and decision-making, which is highly a risky process as the cloud remains to be untrusted source to store the sensitive user data. Any compromise in smart city data will lead to adverse effects with irreversible loss of information. This creates the need for significant security features for smart city applications [3, 4].

Artificial intelligence (AI) is the most powerful tool to combat emerging security threats across the smart city applications. AI can be adopted to effectively monitor various aspects of smart city applications ranging from real-time traffic monitoring to data analytics and decision-making. More specifically, predictive modeling has the most wider scope in securing smart city applications. These models can be applied to monitor various tasks and to alert users in case of data breaches and security vulnerabilities. They are most widely used to monitor and discover circumstances that are suspicious and vulnerable. In short terms, this technique is near equivalent to human intelligence and provides several advantages when it comes to securing smart city applications [5].

However, with growing amount of big data, securing smart cities with traditional approaches is highly complicated [6–8]. Security researchers are highly concerned about the lack of explainability measures of the deep learning techniques in predicting security threats. This in turn limits the applications of artificial intelligence techniques for many real-world critical applications. Also, the potential of predictive models varies significantly with increasing complexity of the datasets. Hence, the efficient approaches are required to deal with massive and complicated data [9, 10].

Explainable artificial intelligence (XAI) is an emerging paradigm that provides an efficient solution to the aforementioned security issues. These techniques help to interpret the decision made by deep learning models, thereby offering more clarity on the final decision. The effective implementation of this model overcomes the key

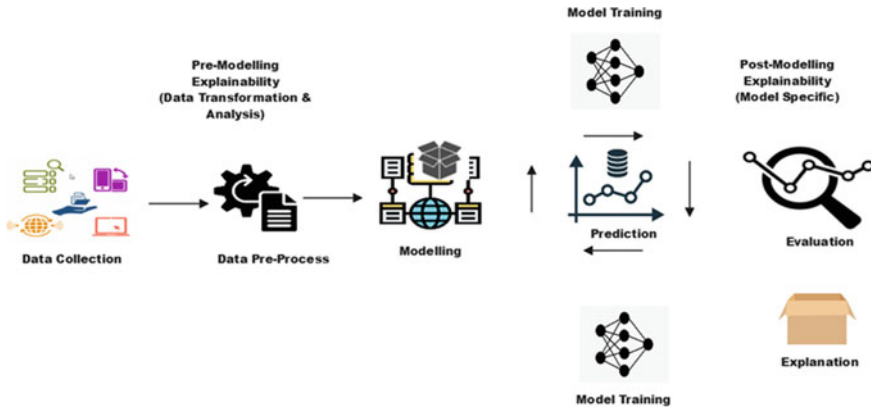


Fig. 1 Overview of explainable artificial intelligence technique

barriers associated with the smarty city security and reduces the model errors with improved trouble shooting features. This technique explicitly understands the important features associated with the predicted outcome, thereby providing more clarity on the decision to the end users. Unfortunately, the existing XAI techniques cannot be directly applied to secure the smart city applications due to various reasons. First is due to adversary resistance, and next is robustness, explanation methods, and many more. Hence, exploring more in this regard will add significant value to the research community and offer improved solution to the smart city applications. In this background, this paper presents an novel and innovative explainable deep reinforcement learning technique for securing smart city applications. A more advanced learning paradigm called deep reinforcement learning is adopted to deal with complex security issues across the smart city applications. The proposed model offers increased actionability by providing model guidance, patching, and inspection. The rest of the paper is organized as follows. Section 3 describes the proposed model, Sect. 4 briefly illustrates the results and discussions. Section 5 concludes the proposed work with future research directions (Fig. 1).

2 Related Works

In this work, the authors emphasize the significance of the security measures across the smart city applications [11]. The techniques such as cryptography, blockchain, game theory, machine learning, and biometrics are mainly used to preserve the security measures in smarty city systems. This work provides a brief description on smart city architectures, issues, and emerging trends.

In [12], the authors address the various security issues in IoT and their effective countermeasures. This work deeply analyzes the integration of cloud computing and

IoT techniques for smart city applications and their emerging security threats. This work also provides some effective countermeasures to protect cybersecurity features.

In [13], the authors assess the impact of emerging technologies on smart city implementations. The technological advancements are assessed with respect to various applications such as smart transportation, smart homes, smart health. The author explores specifically on cutting-edge technologies to enhance the role of smart cities from a future perspective.

In [14], the authors enforce the need for smart security to smart city applications. The analysis is made from various perspective to assess the security features across the smart city applications. It is found that the security issues at the application level are comparatively greater than the other layers.

In [15], the author integrates the artificial intelligence and blockchain technology to secure the smart city application. The use of blockchain ensures the data integrity and security features, whereas the artificial intelligence technique enables efficient management of intrusion detection systems to predict the security threats with increased security measures.

Secure and lightweight protocol for smart city surveillance [16]. This approach is implemented in VANET environment and the results are analyzed. The algorithm makes use of real-time data for its implementation and finds the flaws across VANET in an efficient manner. The bitwise X-OR operations and DLA-based key exchange protocols are implemented to improve the security measures.

An efficient intrusion detection system using artificial intelligence techniques for smart cities is given in [17]. The prime objective of this work is to prevent Distributed Denial of Service Attack (DDoS) across the smart cities. This approach works on the basis of the restricted Boltzmann machine learning algorithm to overcome the security threats. The performance of the proposed approach is evaluated and it is found to be comparatively better than existing techniques.

An machine learning-based intrusion detection system for securing smart city applications is given in [18]. The objective here is to preserve the security, privacy, and integrity measures. This approach works on the basis of the hybrid optimization algorithm. The objective here is to improve the high-level detection accuracy of the security threats. The analysis is made through the use of hybrid genetic algorithm.

A detailed review on the use of explainable artificial intelligence techniques for security application is given in [19]. The authors emphasize the data centric and model centric explainable artificial intelligence techniques. Further, a detailed study on explainable artificial intelligence for health care, smart city, smart transport, and many more is provided in detail.

An explainable artificial intelligence called DARPA is given in [20]. This approach provides the more efficient models in a more explainable and interpretable manner. From a security perspective, this approach provides more significant results and improved performance measures.

It is observed from the literature [21–25] that security remains to be the majority concern across smart city applications. Hence, more advanced research is required in this background.

Citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], as well as a URL [5].

3 Proposed Model

Inspired by the greater success of artificial intelligence across various applications, in this paper, explainable artificial intelligence technique is used for securing smart city application. The working of the proposed approach is clearly illustrated in Fig. 2. The key idea here is to effectively train the deep neural network algorithm based on reinforcement learning with explainability. This approach is based on various actions and states.

3.1 Background and Problem Setup

In general, reinforcement learning algorithms are used to address the sequential decision-making problems across complex environments the neural network-based agent. The techniques of deep reinforcement learning are most prevalent in security-related applications as they can efficiently handle the complex sequential decision-making tasks. In this research work, we intend to automate the complex security tasks

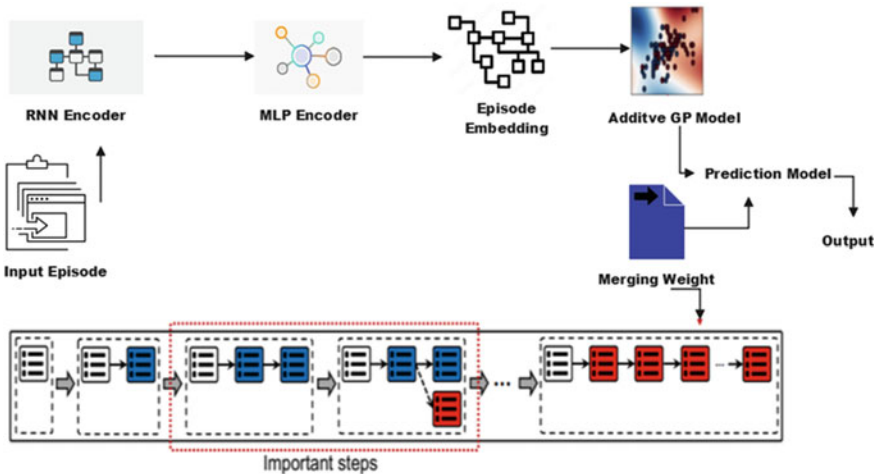


Fig. 2 Overview of the proposed explainable deep reinforcement learning

with the proposed explainable deep reinforcement learning techniques. In order to attain this objective, first it is necessary to model the reinforcement learning problem.

First, any kind of sequential decision-making problem could be formalized into reinforcement learning (RL) problem, when an agent analyzes the target environment and performs appropriate actions to attain the final objective. The agent attains a reward, whenever the task is completed in an optimal manner. Here, the observation of environment is given as an input and the output is the appropriate action taken by the agent.

Here, in this paper, we define a deep reinforcement learning (DRL) problem using the Markov chain decision process. Consider a tuple $\langle \mathbf{P}, \mathbf{Q}, \mathbf{R}, \mathbf{S} \rangle$ where \mathbf{P} and \mathbf{Q} represent the final state and action sets. In particular, the state of the agent at a certain point of time is given as P_t and action as Q_t . The transition function is then given as $\mathbf{T}: \mathbf{P} \times \mathbf{Q} \rightarrow \mathbf{P}$, where $T_{pp'}^q = \mathbb{R} [P_{t+1} = Q' | P_t = P, q_t = q]$, and this represents the probability that the agent transits from the state \mathbf{P} to \mathbf{P}' by performing action q at the time interval t . Here, the reward function is defined as $\mathbf{S}: \mathbf{P} \times \mathbf{Q} \rightarrow \mathbf{S}$, S_p denotes the reward, if the agent performs an action q at the state p . The ultimate objective here is to train the policy network $\pi(q|p)$ for the agent that can increase the total reward of the agent. In a more formal manner, the total reward for the agent is given using state value function provided as, $U_\pi(P) = \sum_{q \in Q} (q|p) \left(T_p^q + \gamma \sum_{p' \in P} T_{pp'}^q U_\pi(P') \right)$; similarly, the action value function is computed as, $M_\pi(P, q) = S_p^q + \gamma \sum_{p' \in P} T_{pp'}^q \sum_{q' \in Q} T_{p'r'}^q \sum_{q'' \in Q} \pi(q''|P') N_\pi(P', q')$. Such that, $\gamma \in [0, 1]$ denotes the discount factor that helps to reduce the uncertainty measure of agent's future functions.

3.2 Problem Setup

Now let us consider the deep reinforcement learning (DRL) with a agent that is trained with policy gradient. The major focus of the proposed work is to identify the significant that leads to the final result. In order to attain this objective, the access should be given only to the agent actions, reward, and environment states. Here, we assume the value of either the policy network or Q function. The given N iterations, $T = \{A^{(i)}, b_i\}_{i=1:N}$ of the final target agent. Then, the i th episode with length T is given as $A^{(i)} = \{P^{(i)}, q_t\}_{t=1:T}$. Here, $P_t^{(i)} \in R^{jk}$ and, $Q_t^{(i)} \in R^{jq}$ represent the state and the corresponding action steps at t . The final reward of the episode is computed by b_i . The objective here is to emphasize the most important L steps associated with episode $A^{(i)}$.

3.3 Explanation Model Design of the Proposed Reinforcement Learning Algorithm

In this paper, we propose a novel and efficient self-explainable reinforcement learning model for securing smart city applications. To improve the feature significance, i.e., to better find the associations learned by deep learning-based predictive model, we incorporate explainability to the final layer rather than the input layer associated with the predictive model. In formal terms, the proposed model is written as $h(f(a))$, where $f(\cdot)$ denotes the feature extractor and $h(\cdot)$ represent the explainable prediction model. The second major contribution is that we design an deep Gaussian process for feature extractor to acquire the correlation between each step and those across various iteration. This is often exhibited as set of episodes collected across the same DRL agents, and apart from finding various levels of correlations, the major significance of the proposed model over conventional deep neural networks is that it models the joint distribution of the output signals, dealing effectively with the output signal uncertainty. An interpretable Bayesian predictor model is designed finally to infer the importance of deliver step and distribution of final rewards.

Given a target agent $A^{(i)}$, the EDGE first inputs are provided to the recurrent neural network encoder, that significantly embeds each step into the episode $\{g_t^{(i)}\}_{t=1:T}$. The value of episode embedding $e^{(i)}$ is obtained by passing the EDGE to shallow NLP. Then, proposed framework is used to process $\{g_t^{(i)}\}_{t=1:T}$ and $e^{(i)}$ and then to find the value of latent representation measure associated with entire episode $f_{1:T}^{(i)}$. This function representation can easily acquire the correlation across the steps and episodes. Finally, the EDGE inputs $f_{1:T}^{(i)}$ to attain the predicted final reward associated with input episode. Our proposed prediction model is based on linear regression model. The regression coefficient value is used to acquire the input episode.

In general, the Gaussian process represents the infinite collection of random variables. Thus, any finite subset of variables shows the multivariate Gaussian distribution. In Statistics, the Gaussian processes other defined as non-parametric functions given by $f: \mathbf{A} \rightarrow \mathbf{R}$. In more formal terms, if f has GP, previously, $f \sim \mathbf{GP}(\mathbf{0}, L_r)$, such that $L_r(\cdot, \cdot)$ denotes the positive semi-definite kernel function using the parameter γ . Thus, any finite (collection of $f \in R^N$ adheres multivariate Gaussian distribution represents as $(f|\mathbf{A}) \sim \mathbf{N}(0, L_{AA})$. At this point, $L_{AA} \in R^{N \times N}$ as the covariance matrix, with $(L_{AA})\mathbf{id} = L_\gamma(A_i, B_d)$. Here, in the proposed approach, we make we of square exponential kernel function $L_\gamma(A_i, A_d) = \exp(-1/2(A_i - B_d)$ and $\gamma - \theta_L$. However, the major challenges here is that the convention Gaussian process with SE kernel assumes that input space is Euclidian. This is invalid for a real-world data centric application like smart cities with high-dimensional application inputs. To overcome this challenge, the proposed model applies dimensionalities reduction used in DNN and makes use of GP to latent space. In this way, the proposed model works well ever for computer datasets.

In the proposed model, we acquire the sequential dependency across the episode through RNN and deep net with a kernel function. For a particular episode, the first

step is to concatenate the state and action ($g_t^{(i)} = [p_t^{(i)}, g_t^{(i)}]$), this input is further given to RNN g_θ , and the value of latent representation associated with the episode is computed as $\left\{g_t^{(i)}\right\}_{t=1:T}$, where $g_t^{(i)} \in R^M$, which denotes the state action at the time t . Then, we compute the episode embedding by providing the last step's hidden representation as MLP $e_{\theta_1}: g_t^{(i)} \rightarrow e^{(i)} \in R^M$. Once the value of $\left\{g_t^{(i)}\right\}_{t=1:T}$ is obtained and $e^{(i)}$, the additive GP framework can be used to capture the correlation between steps and across various iterations.

In formal terms, additive GP is computed as $f = \sum_c \alpha_c f_c$, which is the weighted sum of c independent GPS. In this case, $f_c \sim \text{GP}(0, L_c)$ denotes the c th GP component, and this is the point where the covariance function L_c can be formally applied to input features. Following by construct, we construct our own GP framework as the sum of components f_e and f_t . In specific $f_t \sim \text{GP}(0, L_{\gamma t})$ model the correlation between time steps. Formally, the measure of covariance between L th step in iterate j is computed as $L_{\gamma t}(g_t^{(i)}, g_t^{(d)})$. Moving toward the modeling of correlation across individual steps, $f_e \sim \text{GP}(0, L_{\gamma e})$ acquires the higher-level cluster structures within the collected episodes. Formally, $L_{\gamma t}(e^{(i)}, e^{(c)})$ denotes the episode-level covariance across any pair of steps i and c .

The final model is defined as $f = \alpha_t f_t + \alpha_e f_e$. Here, α_t and α_e denote the weight components. Then, $T \in R^{\text{NX}T} X(j_p + j_q)$, $f \in R^{NT}$ is represented as $f|A \sim N(0, 1 = \alpha_t^2 L_{\gamma t} + \alpha_e^2 L_{\gamma e})$ denotes the collected episode representation. Here, the flattened matrix of T is attained as $X \in R^{NT} X(d_p + d_q)$.

3.4 Prediction Model

To assure explain ability, we make use of the linear regression as the basis for our prediction model. Here, the linear regression model denotes the significance of every input entity. The first step here is to convert the flattened response f to the matrix of forms $F \in R^{\text{NX}T}$. Here, the i th encoding given by the proposed model is computed as $F^{(i)} \in R^T$. Then, the next step is to find the conditional likelihood of the discrete and continuous final reward. In such a case, when a_i is continuous, we can make use of the conventional GP regression model. The mixing weight is computed as $A_i = F^{(i)} D^T + E_1$, where $D \in R^{1 \times T}$ and the observation noise is computed as $E_1 \sim N(0, \sigma^2)$. Further, $b_i | F^{(i)} \sim N(F^{(i)} D^T, \sigma^2)$ gives the conditional likelihood distribution. In the case of the discrete final reward with finite number of possible values, the proposed approach makes use of Softmax prediction model to perform classification. Such that, the categorical distribution is given as $P(b_i = l | F^{(i)}) = \frac{\exp(F^{(i)} D^T)_l}{\sum_L \exp(F^{(i)} D^T)_L}$, where k represents the total number of classes and $D \in R^{1 \times T}$ is the mixing weight.

Finally, the mixing weight constant is computed as $f|A \sim N(0, l = \alpha_t^2 l_{\gamma t} + \alpha_e^2 l_{\gamma e}) b_i | F^{(i)} \sim \{\text{cal}(\text{Softmax}(F^{(i)} D^T)) \text{ if conducting class or } \{N(F^{(i)} D^T, \sigma^2) \text{ otherwise}\}$.

From this, it is observed that the step importance derived from the mixing weight is the global explanation from the observation, and we can state that the high correlation tends to possess a joint effect over the game result. Thus, we can converge global explanation with step correlations in $L_t(A, A)$ to acquire the fine-grained clarity of every game. To be more specific, for any given episode and important steps represented by mixing weight, we can easily identify the steps that have higher correlation, in addition to globally important steps. All together form the local explanation associated with the episode. The episode correlation associated with $L_e(A, A)$ unleashes the cluster structure within a series of episodes. This helps to differentiate and categorize the explanations of similar episodes.

4 Results and Discussions

The performance of the proposed deep reinforcement learning model is evaluated based on various standard matrices and it is effectively compared with several conventional approaches. For every task associated with the proposed model, a well-trained policy is used as a target agent and compared with the proposed approach using various baseline parameters. The simulation setup is created using MATLAB and Simulink with good quality system configurations. An initial simulation model is created for smart traffic management system. In general, the smart traffic management system is concerned with efficient approaches to minimize the congestion measures and enhance the safety of streets through IoT technology. Under such circumstances, the data is collected through IoT sensors in the form of images and videos. The data has to be processed by the cloud server in a more efficient manner and should be returned to the users. In such cases, there exists a probability of emerging security threats as the data needs to continuously collected and sent to the cloud servers for response and decision-making. The proposed explainable deep reinforcement learning algorithm is implemented under this setup to assess its performance from security perspective. The proposed model is evaluated under various dimensions which include stability, explainability, performance, and efficiency measures.

In Fig. 3, the value of mean and fidelity score associated with every model is computed and it is compared to find the better results. Here, the components of 'X' axis represent the various choices of L and it corresponds to different number of episodes. In this experiment, the proposed approach makes use of Rational Net through which the global explanations are derived using the mixing weight associated with the evaluation.

Next, the model fitting performance is evaluated based on various parameters such as explainability, stability, and efficiency, and the results are illustrated in figure. The explainability measures associated with each selected model is assessed using random perturbations. It also helps to find the unique correlation measures that exist between every variables. It is observed from the result that proposed model could better fit with problem scenario of IoT security and provides greater explainability measures. Similarly, based on the perturbation strength, the stability

Fig. 3 Comparison of fidelity measures with various approaches

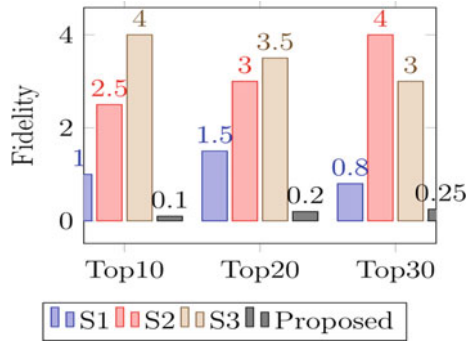
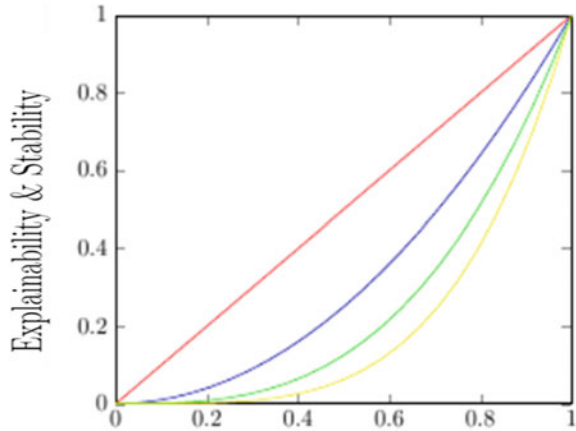


Fig. 4 Comparison of performance measures with various approaches



measures are evaluated and the proposed approach provides comparatively better results. Next, the efficiency is computed based on the amount of time required for training and explanation process associated with every method. Based on the run time of deriving explainability, the efficiency measures are computed. The proposed approach provides significantly better results in terms of efficiency as well (Fig. 4).

5 Conclusion

In this paper, we introduce a novel and innovative explainable artificial intelligence-based deep reinforcement learning for securing smart city applications. The model is constructed based on deep reinforcement learning algorithm that makes use of the black-box approximation to interpret explainability of the model. The experimental results provide improved fidelity measures, explainability, stability, and efficiency measures. Further, the proposed model provides numerous advantages to the end users from various perspectives. In future, this work could be extended to be

applied for data-intensive critical application with higher complexity. The explainability factor associated with the proposed approach is improved efficiently to provide better clarity on the underlying security parameters in the system.

References

1. Zhang K et al (2017) Security and privacy in smart city applications: challenges and solutions. *IEEE Commun Mag* 55(1):122–129
2. Lacinák M, Ristvej J (2017) Smart city, safety and security. *Procedia Eng* 192:522–527
3. Ma C (2021) Smart city and cyber-security; technologies used, leading challenges and future recommendations. *Energy Rep* 7:7999–8012
4. Lv Z et al (2021) AI-empowered IoT security for smart cities. *ACM Trans Internet Technol* 21(4):1–21
5. Wang P, Ali A, Kelly W (2015) Data security and threat modeling for smart city infrastructure. In: 2015 International conference on cyber security of smart cities, industrial control system and communications (SSIC). IEEE
6. Kitchin R, Dodge M (2019) The (in) security of smart cities: vulnerabilities, risks, mitigation, and prevention. *J Urban Technol* 26(2):47–65
7. Habibzadeh H et al (2018) Sensing, communication and security planes: a new challenge for a smart city system design. *Comput Netw* 144:163–200
8. Haque AKM, Bahalul BB, Dhiman G (2022) Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. *Exp Syst* 39(5):e12753
9. Chen D, Wawrzynski P, Lv Z (2021) Cyber security in smart cities: a review of deep learning-based applications and case studies. *Sustain Cities Soc* 66:102655
10. Braun T et al (2018) Security and privacy challenges in smart cities. *Sustain Cities Soc* 39:499–507
11. Cui L et al (2018) Security and privacy in smart cities: challenges and opportunities. *IEEE Access* 6:46134–46145
12. Singh D et al (2020) Security issues in IoT and their countermeasures in smart city applications. In: *Advanced computing and intelligent engineering: proceedings of ICACIE 2018*, vol 2. Springer Singapore
13. Rao SK, Prasad R (2018) Impact of 5G technologies on smart city implementation. *Wireless Pers Commun* 100:161–176
14. Moch N, Wereda W (2020) Smart security in the smart city. *Sustainability* 12(23):9900
15. Singh J et al (2022) Artificial intelligence and blockchain technologies for smart city. *Intell Green Technol Sustain Smart Cities* 317–330
16. Akram MW et al (2021) A secure and lightweight drones-access protocol for smart city surveillance. *IEEE Trans Intell Transp Syst* 23(10):19634–19643
17. Elsaedy A et al (2019) Intrusion detection in smart cities using restricted Boltzmann machines. *J Netw Comput Appl* 135:76–83
18. Gupta SK, Tripathi M, Grover J (2022) Hybrid optimization and deep learning based intrusion detection system. *Comput Electric Eng* 100:107876
19. Zhang Z et al (2022) Explainable artificial intelligence applications in cyber security: state-of-the-art in research. *IEEE Access*
20. Gunning D, Aha D (2019) DARPA's explainable artificial intelligence (XAI) program. *AI Mag* 40(2):44–58
21. Aloqaily M et al (2019) An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Netw* 90:101842
22. Li D et al (2019) IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. *Int J Inf Manag* 49:533–545

23. Elsaedy AA et al (2020) Replay attack detection in smart cities using deep learning. IEEE Access 8:137825–137837
24. Elrawy MF, Awad AI, Hamed HFA (2018) Intrusion detection systems for IoT-based smart environments: a survey. J Cloud Comput 7(1):1–20
25. Saba T (2020) Intrusion detection in smart city hospitals using ensemble classifiers. In: 2020 13th International conference on developments in eSystems engineering (DeSE). IEEE

A Review of IoT Security Solutions Using Machine Learning and Deep Learning



Anamika Chauhan and Kapil Sharma

Abstract The Internet of Things (IoT) is a rapidly developing field, projected to connect 22 billion smart devices in a global market worth 1567 billion USD by 2025. The integrated and multidisciplinary nature of these resource-constrained devices responsible for construction of IoT systems renders them susceptible to security attacks. Conventional methods of ensuring security are relatively inefficient as the types, surfaces and severity of attacks continue to evolve. Promising alternatives offered by machine learning (ML) and deep learning (DL) can be employed to embed intelligence in the system, by facilitating the detection of compromised security. In this survey paper, a discussion of IoT infrastructure, security concerns, types and surfaces of attacks prefaces a systematic, layer-wise review of the ML/DL models and frameworks to ensure system security. We also present the current challenges and prospective directions of research concerning the utilization of ML/DL techniques in offering system security in an IoT environment.

Keywords Internet of Things · IoT security · Attacks · Privacy · Machine learning · Deep learning · Smart devices

1 Introduction

IoT is primarily concerned with a distributed and interconnected network of actual devices possessing limited storage and computation which communicate via wired or wireless technologies. These physical entities are capable of data collection, processing and exchange because they are supplemented with software, electronics (sensors, actuators, etc.) and Internet accessibility [102]. The use of IoT technology

A. Chauhan (✉) · K. Sharma
Department of Information Technology, Delhi Technological University, Delhi, India
e-mail: anamika@dtu.ac.in

K. Sharma
e-mail: kapil@ieee.org

is spread across but not limited to agriculture, healthcare, retail, military and energy industries [25].

An increasing number of smart devices carry highly confidential user/client information like location, health data and contact details making the security of IoT devices a pressing concern. However, owing to the complexity and integrative, large-scale nature of an IoT system, maintenance of security is an uphill task in such systems. IoT devices mostly work in an unguarded environment and are impeded in terms of resources like battery, computation power, memory, bandwidth, etc. This makes them very configurable and renders complex algorithm-based security techniques infeasible. The landscape of threats from adversaries is also increased by usage of enabling technologies like cloud computing, fog computing and software-defined networking in Internet of Things [47]. Moreover, the introduction of new attack surfaces because of the interconnected and interdependent nature of IoT makes it less secure to attacks than a traditional computing system [80].

Conventional security schemes, like encryption, access control and authentication are inadequate for large interconnected systems due to inherent vulnerabilities of each component. ML and DL classifiers and frameworks offer a viable alternative for this problem. Machine learning (ML) refers to algorithms that learn from past experience to optimize performance criteria [106]. Deep learning (DL) is its sub-domain that employs nonlinear layers for feature abstraction and transformation. These classifiers and frameworks can be used to train models to recognize attacks of different types and offer appropriate protection from the same. In dynamic networks, such models outperform conventional security techniques due to ability to learn to detect and thereby prevent new attacks from causing system damage.

This research presents a recent review of IoT security solutions with focus on ML and DL frameworks, following objectives were outlined to achieve:

1. To begin, a brief description of various layers in the IoT infrastructure has been provided.
2. A graphical summary of the security concerns and types of threats along with the potential attacks that can be encountered at each component of IoT infrastructure follows.
3. A systematic review of the ML and DL classifiers and frameworks that helps to offer security to IoT environment at each layer.
4. Conclude with the discussion of research gap and future challenges to facilitate a prospective direction of research.

2 IOT Infrastructure and Attacks

The generic architecture of IoT comprises of three main layers: perception layer, network layer and web (or application) layer [43]. The infrastructure has been diagrammatically displayed in Fig. 1. The attacks are not limited to one layer but span across several layers making detection and mitigation a complex task.

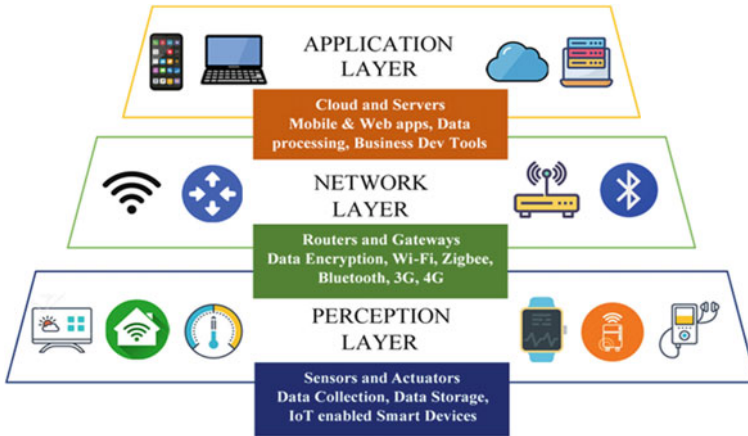


Fig. 1 IoT architecture

2.1 Perception Layer

The perception or the physical layer primarily takes care of hardware components like sensors and devices which are used to send and receive data through various communication protocols, such as Bluetooth, Zigbee, RFID, etc. [24, 117]. Different protocols like LAN (IEEE 802.11ah), PAN (IEEE 802.15.4e, Z-Wave) and cellular networks (LTE-M, EC-GSM) are used for the connection established.

2.2 Network Layer

The network layer is a transmission medium that establishes a connection between devices and smart services. The protocols used for this purpose include GSM, Wi-Fi, 3-5G, IPv6, etc. [122]. This layer houses local clouds and servers for storage and processing of information that acts as a middleware joining the perception layer and this layer [107]. The functions of a middleware include cooperation between different IoT devices [107, 117], device discovery [117] and awareness of a device with respect to its surrounding IoT devices.

2.3 Web/Application Layer

In IoT frameworks, the web layer offers support to IoT users with the help of web and mobile-based software applications. Few of the prevalent services include smart healthcare [63], smart homes/buildings [12], smart transportation [8], smart agriculture [128], smart grid [88], etc.

2.4 Threats in IoT

Attacks occurring in IoT environments can be broadly identified as cyberattacks and physical attacks. All of these attacks may affect a single layer or across layer depending on the mechanism of the attack.

2.4.1 Cyberattacks

These attacks comprise the threats that hack the overall system and target various IoT nodes in a wireless network so as to manipulate (modify, erase, steal, destroy) the information of a user. Cyberattacks can be further categorized into active and passive attacks.

2.4.2 Physical Attacks

These attacks cause physical damage to IoT devices and consequently, physical devices like routers, sensors, mobile, camera, etc., are susceptible to such attacks. Herein, the attackers interrupt the service but the network is not required to intrude into the system [26, 113, 142] (Fig. 2).

Attack Name	Attack Surface	Attack Type
Authentication Attack		Active
Node Capture		Physical
Radio Interference		Active
User Tracking		Passive
Blackhole Attack		Active
Hello Flood Attack		Active
Man-in-the-middle attack		Active
Port Attack		Passive
Rank Attack		Active
Replay Attack		Active
Selective Forwarding (Grey Hole) Attack		Active
Sinkhole Attack		Active
Sybil Attack		Active
Traffic Analysis		Passive
Wormhole Attack		Active
Bluejacking		Active
Bluesnarfing		Passive
Malware, Spyware And Spamware Attack		Active
Spear-Phishing Attack		Active

Attack Name	Attack Surfaces	Attack Type
Spoofing		Active
Jamming		Active
Sniffing Attack		Active/Passive
Injection Attack		Active
Denial of Service		Active
Eavesdropping		Passive

KEY	
	Perception Layer
	Network Layer
	Application Layer

Fig. 2 Threats in IoT

3 Review of Solutions Using Machine Learning and Deep Learning in Security of IOT

The paper has divided the review according to the ML and DL solutions applied in IoT according to the infrastructural layers of IoT. Table 1 outlines all the papers reviewed for their ML/DL solutions presented and the relevant IoT threats addressed by them.

3.1 Perception Layer

Wang et al. suggested an authentication system to be implemented by training an extreme learning machine model to detect spoofing. The model utilizes multidimensional characters of radio channels and exhibits improved accuracy. The authors in [119] successfully demonstrate the application of deep learning in building authentication schemes. The proposed model involves employment of Wi-Fi signals produced from Wi-Fi-enabled IoT appliances (such as smart television, smart air-conditioners, etc.) to understand specific physiological and behavioural features of individual human beings related to their day-to-day activities (e.g. walking).

A deep neural network is built by extraction of these features and generation of Wi-Fi fingerprints unique to each user. Several studies by Xiao et al. developed schemes for authentication of the PHY layer that utilizes radio channel data and implements RL to achieve optimum test threshold in spoofing detection. Their results show that Q-learning-based methods can lead to a 64.3% decrease in authentication error.

In a recent study, Senigagliesi et al. [116] have tried to achieve a trade-off between IoT safety and complexity since many physical layer authentication schemes using ML or DL suffer because of huge computational burden of the algorithms, while having to process enormous amounts of data generated at the PHY layer. The proposed framework makes use of principal component analysis (PCA) with t-distributed stochastic neighbour embedding (t-SNE).

Fang et al. [45] demonstrated the training process of physical layer authentication in the form of a convex problem which is tackled by implementing an adaptive algorithm is built upon kernel least mean square. In [37], the authors have suggested enhancement of the security architecture of RF systems in IoT networks by performing real-time verification of wireless nodes using RF-PUF, which is a framework fabricated using neural networks. Similarly, Liao et al. [81] proposed the usage of data augmentation along with DNN to enhance the accuracy in their multi-user authentication model. Namvar et al. [99] suggested a centralized system to address jamming attacks in IoT networks comprising of appliances with limited resources.

Aref et al. in [23, 84] have based their study on a system of multiple WACRs where each radio tries to escape the signals from other radios and a jamming signal that spans over the full spectrum. The anti-jamming model uses the spectrum knowledge

Table 1 AL/ML techniques applied for IoT attacks

Reference	ML/DL technique used	Threat
<i>Perceptron layer</i>		
[90]	RL	Spoofing
[91]	Optimization algorithms	Spoofing
[92]	CMC, t-SNE	Spoofing/eavesdropping
[93]	Optimization algorithms	Authentication attack
[94]	ANN	Authentication attack
[95]	ANN	Authentication attack
[96]	RL	Jamming
[97]	RL	Jamming
[98]	RL	Jamming
[99]	CNN, RL	Jamming
[100]	SVM	Poisoning attack
[101]	SVM, KNN	Injection attack
[102]	Optimization algorithms	Eavesdropping
[103]	Deep RL	Eavesdropping
[104]	Optimization algorithms	Eavesdropping
<i>Network layer</i>		
[105]	RNN	Sybil attack
[106]	SVM	Sybil attack
[107]	PCA, CMC, SSL/HDL	Sybil attack
[108]	KNN	Rank attack
[109]	DT, KMC, SSL/HDL	Wormhole attack
[111]	Deep RL	Routing attack
[112]	ANN	HF, VN and rank attack
[113]	DBN	HF
[114]	DT, RF, CNN	Botnet flooding attack
[115]	ANN	Man-in-the-middle attack
[116]	Regression	Blackhole, DDOS
[117]	SVM, KNN, DT, RF, KD TREE, ANN	DDOS
[118]	ANN	DDOS
[120]	SVM, LR	Intrusion attack
[121]	CNN	SF, VN and hole attack
[122]	ANN	DDOS, OS and hole attack
[123]	KMC, SAE, SSL/HDL	Impersonation attack
[124]	ANN, SAE, SSL/HDL	Impersonation attack
[125]	OPF	SF, rank and hole attacks
[126]	SAE	Impersonation and injection

(continued)

Table 1 (continued)

Reference	ML/DL technique used	Threat
<i>Web/application layer</i>		
[127]	SVM, DBN	Malware
[128]	CNN	Malware
[129]	SSL/HDL	Malware
[130]	KNN, RF	Malware
[131]	CNN	Malware
[132]	RNN	Web spam
[133]	SSL/HDL	Abnormal behaviour
[134]	ANN	Intrusion attack

from the WACRs and RL-based algorithm to shun both the jamming signal and the interference signals of other radios. Han et al. [56] used deep CNN along with RL to enhance the efficacy of the Q-learning-based algorithm. Baracaldo et al. [30] recommended supervised ML model which uses data provenance for identifying and filtering poisoning attacks.

Ozay et al. [104] focuses on detecting false data injection attacks that hit the physical layer in the smart grid. The model proposed by them uses SVMs and KNNs to demonstrate that learning algorithms perform better in detecting observable and unobservable attacks than algorithms that employ state vector estimation (SVE) methods.

A machine learning-based antenna design was proposed by Hong et al. [60] for IoT communications that use ambient backscatter technology. The authors in [15] designed a model for the secure transmission of medical information from patients to health service providers with the implementation of DRL. Both these studies help in minimizing the issue of eavesdropping, which if unaddressed, is a major threat to the privacy of the system and users.

3.2 Network Layer

Being the most extensive surface of the IoT system, ensuring the security of network layer against intruders is imperative for maintenance of the overall functioning of the IoT system.

Zhang et al. proposed a method capable of automatically predicting the occurrence of a sybil attack using artificial RNN architecture of long short-term memory (LSTM). Saraswathi et al. developed a support vector regression model for rigid sybil attack detection in IoT-enabled WSNs. Deng et al. [40] put forth a system capable

of detecting intrusion caused by sybil attack. By amalgamating fuzzy c-means clustering with feature selection-oriented principle component analysis, the proposed method was found to be successful in raising detection effectiveness.

Neerugatti and Reddy [100] proposed a rank attack detection technique based on KNN algorithm. Three new machine learning-based models capable of identifying wormhole attacks in IPv6 over low-power wireless personal area networks for IoT were proposed by Shukla [120]: an unsupervised k-means clustering IDS, an intrusion detection system based on decision tree and a dual-level, hybrid combining the aforementioned two systems. Fatima-Tuz-Zahra et al. [49] proposed an ML-based framework potent of recognizing attacks-wormhole and rank. Guo et al. [52] proposed a DRL rooted protocol for routing that possessed quality of service awareness. It helps prevent routing attacks by means of network interaction, intelligence extraction from archival network demand records and dynamic optimization of decision-making policy employed. Yavuz et al. proposed a deep neural network model capable of three-ringed attack identification, viz., decreased rank, hello flood and version number in a 1000-node simulation of an IoT network.

Sai Srinivas and Manivannan [5] developed a powerful, original model for recognition and mitigation of HELLO flooding attacks using a combination of deep belief network with improved rider optimization algorithm. The work suffered from the limitation that on changing the routing protocol, the underlying security structure may be rendered invalid. Kim et al. [71] developed an ML/DL-based framework potent of identifying botnet flooding attacks in an IoT system. The dataset used for generated by introducing botnets of the likes of Mirai and Bashlite into four smart household devices.

Cañedo and Skjellum [36] trained ANN so as to facilitate identification of man-in-the-middle attack and tampering of the data sent from an edge to the smart object in an IoT system. Amouri et al. [20] put forth a system for identification of blackhole and distributed denial of service attacks. Herein, linear regression is performed by super node for the collected correctly classified instances from dedicated sniffers so as to discern anomalous nodes. Doshi et al. [42] employed flow-based and protocol-agnostic traffic data along with low-cost machine learning algorithms for detection of distributed denial of service attacks in intermediary network nodes. A number of classifiers like KNN, KDTree, linear-kernel support vector machine. A multilevel perceptron using traces of Internet packet was trained by Hodo et al. [59], and was assessed for its ability to forestall DDoS attacks from occurring. Saied et al. [115] employed an ANN for real-time identification of DDoS attacks, both previously discovered and otherwise. Having learnt from training samples, the model was able to detect zero-day attack features. The model's detection probabilities improved as improved datasets containing latest features of known DDoS attacks were used for training purposes.

A framework for intrusion detection and mitigation at the network level was proposed by Nobakht et al. [101] for dealing with potential intrusions in household devices. It employed classifiers like, logistic regression and rbf-kernel support vector machine for identifying compromised hosts. Using convolutional neural networks, Kamel and Elhamayed [66] demonstrated how detection of network attacks, namely

selective forwarding, version, wormhole and sinkhole can minimize the consumption of power in smart healthcare devices. The proposed model simulated an IoT system containing benign and anomalous nodes.

Thamilarasu and Chawla [133] developed an intelligent IDS capable of detecting anomalous behaviour in five variants of network attacks: DDoS, blackhole, sinkhole and wormhole and opportunistic service. By a combination of network virtualization and deep neural network classifier, the model accomplished average recall and precision rate of up to 97% and 95%, respectively, amid varied intrusion scenarios. Aminanto and Kim [18] proposed a completely unsupervised method capable of recognizing impersonation attacks in wireless fidelity networks without prior data label. Aminanto et al. [17] put forth a framework that amalgamated stacked and weighted feature selection for recognizing impersonation attacks like spoofing, man-in-the-middle and replay attack in wireless fidelity networks, an IoT-enabling technology. A neural network was fed the combined features for training. Bostani and Sheikhan [33] developed a confluence of anomaly and specification systems for detecting intrusion in an IoT system. It enabled the identification of sinkhole and selective-forwarding attacks in IPv6 over low-power wireless personal area networks, and its usage can be broadened to recognize other hole attacks like black hole and wormhole alongside rank attacks. A deep learning approach employing SAE with dual and triple hidden layers was proposed by Thing [134]. A table of various ML/DL techniques used for various attacks is given in Table 1.

4 Research Gap and Future Work Challenges

This section elaborates upon the various challenges that come in with using ML and DL-based approaches for achieving security in IoT systems. The main gaps in existing research and the corresponding future routes that can be taken are explained in the following sections.

4.1 Challenges Arising from Data Used in IoT

1. Availability of real-world data
2. Preprocessing required to obtain high-quality data
3. Need for data augmentation for small datasets
4. Need for data fusion from heterogeneous sources
5. Maintain the privacy of users and data protection.

4.2 Challenges with ML and DL Algorithms

ML/DL algorithms have been seen to be very specific to the problems they are solving and one model may not give satisfactory results with other similar problem statements. Hence, more insight is required so as to enhance these algorithms for a wider use. In spite of the tremendous growth in the field of neural networks, they are still black boxes.

Applications like autonomous vehicles, Internet banking, etc., work with real-time inputs and outputs. Such systems have high safety concerns and cannot suffer latency. Moreover, real systems not only experience lag in sensors, actuators and feedback systems, but also encounter inference in real time [75]. Existing ML and DL algorithms do not imitate such real-time response.

The ever-increasing developments in ML and DL fields have encouraged hackers to use them and launch attacks in the system. Studies have shown the use of ML (via SVMs) [57, 77] and DL [85] to break cryptographic systems. RNNs have been shown to learn decryption through algorithmic representations of ciphers for performing cryptanalysis. In [126], a DNN was deceived through perturbation. Generative adversarial networks (GANs) and other models of ML have been shown to be launching attacks on IoT systems.

4.3 Constraints on Resources

The devices used in IoT systems have numerous constraints like those of storage space, computational capacity and such resources that limit intelligent approaches to be implemented in these systems [74]. The use of new technologies like GPUs, provides speed in computational tasks and is important in improving the performance of algorithms [78], but at the same time drains battery power. Cyberattacks like Mirai [31] on low power and resource limited devices direct attention towards security of IoT devices and data generated.

4.4 Trade-Off with Security

There may exist many scenarios where other capabilities of IoT devices must overpower the security mechanisms. For instance, an IoMT system must possess the ability to be accessible in emergency situations without caring about the security. Easy access to a device implanted in a patient might be required in cases of emergency so as to save the person's life. This is why a balance has to be achieved in the design phase itself between the safety of the concerned human and the security of the device [16, 34].

5 Conclusion

With all the ease IoT brings to us by integrating the physical and the virtual world, security concerns with IoT objects and networks have been on the rise. Any breach in the security of IoT systems can put very critical and private information at stake and hence, the safety of IoT is not something one can compromise with. Security techniques using ML and DL are being exploited for enabling security mechanisms in IoT systems, owing to their self-learning and self-optimizing nature.

In this study, a review of the IoT security structure and usage of machine and deep learning in IoT security has been presented. The infrastructure of IoT has been discussed, followed by the diverse threats and challenges posed to the security of IoT system. The paper discusses in detail layer-wise attacks or threats faced by the whole architecture of IoT. This leads to the comprehensive review of existing techniques using ML or DL for the security of IoT.

The survey seeks to deliver a constructive insight to researchers to enable them to understand the needs of IoT security and make useful progress in transforming IoT security from conventional approaches to intelligent end-to-end systems. It is also highlighted that it is crucial to devise solutions that reduce computational complexity while taking utmost care of the security of devices.

References

1. Abdallah A, Shen X (2017) Lightweight security and privacy preserving scheme for smart grid customer-side networks. *IEEE Trans Smart Grid*. <https://doi.org/10.1109/TSG.2015.2463742>
2. Abdmeziem MR, Tandjaoui D (2015) An end-to-end secure key management protocol for e-health applications. *Comput Electr Eng*. <https://doi.org/10.1016/j.compeleceng.2015.03.030>
3. Abeshu A, Chilamkurti N (2018) Deep learning: the frontier for distributed attack detection in fog-to-things computing. *IEEE Commun Mag*. <https://doi.org/10.1109/MCOM.2018.1700332>
4. Abomhara M, Køien GM (2015) Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. *J Cyber Sec Mob*. <https://doi.org/10.13052/jcsm2245-1439.414>
5. Aditya Sai Srinivas T, Manivannan SS (2020) Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm. *Comput Commun*. <https://doi.org/10.1016/j.comcom.2020.03.031>
6. Ahmadi H, Arji G, Shahmoradi L, Safdari R, Nilashi M, Alizadeh M (2019) The application of internet of things in healthcare: a systematic literature review and classification. <https://doi.org/10.1007/s10209-018-0618-4>
7. Ahmed AIA, Ab Hamid SH, Gani A, Khan S, Khan MK (2019) Trust and reputation for Internet of Things: fundamentals, taxonomy, and open research challenges. <https://doi.org/10.1016/j.jnca.2019.102409>
8. Ahmed E, Yaqoob I, Hashem IAT, Khan I, Ahmed AIA, Imran M, Vasilakos AV (2017) The role of big data analytics in Internet of Things. *Comput Netw*. <https://doi.org/10.1016/j.comnet.2017.06.013>
9. Airehrour D, Gutierrez J, Ray SK (2016) Secure routing for internet of things: a survey. *J Netw Comput Appl*. <https://doi.org/10.1016/j.jnca.2016.03.006>

10. Airehrour D, Gutierrez JA, Ray SK (2019) SecTrust-RPL: a secure trust-aware RPL routing protocol for Internet of Things. *Futur Gener Comput Syst.* <https://doi.org/10.1016/j.future.2018.03.021>
11. Akhunzada A, Gani A, Anuar NB, Abdelaziz A, Khan MK, Hayat A, Khan SU (2016) Secure and dependable software defined networks. <https://doi.org/10.1016/j.jnca.2015.11.012>
12. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. *IEEE Commun Surveys Tutorials.* <https://doi.org/10.1109/COMST.2015.2444095>
13. Al-Garadi MA, Mohamed A, Al-Ali AK, Du X, Ali I, Guizani M (2020) A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Commun Surveys Tutorials.* <https://doi.org/10.1109/COMST.2020.2988293>
14. Alaba FA, Othman M, Hashem IAT, Alotaibi F (2017) Internet of things security: a survey. <https://doi.org/10.1016/j.jnca.2017.04.002>
15. Allahham MS, Abdellatif AA, Mohamed A, Erbad A, Yaacoub E, Guizani M (2020) I-SEE: intelligent, secure and energy-efficient techniques for medical data transmission using deep reinforcement learning. *IEEE Internet Things J.* <https://doi.org/10.1109/jiot.2020.3027048>
16. Altawy R, Youssef AM (2016) Security tradeoffs in cyber physical systems: a case study survey on implantable medical devices. *IEEE Access.* <https://doi.org/10.1109/ACCESS.2016.2521727>
17. Aminanto ME, Choi R, Tanuwidjaja HC, Yoo PD, Kim K (2017) Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. *IEEE Trans Inform Forensics Sec.* <https://doi.org/10.1109/TIFS.2017.2762828>
18. Aminanto ME, Kim K (2018) Improving detection of Wi-Fi impersonation by fully unsupervised deep learning. In: *Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and lecture notes in bioinformatics).* https://doi.org/10.1007/978-3-319-93563-8_18
19. Ammar M, Russello G, Crispo B (2018) Internet of Things: a survey on the security of IoT frameworks. *J Inform Sec Appl.* <https://doi.org/10.1016/j.jisa.2017.11.002>
20. Amouri A, Alaparthi VT, Morgera SD (2020) A machine learning based intrusion detection system for mobile internet of things. *Sensors (Switzerland).* <https://doi.org/10.3390/s20020461>
21. Andrea I, Chrysostomou C, Hadjichristofi G (2016) Internet of Things: security vulnerabilities and challenges. In: *Proceedings of IEEE symposium on computers and communications.* <https://doi.org/10.1109/ISCC.2015.7405513>
22. Anu P, Vimala S (2018) A survey on sniffing attacks on computer networks. In: *Proceedings of 2017 international conference on intelligent computing and control, I2C2 2017.* <https://doi.org/10.1109/I2C2.2017.8321914>
23. Aref MA, Jayaweera SK, Machuzak S (2017) Multi-agent reinforcement learning based cognitive anti-jamming. In: *IEEE wireless communications and networking conference, WCNC (2017).* <https://doi.org/10.1109/WCNC.2017.7925694>
24. Asghari P, Rahmani AM, Javadi HHS (2018) Service composition approaches in IoT: a systematic review. *J Netw Comput Appl.* <https://doi.org/10.1016/j.jnca.2018.07.013>
25. Asghari P, Rahmani AM, Javadi HHS (2019) Internet of Things applications: a systematic review. *Comput Netw.* <https://doi.org/10.1016/j.comnet.2018.12.008>
26. Ashibani Y, Mahmoud QH (2017) Cyber physical systems security: analysis, challenges and solutions. *Comput Secur.* <https://doi.org/10.1016/j.cose.2017.04.005>
27. Atzori L, Iera A, Morabito G (2010) The Internet of Things: a survey. *Comput Netw.* <https://doi.org/10.1016/j.comnet.2010.05.010>
28. Azmoodeh A, Dehghantanha A, Choo KKR (2019) Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning. *IEEE Trans Sustain Comput.* <https://doi.org/10.1109/TSUSC.2018.2809665>
29. Bahtiyar Š, Ufuk Çağlayan M (2012) Extracting trust information from security system of a service. *J Netw Comput Appl.* <https://doi.org/10.1016/j.jnca.2011.10.002>

30. Baracaldo N, Chen B, Ludwig H, Safavi A, Zhang R (2018) Detecting poisoning attacks on machine learning in IoT environments. In: Proceedings of 2018 IEEE international congress on internet of things, ICIOT 2018—Part of the 2018 IEEE world congress on services. <https://doi.org/10.1109/ICIOT.2018.00015>
31. Bertino E, Islam N (2017) Botnets and internet of things security. *Computer*. <https://doi.org/10.1109/MC.2017.62>
32. Bose T, Bandyopadhyay S, Ukil A, Bhattacharyya A, Pal A (2015) Why not keep your personal data secure yet private in IoT? Our lightweight approach. In: 2015 IEEE 10th international conference on intelligent sensors, sensor networks and information processing, ISSNIP 2015. <https://doi.org/10.1109/ISSNIP.2015.7106942>
33. Bostani H, Sheikhan M (2017) Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. *Comput Commun*. <https://doi.org/10.1016/j.comcom.2016.12.001>
34. Camara C, Peris-Lopez P, Tapiador JE (2015) Security and privacy issues in implantable medical devices: a comprehensive survey. <https://doi.org/10.1016/j.jbi.2015.04.007>
35. Campioni F, Choudhury S, Al-Turjman F (2019) Scheduling RFID networks in the IoT and smart health era. *J Ambient Intell Humaniz Comput* 10(10):4043–4057. <https://doi.org/10.1007/s12652-019-01221-5>
36. Canedo J, Skjellum A (2016) Using machine learning to secure IoT systems. In: 2016 14th annual conference on privacy, security and trust, PST 2016. <https://doi.org/10.1109/PST.2016.7906930>
37. Chatterjee B, Das D, Maity S, Sen S (2019) RF-PUF: enhancing IoT security through authentication of wireless nodes using in-situ machine learning. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2018.2849324>
38. Chen Z, Ma N, Liu B (2015) Lifelong learning for sentiment classification. In: ACL-IJCNLP 2015—53rd annual meeting of the association for computational linguistics and the 7th erational joint conference on natural language processing of the Asian federation of natural language processing, proceedings of the conference. <https://doi.org/10.3115/v1/p15-2123>
39. Cherry S (2005) Secrets and lies: digital security in a networked world [Books]. *IEEE Spectr*. <https://doi.org/10.1109/mspec.2000.873914>
40. Deng L, Li D, Yao X, Cox D, Wang H (2019) Mobile network intrusion detection for IoT system based on transfer learning algorithm. *Clust Comput*. <https://doi.org/10.1007/s10586-018-1847-2>
41. Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for Internet of Things. *Futur Gener Comput Syst*. <https://doi.org/10.1016/j.future.2017.08.043>
42. Doshi R, Apthorpe N, Feamster N (2018) Machine learning DDoS detection for consumer internet of things devices. In: Proceedings of 2018 IEEE symposium on security and privacy workshops, SPW 2018. <https://doi.org/10.1109/SPW.2018.00013>
43. Elazhary H (2019) Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions. <https://doi.org/10.1016/j.jnca.2018.10.021>
44. Fadlullah ZM, Tang F, Mao B, Kato N, Akashi O, Inoue T, Mizutani K (2017) State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Commun Surveys Tutorials*. <https://doi.org/10.1109/COMST.2017.2707140>
45. Fang H, Wang X, Hanzo L (2019) Learning-aided physical layer authentication as an intelligent process. *IEEE Trans Commun*. <https://doi.org/10.1109/TCOMM.2018.2881117>
46. Fang S, Wang T, Liu Y, Zhao S, Lu Z (2019) Entrapment for wireless eavesdroppers. In: Proceedings of IEEE INFOCOM. <https://doi.org/10.1109/INFOCOM.2019.8737394>
47. Farris I, Taleb T, Khettab Y, Song J (2019) A survey on emerging SDN and NFV security mechanisms for IoT systems. *IEEE Commun Surveys Tutorials*. <https://doi.org/10.1109/COMST.2018.2862350>

48. Faruki P, Bharmal A, Laxmi V, Ganmoor V, Gaur MS, Conti M, Rajarajan M (2015) Android security: a survey of issues, malware penetration, and defenses. *IEEE Commun Surveys Tutorials*. <https://doi.org/10.1109/COMST.2014.2386139>
49. Fatima-Tuz-Zahra, Jhanjhi NZ, Brohi SN, Malik NA (2019) Proposing a rank and wormhole attack detection framework using machine learning. In: *MACS 2019—13th international conference on mathematics, actuarial science, computer science and statistics, proceedings*. <https://doi.org/10.1109/MACS48846.2019.9024821>
50. Gope P, Sidkar B (2019) Privacy-aware authenticated key agreement scheme for secure smart grid communication. *IEEE Trans Smart Grid*. <https://doi.org/10.1109/TSG.2018.2844403>
51. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): a vision, architectural elements, and future directions. *Futur Gener Comput Syst*. <https://doi.org/10.1016/j.future.2013.01.010>
52. Guo X, Lin H, Li Z, Peng M (2019) Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT. *IEEE Internet Things J*. <https://doi.org/10.1109/jiot.2019.2960033>
53. Gusmeroli S, Haller S, Harrison M, Kalaboukas K, Tomasella M, Vermesan O, Wouters K (2009) Vision and challenges for realizing the internet of things
54. Haider SA, Adil MN, Zhao MJ (2020) Optimization of secure wireless communications for IoT networks in the presence of eavesdroppers. *Comput Commun*. <https://doi.org/10.1016/j.comcom.2020.02.027>
55. Hajiheidari S, Wakil K, Badri M, Navimipour NJ (2019) Intrusion detection systems in the Internet of things: a comprehensive investigation. <https://doi.org/10.1016/j.comnet.2019.05.014>
56. Han G, Xiao L, Poor HV (2017) Two-dimensional anti-jamming communication based on deep reinforcement learning. In: *ICASSP, IEEE international conference on acoustics, speech and signal processing—Proceedings*. <https://doi.org/10.1109/ICASSP.2017.7952524>
57. Heuser A, Zohner M (2012) Intelligent machine homicide. https://doi.org/10.1007/978-3-642-29912-4_18
58. Hiromoto RE, Haney M, Vakanski A (2017) A secure architecture for IoT with supply chain risk management. In: *Proceedings of the 2017 IEEE 9th international conference on intelligent data acquisition and advanced computing systems: technology and applications, IDAACS 2017*. <https://doi.org/10.1109/IDAACS.2017.8095118>
59. Hodo E, Bellekens X, Hamilton A, Dubouilh PL, Iorkyase E, Tachtatzis C, Atkinson R (2016) Threat analysis of IoT networks using artificial neural network intrusion detection system. In: *2016 international symposium on networks, computers and communications, ISNCC 2016*. <https://doi.org/10.1109/ISNCC.2016.7746067>
60. Hong T, Liu C, Kadoch M (2019) Machine learning based antenna design for physical layer security in ambient backscatter communications. *Wirel Commun Mob Comput*. <https://doi.org/10.1155/2019/4870656>
61. Huang J, Zhang X, Tan L, Wang P, Liang B (2014) AsDroid: detecting stealthy behaviors in Android applications by user interface and program behavior contradiction. In: *Proceedings of international conference on software engineering*. <https://doi.org/10.1145/2568225.2568301>
62. Hussain F, Hussain R, Hassan SA, Hossain E (2020) Machine learning in IoT security: current solutions and future challenges. *IEEE Commun Surveys Tutorials*. <https://doi.org/10.1109/COMST.2020.2986444>
63. Islam SM, Kwak D, Kabir MH, Hossain M, Kwak KS (2015) The internet of things for health care: a comprehensive survey. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2015.2437951>
64. Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. <https://doi.org/10.1126/science.aaa8415>
65. Jung B, Han I, Lee S (2001) Security threats to Internet: a Korean multi-industry investigation. *Inform Manage*. [https://doi.org/10.1016/S0378-7206\(01\)00071-4](https://doi.org/10.1016/S0378-7206(01)00071-4)
66. Kamel SOM, Elhamayed SA (2020) Mitigating the impact of IoT routing attacks on power consumption in IoT healthcare environment using convolutional neural network. *Int J Comput Netw Inform Sec*. <https://doi.org/10.5815/ijcnis.2020.04.02>

67. Karimipour H, Dinavahi V (2017) Robust massively parallel dynamic state estimation of power systems against cyber-attack. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2017.2786584>
68. Kaur G, Tomar P, Singh P (2018) Internet of things and big data analytics toward next-generation intelligence
69. Kaur N, Verma S, Kavita (2018) A survey of routing protocols in wireless sensor networks. *Int J Eng Technol (UAE)*
70. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J (2019) Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*. <https://doi.org/10.1186/s42400-019-0038-7>
71. Kim J, Shim M, Hong S, Shin Y, Choi E (2020) Intelligent detection of iot botnets using machine learning and deep learning. *Appl Sci (Switzerland)* 10(19):1–22. <https://doi.org/10.3390/app10197009>
72. Kimani K, Oduol V, Langat K (2019) Cyber security challenges for IoT-based smart grid networks. *Int J Crit Infrastruct Prot*. <https://doi.org/10.1016/j.ijcip.2019.01.001>
73. Koliass C, Kambourakis G, Stavrou A, Voas J (2017) DDoS in the IoT: Mirai and other botnets. *Computer*. <https://doi.org/10.1109/MC.2017.201>
74. Lane ND, Bhattacharya S, Georgiev P, Forlivesi C, Jiao L, Qendro L, Kawsar F (2016) DeepX: a software accelerator for low-power deep learning inference on mobile devices. In: 2016 15th ACM/IEEE international conference on information processing in sensor networks, IPSN 2016—Proceedings. <https://doi.org/10.1109/IPSIN.2016.7460664>
75. Lei L, Tan Y, Zheng K, Liu S, Zhang K, Shen X (2020) Deep reinforcement learning for autonomous internet of things: model, applications and challenges. *IEEE Commun Surveys Tutorials*. <https://doi.org/10.1109/COMST.2020.2988367>
76. Leloglu E (2017) A review of security concerns in internet of things. *J Comput Commun*. <https://doi.org/10.4236/jcc.2017.51010>
77. Lerman L, Bontempi G, Markowitch O (2015) A machine learning approach against a masked AES: reaching the limit of side-channel attacks with a learning model. *J Crypto-graph Eng*. <https://doi.org/10.1007/s13389-014-0089-3>
78. Li H, Ota K, Dong M (2018) Learning IoT in edge: deep learning for the internet of things with edge computing. *IEEE Netw*. <https://doi.org/10.1109/MNET.2018.1700202>
79. Li J, Zhao H, Chen X, Chu Z, Zhen L, Jiang J, Pervaiz H (2020) Secrecy wireless-powered sensor networks for internet of things. *Wirel Commun Mob Comput* 2020:1–12. <https://doi.org/10.1155/2020/8859264>
80. Liang N (2020) Security transmission and storage of internet of things information based on blockchain. *IOP Conf Ser Mater Sci Eng* 750:012164. Institute of Physics Publishing. <https://doi.org/10.1088/1757-899X/750/1/012164>
81. Liao RF, Wen H, Chen S, Xie F, Pan F, Tang J, Song H (2020) Multiuser physical layer authentication in internet of things with data augmentation. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2019.2960099>
82. Liu J, Zhang C, Fang Y (2018) EPIC: a differential privacy framework to defend smart homes against internet traffic analysis. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2018.2799820>
83. Lopez J, Roman R, Alcaraz C (2009) Analysis of security threats, requirements, technologies and standards in wireless sensor networks. In: *Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and lecture notes in bioinformatics)*. https://doi.org/10.1007/978-3-642-03829-7_10
84. Machuzak S, Jayaweera SK (2016) Reinforcement learning based anti-jamming with wideband autonomous cognitive radios. In: 2016 IEEE/CIC international conference on communications in China, ICC China 2016. <https://doi.org/10.1109/ICCChina.2016.7636793>
85. Maghrebi H, Portigliatti T, Prouff E (2016) Breaking cryptographic implementations using deep learning techniques. In: *Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and lecture notes in bioinformatics)*. https://doi.org/10.1007/978-3-319-49445-6_1

86. Makhdoom I, Abolhasan M, Lipman J, Liu RP, Ni W (2019) Anatomy of threats to the internet of things. *IEEE Commun Surveys Tutorials*. <https://doi.org/10.1109/COMST.2018.2874978>
87. Makkar A, Kumar N (2020) An efficient deep learning-based scheme for web spam detection in IoT environment. *Future Gener Comput Syst*. <https://doi.org/10.1016/j.future.2020.03.004>
88. Marjani M, Nasaruddin F, Gani A, Karim A, Hashem IAT, Siddiqua A, Yaqoob I (2017) Big IoT data analytics: architecture, opportunities, and open research challenges. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2017.2689040>
89. McLaughlin N, Del Rincon JM, Kang BJ, Yerima S, Miller P, Sezer S, Safaei Y, Trickle E, Zhao Z, Doupe A, Ahn GJ (2017) Deep android malware detection. In: *CODASPY 2017—Proceedings of the 7th ACM conference on data and application security and privacy*. <https://doi.org/10.1145/3029806.3029823>
90. Mendez Mena D, Papapanagioutou I, Yang B (2018) Internet of things: Survey on security. <https://doi.org/10.1080/19393555.2018.1458258>
91. Mikołajczyk A, Grochowski M (2018) Data augmentation for improving deep learning in image classification problem. In: *2018 international interdisciplinary PhD workshop, IIPHDW 2018*. <https://doi.org/10.1109/IIPHDW.2018.8388338>
92. Miorandi D, Sicari S, De Pellegrini F, Chlamtac I (2012) Internet of things: vision, applications and research challenges. <https://doi.org/10.1016/j.adhoc.2012.02.016>
93. Mishra AK, Tripathy AK, Puthal D, Yang LT (2019) Analytical model for sybil attack phases in internet of things. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2018.2843769>
94. Mishra P, Pilli ES, Varadharajan V, Tupakula U (2017) Intrusion detection techniques in cloud environment: a survey. <https://doi.org/10.1016/j.jnca.2016.10.015>
95. Mohammadi M, Al-Fuqaha A, Sorour S, Guizani M (2018) Deep learning for IoT big data and streaming analytics: a survey. <https://doi.org/10.1109/COMST.2018.2844341>
96. Mohammadi S, Mirvaziri H, Ghazizadeh-Ahsae M, Karimipour H (2019) Cyber intrusion detection by combined feature selection algorithm. *J Inform Sec Appl*. <https://doi.org/10.1016/j.jisa.2018.11.007>
97. Moosavi SR, Nguyen Gia T, Rahmani AM, Nigussie E, Virtanen S, Isoaho J, Tenhunen H (2015) 6th international conference on ambient systems, networks and technologies (ANT 2015). SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. *Procedia Comput Sci*
98. Mosenia A, Jha NK (2017) A comprehensive study of security of internet-of-things. *IEEE Trans Emerg Top Comput*. <https://doi.org/10.1109/TETC.2016.2606384>
99. Namvar N, Saad W, Bahadori N, Kelley B (2016) Jamming in the internet of things: a game-theoretic perspective. In: *2016 IEEE global communications conference, GLOBECOM 2016—Proceedings*. <https://doi.org/10.1109/GLOCOM.2016.7841922>
100. Neerugatti V, Reddy ARM (2019) Machine learning based technique for detection of rank attack in RPL based internet of things networks. *Int J Innov Technol Explor Eng*. <https://doi.org/10.35940/ijitee.I3044.0789S319>
101. Nobakht M, Sivaraman V, Boreli R (2016) A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In: *Proceedings of 2016 11th international conference on availability, reliability and security, ARES 2016*. <https://doi.org/10.1109/ARES.2016.64>
102. Nord JH, Koohang A, Paliszkiwicz J (2019) The Internet of Things: review and theoretical framework. <https://doi.org/10.1016/j.eswa.2019.05.014>
103. Nweke HF, Teh YW, Al-garadi MA, Alo UR (2018) Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. <https://doi.org/10.1016/j.eswa.2018.03.056>
104. Ozay M, Esnaola I, Yarman Vural FT, Kulkarni SR, Poor HV (2016) Machine learning methods for attack detection in the smart grid. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2015.2404803>
105. Rana R (2017) Man-in-the-middle attack. *Int J Rec Adv Eng Res*. <https://doi.org/10.24128/ijraer.2017.bc45wx>

106. Rayan Z, Alfonse M, Salem ABM (2018) Machine learning approaches in smart health. *Procedia Comput Sci.* <https://doi.org/10.1016/j.procs.2019.06.052>
107. Razzaque MA, Mилоjevic-Jevric M, Palade A, Cla S (2016) Middleware for internet of things: a survey. *IEEE Internet Things J.* <https://doi.org/10.1109/JIOT.2015.2498900>
108. ur Rehman A, Rehman SU, Raheem H (2019) Sinkhole attacks in wireless sensor networks: a survey. *Wirel Pers Commun.* <https://doi.org/10.1007/s11277-018-6040-7>
109. Ren J, Guo H, Xu C, Zhang Y (2017) Serving at the edge: a scalable IoT architecture based on transparent computing. *IEEE Netw.* <https://doi.org/10.1109/MNET.2017.1700030>
110. Restuccia F, D'Oro S, Melodia T (2018) Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet Things J.* <https://doi.org/10.1109/JIOT.2018.2846040>
111. Riahi Sfar A, Natalizio E, Challal Y, Chtourou Z (2018) A roadmap for security challenges in the Internet of Things. *Dig Commun Netw.* <https://doi.org/10.1016/j.dcan.2017.04.003>
112. Rieback MR, Crispo B, Tanenbaum AS (2006) Is your cat infected with a computer virus? In: Proceedings of fourth annual IEEE international conference on pervasive computing and communications, PerCom 2006. <https://doi.org/10.1109/PERCOM.2006.32>
113. Roman R, Zhou J, Lopez J (2013) On the features and challenges of security and privacy in distributed internet of things. *Comput Netw.* <https://doi.org/10.1016/j.comnet.2012.12.018>
114. Saggi MK, Jain S (2018) A survey towards an integration of big data analytics to big insights for value-creation. *Inf Process Manage.* <https://doi.org/10.1016/j.ipm.2018.01.010>
115. Saied A, Overill RE, Radzik T (2016) Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing.* <https://doi.org/10.1016/j.neucom.2015.04.101>
116. Senigagliales L, Baldi M, Gambi E (2020) Physical layer authentication techniques based on machine learning with data compression. In: 2020 IEEE conference on communications and network security, CNS 2020. <https://doi.org/10.1109/CNS48642.2020.9162280>
117. Sethi P, Sarangi SR (2017) Internet of things: architectures, protocols, and applications. <https://doi.org/10.1155/2017/9324035>
118. Sezer OB, Dogdu E, Ozbayoglu AM (2018) Context-aware computing, learning, and big data in internet of things: a survey. <https://doi.org/10.1109/JIOT.2017.2773600>
119. Shi C, Liu J, Liu H, Chen Y (2017) Smart User authentication through actuation of daily activities leveraging wifi-enabled IoT. In: Proceedings of the international symposium on mobile ad hoc networking and computing (MobiHoc). <https://doi.org/10.1145/3084041.3084061>
120. Shukla P (2018) ML-IDS: a machine learning approach to detect wormhole attacks in Internet of Things. In: 2017 intelligent systems conference, IntelliSys 2017. <https://doi.org/10.1109/IntelliSys.2017.8324298>
121. Sicari S, Rizzardi A, Grieco LA, Coen-Porisini A (2015) Security, privacy and trust in Internet of things: the road ahead. <https://doi.org/10.1016/j.comnet.2014.11.008>
122. Singh A, Payal A, Bharti S (2019) A walkthrough of the emerging IoT paradigm: visualizing inside functionalities, key features, and open issues. <https://doi.org/10.1016/j.jnca.2019.06.013>
123. Spachos P, Papapanagiotou I, Plataniotis KN (2018) Microlocation for smart buildings in the era of the Internet of Things: a survey of technologies, techniques, and approaches. *IEEE Sig Process Mag.* <https://doi.org/10.1109/MSP.2018.2846804>
124. Srivastava S, Singh M, Gupta S (2018) Wireless sensor network: a survey. In: 2018 international conference on automation and computational engineering, ICACE 2018. <https://doi.org/10.1109/ICACE.2018.8687059>
125. Steinhubl SR, Muse ED, Topol EJ (2015) The emerging field of mobile health. <https://doi.org/10.1126/scitranslmed.aaa3487>
126. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. *IEEE Trans Evol Comput.* <https://doi.org/10.1109/TEVC.2019.2890858>
127. Su X, Zhang D, Li W, Zhao K (2016) A deep learning approach to android malware feature learning and detection. In: Proceedings of 15th IEEE international conference on trust, security and privacy in computing and communications, 10th IEEE international conference on big data

- science and engineering and 14th IEEE international symposium on parallel and distributed Proce. <https://doi.org/10.1109/TrustCom.2016.0070>
128. Suma N, Samson SR, Saranya S, Shanmugapriya G, Subhashri R (2017) IOT based smart agriculture monitoring system. *Int J Rec Innov Trends Comput Commun*
 129. Suthaharan S (2014) Big data classification: problems and challenges in network intrusion prediction with machine learning. *Perform Eval Rev.* <https://doi.org/10.1145/2627534.2627557>
 130. Syed NF, Baig Z, Ibrahim A, Valli C (2020) Denial of service attack detection through machine learning for the IoT. *J Inform Telecommun.* <https://doi.org/10.1080/24751839.2020.1767484>
 131. Tahsien SM, Karimipour H, Spachos P (2020) Machine learning based solutions for security of Internet of Things (IoT): a survey. *J Netw Comput Appl.* <https://doi.org/10.1016/j.jnca.2020.102630>
 132. Tarricone L, Grosinger J (2020) Augmented RFID technologies for the internet of things and beyond. *Sensors* 20(4):987. <https://doi.org/10.3390/s20040987>
 133. Thamilarasu G, Chawla S (2019) Towards deep-learning-driven intrusion detection for the internet of things. *Sensors (Switzerland).* <https://doi.org/10.3390/s19091977>
 134. Thing VL (2017) IEEE 802.11 network anomaly detection and attack classification: a deep learning approach. In: *IEEE wireless communications and networking conference, WCNC.* <https://doi.org/10.1109/WCNC.2017.7925567>

A Study on High-Resolution Algorithms MUSIC, MVDR, ESPRIT, Beamscan, and Root-MUSIC for Narrowband Signals



Meenal Job and Ram Suchit Yadav

Abstract This article offers a thorough explanation of several high-resolution direction of arrival (DOA) estimation methods. Here we have analyzed Beamscan, minimum variance distortionless response (MVDR), multiple signal classification (MUSIC), Root-MUSIC, and estimation of signal parameters via rotational invariance techniques (ESPRIT) algorithms. For our study, we consider narrowband signals impinging on the sensors. The operating frequency of the system is 300 MHz. DOA of signal is estimated from the peaks of the output signals. To obtain the spatial spectrum, a traditional beam is formed and scanned over the direction of interest. Beamscan is unable to resolve signals when they come from directions that are closer than the beamwidth. Over the designated area, the MVDR beam is examined and because of its narrower beamwidth, it has a greater resolution. MVDR accurately calculates the signal's DOA. It is vulnerable to positional mistakes; hence in this case, MUSIC offers precise DOA estimate and improved spatial resolution. In the MUSIC algorithm, the DOA is estimated by the search of maxima in the polynomial which causes an increase in computing time, whereas in the Root-MUSIC algorithm the DOA can be estimated by the search for '0' in a polynomial. But this algorithm assumes that linear antennas are uniformly spaced. In the above algorithm, the calculation for the search of the roots of the polynomial can be avoided by the ESPRIT algorithm. In this, rotational invariance property of signal subspace is used for DOA calculation.

Keywords MVDR · MUSIC · ULA · ESPRIT

M. Job (✉) · R. S. Yadav

Department of Electronics and Communication, University of Allahabad, Prayagraj, India
e-mail: meenalmuduli@gmail.com

1 Introduction

The technique of determining the direction of electromagnetic waves or acoustic waves impinging on a sensor array is known as direction of arrival (DOA) estimation [1]. Direction of arrival estimation is used for locating and tracking the signals. DOA has many applications in the field of communication, military, seismology, acoustics, oceanography radar, smart antenna, sonar, and many more [2]. This paper presents a comparison and evaluation for different subspace-based methods. The particular structure of the signal's correlation matrix contains details about the signal propagation model. It is the foundation for the methods that use the concept of subspace [3]. The idea is to divide the data space into the signal subspace and noise subspace. In order to execute these high-resolution approaches based on the signal's decomposition into orthogonal subspaces, the autocorrelation matrix of the observation vector must first be properly analyzed [4, 5]. Bartlett in 1950 tried to present a periodogram analysis of continuous spectra for wireless signals [6]. In 1986, Schmidt proposed the MUSIC algorithm which evaluates the frequency content of the signal using eigenspace method [7]. Roy et al. have come up with a new approach in which the estimation of signal parameters is done by using rotational invariance techniques. Different environments need different modifications in the existing algorithms. Array structure has an essential role in estimating the direction of arrival of the signals [8]. The DOA estimation methods using array antennas are used in many different study disciplines and have attracted a lot of attention in the literature. When sources are uncorrelated or weakly correlated, these approaches have a high resolving power. The benefit is that source locations and network geometry are the sole factors that affect subspaces.

1.1 *Beamscan*

Beamscan is a method for obtaining a spatial spectrum by forming a regular beam and scanning it across desired directions.

1.2 *Multiple Signal Classification*

Schmidt and his associates proposed the Multiple Signal Classification (MUSIC) algorithm in 1979. It is a subspace technique that offers highly accurate DOA estimates. This algorithm provides an estimation for the DOA of signals, number of incoming signals, and the cross-correlation between signals, interferences, and noise [9].

1.3 Minimum Variance Distortionless Method

Another adaptive beamformer approach, called minimum variance distortionless response (MVDR), was published by Capon in 1969. This method can resolve signals that are separated by a small portion of antenna beamwidth [10].

1.4 Root-MUSIC

The basic idea of Root-MUSIC algorithm is to create a polynomial of degree $2(M - 1)$ and retrieve its roots. Eigenvectors of the sensor array correlation matrix serve as the foundation for the Root-MUSIC approach. By looking at the spectrum polynomial's roots, it is able to determine the signals' estimate [11, 12].

1.5 Estimation of Signal Parameters via Rotational Invariant Techniques.

ESPRIT is a method for figuring out the parameters of a mixture of incoming signals against a background of background noise [13].

2 Signal Model

Input sample $S_m(t)$, $m = 1 \dots M$ (Fig. 1)

$$x(t) = \sum_{m=1}^M S_m(t)w_m^* \tag{1}$$

Output sample

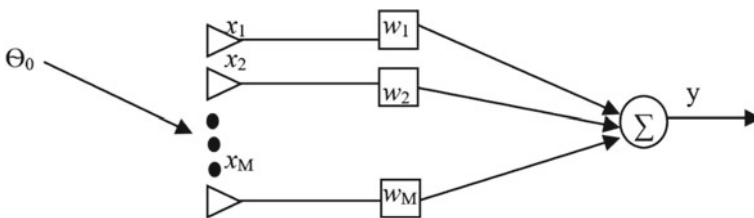


Fig. 1 Block diagram of signal model

$$y(t) = e^{j\omega t} \sum_{m=1}^M e^{-\omega\tau_m} w_m^*, \quad (2)$$

where * indicates complex conjugate.

Let the received signal by the first sensor is $S_1(t)$ with zero phase. Then the signal received by m th sensor is given by

$$s_m(t) = e^{j\omega(t-\tau_m)}. \quad (3)$$

Here, τ_m is propagation delay and $m = 1, 2, 3, \dots, M$.

$$x(t) = e^{j\omega t} \sum_{m=1}^M e^{-\omega\tau_m} w_m^*. \quad (4)$$

Response of narrow band beamformer at $\tau_0 = 0$ is given by:

$$P(\omega, \Theta) = \sum_{m=1}^M e^{-\omega\tau_m} w_m^* = W^H d(\omega, \Theta), \quad (5)$$

where weight vector is given as:

$$W = [w_1 w_2 w_3 \dots w_m]^T. \quad (6)$$

Array response vector $d(\omega, \theta)$ is given as:

$$d(\omega, \Theta) = [1 e^{-j\omega\tau_1} e^{-j\omega\tau_2} \dots e^{-j\omega\tau_{M-1}}]^T. \quad (7)$$

By substituting a constant phase shift for the phase shift in the traditional beamformer's response, the discrete Fourier transform (DFT) yields the following response:

$$2\pi f t_{mb} = 2\pi f_0 t_{mb}, \quad (8)$$

where signal center frequency is f_0 . Applying DFT to each sensor output $x_m(n)$ will result in frequency transformed data,

$$S_m(k) = \sum_{n=1}^0 S_m(n) e^{j2\pi nk/N}, \quad (9)$$

where $0 \leq k \leq N - 1$, $f_k = Kf/N$, where $S_m(K)$ is an estimate of $S_m(f)$.

Conventional beamformer output response in frequency domain is given by

$$x(f, \varphi_b) = \sum_{m=1}^M a_m s_m(f) e^{-j\omega_0 t_{mb}}. \quad (10)$$

Inverse Fourier transform of the above equation gives the output of conventional beamformer in time domain given by:

$$x(t, \varphi_b) = F^{-1}[Y(f, \varphi_b)] = \sum_{m=1}^M a_m s_m(t) e^{-j\omega_0 t_{mb}}. \quad (11)$$

MVDR Algorithm

Minimum variance distortionless response beamformer is capable of determining the weight vectors for beam steering

$$x = \omega^H s. \quad (12)$$

Output power of array is

$$p = \{E|y|^2\} = E\{\omega^H s s^H \omega\} = \omega^H R. \quad (13)$$

Here, R stands for the received signal covariance matrix, and H stands for Hermitian transposition. The optimal weights are chosen, and the unity gain is kept in the direction of the desired signal, reducing the array output power of MVDR. The MVDR Beamformer algorithm is described by

$$\min_w \{w^H R w\} \text{subject to } w^H a(\theta),$$

where $a(\theta)$ is the steering vector given by:

$$a(\theta) = \left(\frac{\exp\{j \frac{2\pi}{\lambda} (\sin \theta) d\}}{\exp\{j \frac{2\pi}{\lambda} (\sin \theta) (m-1) d\}} \right). \quad (14)$$

In this case, d stands for the array element spacing, θ_i for the required angle, and m for the total number of elements. As shown by the optimisation weight vector,

$$W_{\text{MVDR}} = \frac{R^{-1} a(\theta)}{a^H(\theta) R^{-1} a(\theta)}. \quad (15)$$

Consider the signal $S_1(t), S_2(t), \dots, S_k(t)$ at frequency f_0 with direction of arrival of $q_k (k = 1, \dots, K)$ and the received by ULA of m sensors $M > K$.

The received signal is given by

$$x(t) = \sum_{k=1}^k a(\Theta_k) s(t) + n(t), \quad (16)$$

where $a(\Theta_k) = [e^{-j\Phi_{1,k}}, e^{-j\Phi_{2,k}}, \dots, e^{-j\Phi_{M,k}}]^T$.

$$\Phi_{m,k} = 2\pi/\lambda(m-1).d \sin(\Theta_k) \text{ with } m = 1 \dots M$$

T is the transpose, $a(\Theta_k)$ is the steering vector, k th signal depends upon angle of arrival, and $n(t)$ is Gaussian vector of noise.

The matrix notation of signal is given by

$$X = AS + \eta, \quad (17)$$

where $A = [a(\Theta_1), a(\Theta_2), \dots, a(\Theta_k)]$, $S = [S_1(t), S_2(t), \dots, S_K(t)]^T$ and $\eta = [n_1(t), n_2(t), \dots, n_m(t)]^T$.

The correlation matrix is given by,

$$R_{XX} = E\{XX^H\} = AR_{SS}A^H + R_{nn} \quad (18)$$

Noisy observation = Signal space + Noise space,

where X^H is conjugate transpose of X , $R_{nn} = \sigma^2 I$, where I is the identity matrix and σ^2 is the noise powers of square matrix $R_{SS} = \{SS^H\}$, R_{SS} is diagonal of full rank.

Correlation matrix is estimated by an average over N observation.

$$R_{xx} = N^{-1}.XX^H, \quad (19)$$

where N is the number of observation vectors, and $N \times K$ is the complex envelope matrix of k signals.

Music Algorithm

It is a subspace-based algorithm, vectors derived from EN generate a noise subspace orthogonal to steering vectors of these sources, and vectors derived from Es generate a signal subspace collinear with steering vector of sources $a(\Theta_k)$.

$$E_N^H . a(\Theta_k) = 0 \text{ for } k = 1, 2, \dots, k. \quad (20)$$

To find DOA of signal, it is necessary to diagonalize the covariance matrix

$$d^2 = a(\Theta)^H E_N E_N^H a(\Theta) = 0 (E_N = [e_1, e_2, \dots, e_{M-K}]). \quad (21)$$

$C = E_N . E_N^H$ is the projection matrix, $a(\Theta)^H E_N E_N^H a(\Theta)$ is projection of vector $a(\Theta)$ on noise subspace.

Estimating DOA is equivalent to MUSIC pseudospectrum $p(\Theta)$.

$$P_{\text{MUSIC}}(\Theta) = 1/a(\Theta)^H E_N E_N^H a(\Theta) \quad (22)$$

Root-MUSIC algorithm

$$P_{\text{MUSIC}}^{-1}(\Theta) = a(\Theta)^H E_N E_N^H a(\Theta) \quad (23)$$

$$P_{\text{MUSIC}}^{-1}(\Theta) = a(\Theta)^H \cdot C \cdot a(\Theta) \quad (24)$$

Steering vector $a_m(\Theta) = e^{j2\pi/\lambda d(m-1) \sin \Theta}$.

We can write,

$$P_{\text{MUSIC}}^{-1}(\Theta) = \sum_{m=1}^M 1 \sum_{m=1}^M \exp \exp \left(-\frac{j2\pi}{\lambda} (m-1)d \sin \Theta \right) C_{mn} \cdot \exp(-j2\pi/\lambda(ld \sin \Theta)), \quad (25)$$

where $C_l = \sum_{m-n=l} C_{mn}$.

This can be transformed into Root-MUSIC polynomial

$$D(z) = \sum_{l=-m+1}^{M+1} C_l z^l, \quad (26)$$

where $z = e^{-j2\pi/\lambda \sin \Theta}$.

So the angle of arrival of signal is given by $\Theta_m = -\sin((\lambda/2\pi d) \arg(z_m))$.

3 Modeling Parameters

See Table 1.

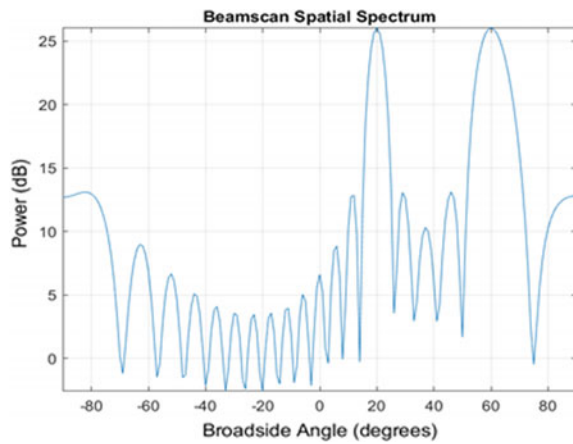
4 Simulation and Result

Here, we'll attempt to estimate the signals' direction of arrival. The peak of the output spatial spectrum, for each technique, shows the direction in which the signals were received. In these algorithms, Beamscan, MVDR, MUSIC, Root-MUSIC, and ESPRIT are analyzed. For the simulation, we have considered that two signals arrive at 20° azimuth, 60° azimuth, and with 0° elevation for both signals. It is a uniform linear array (ULA) of 20 elements with the spacing of $\lambda/2$.

Table 1 Simulation parameters for the beamformer

System parameter	Value
Geometry of the antenna	Ula
Kind of antenna	Isotropic
Operating frequency	300e6
Beam scanning range	+ 90
No. of elements	20 elements
Element spacing	$\lambda/2$
Noise power	0.01
N sample	2000
Elevation and azimuth angle	Variable

Fig. 2 Beamscan spatial spectrum



4.1 Beamscan Direction of Arrival Estimation with ULA

For the Beamscan algorithm, we have seen that the broadside side angle estimate is 20° azimuth and 60° azimuth. Hence, the result of the broadside angle is the same as the azimuth angles (Fig. 2).

4.2 Drawback of Beamscan

If the two incoming signals are very close to each other, then the Beamscan algorithm fails to detect that. For the signals with 30° azimuth and 35° azimuth, the resultant angles are different from the incoming azimuth angles, i.e., 32° and 46° (Fig. 3).

Fig. 3 When the incoming signals are very close

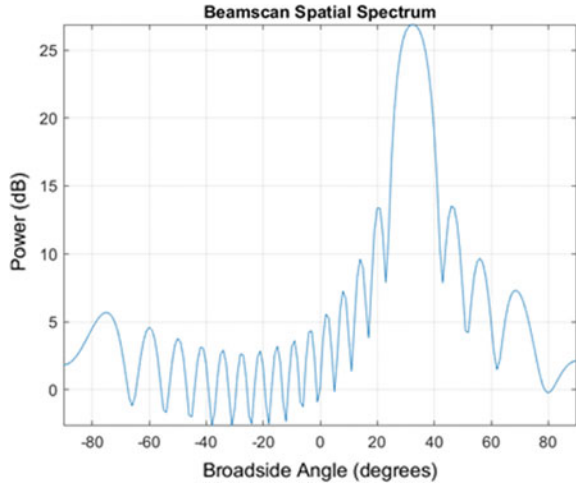
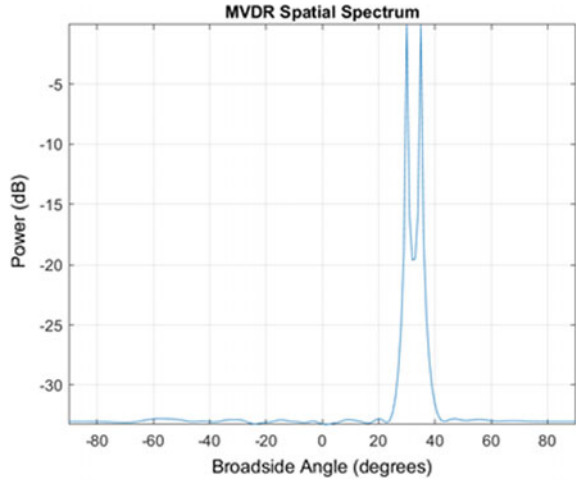


Fig. 4 MVDR output



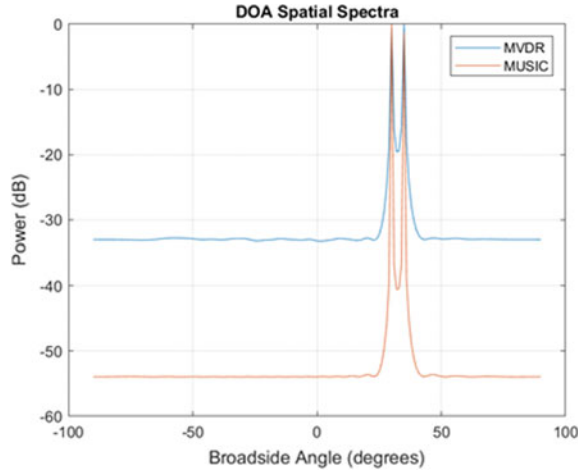
4.3 MVDR Algorithm

MVDR algorithm has shown that the angles have at 30° azimuth and 35° azimuth and both have 0° elevation (Fig. 4).

4.4 Drawback of MVDR

- Sensor position inaccuracy affects MVDR.

Fig. 5 Comparison of MUSIC and MVDR algorithm



- A reliable estimate of the outcome cannot be obtained when the difference between the two signal directions is reduced to a level below the beamwidth of an MVDR beam.

4.5 *MUSIC Algorithm.*

By the comparison of MVDR and MUSIC algorithms, we concluded that the MUSIC algorithm gives more accurate results (Fig. 5).

4.6 *Root-MUSIC Algorithm.*

For the arriving signal at 20° azimuth and 60° azimuth angles, the estimated angle is 60.0026° and 19.9999°.

4.7 *ESPRIT Algorithm.*

For the arriving signal at 20° azimuth and 60° azimuth angle, the estimated angle is 60.0058° and 20.0003°.

5 Conclusion

A conventional beam is produced and scanned across the direction of interest to acquire the spatial spectrum. Beamscan is unable to resolve signals when they come from directions that are closer together than the beamwidth. The MVDR beam is evaluated over the defined area and has a better resolution due to its lower beamwidth. MVDR determines the signals DOA with accuracy. Since it is prone to positional errors, MUSIC in this situation provides a precise DOA estimate and better spatial resolution. In the MUSIC algorithm, the DOA is calculated by looking for polynomial maxima, which increases computation time; however in the Root-MUSIC approach, the DOA can be determined by looking for polynomial '0'. But this approach counts on the uniform spacing of linear antennas. The ESPRIT algorithm can be used in the above algorithm to avoid the calculation required to find the polynomial's roots. This DOA calculation makes use of the signal subspace's rotational invariance characteristic.

References

1. Godara LC (1997) Applications of antenna arrays to mobile communications. Proc IEEE 85:1031–1060, July 1997 and Part II. Proc IEEE 85(8):1195–1244, Aug 1997
2. Liao B, Chan SC (2011) DOA estimation of coherent signals for uniform linear arrays with mutual coupling. In: 2011 IEEE international symposium on circuits and systems (ISCAS), 15–18 May 2011, Rio de Janeiro, Brazil
3. Zhu J, Chan M, Hwang HK (2003) Simulation study on adaptive antenna array. In: IEEE international signal processing conference, Dallas
4. Grice M, Rodenkirch J, Yakovlev A, Hwang HK, Aliyazicioglu Z, Lee A (2007) Direction of arrival estimation using advanced signal processing. In: RAST conference, Istanbul-Turkey
5. Skolnik M (2001) Introduction to RADAR systems, 3rd edn. Mc Graw Hill, New York
6. Bartlett MS (1950) Periodogram analysis and continuous spectra. Biometrika 37:1–16
7. Schmidt R (1986) Multiple emitter location and signal parameter estimation. IEEE Trans Antennas Propag 34:276–289
8. Roy R, Paulraj A, Kailath T (1985) Estimation of signal parameters via rotational invariance techniques—esprit. In: Proceedings of the nineteenth Asilomar conference on circuits, systems and computers, Pacific Grove, CA, USA, 6–8 Nov 1985, pp 41.6.1–41.6.5.
9. Osman L, Sfar I, Gharsallah A (2012) Comparative study of high-resolution direction-of-arrival estimation algorithms for array antenna system. 2(1):72–77
10. Xiao Y, Yin J, Qi H, Yin H, Hua G (2017) MVDR algorithm based on estimated diagonal loading for beamforming. Math Prob Eng. <https://doi.org/10.1155/2017/7904356>
11. Hwang HK, Aliyazicioglu Z (2009) Direction of arrival estimation using a phase array antenna. In: Lecture notes in electrical engineering, 33 LNEE, pp 205–219. https://doi.org/10.1007/978-1-4020-9532-0_16
12. Roy R, Kailath T (1989) ESPRIT-estimation of signal parameters via rotational invariance techniques. IEEE Trans Acoust Speech Sig Process 37(7):984–995
13. Zhou L, Huang D, Duan H, Chen Y (2011) A modified ESPRIT algorithm based on a new SVD method for coherent signals. In: IEEE international conference on information and automation (ICIA), Shenzhen, China, 6–8 June 2011, pp 75–78

TwT: A Texture weighted Transformer for Medical Image Classification and Diagnosis



Mrigank Sondhi , Ayush Sharma , and Ruchika Malhotra 

Abstract Medical imaging is an integral part of disease diagnosis and treatment. However, interpreting medical images can be time-consuming and subjective, making it challenging for healthcare professionals. Recent advances in deep learning show promising results in automating medical image classification and diagnosis. In this paper, we explore the application of texture-features, Central Difference Convolution (CDC) enhanced Convolutional Neural Networks (CNNs) and Compact Convolutional Transformers (CCT) to medical image classification and diagnosis. We compared the performance of existing architectures and our proposed Texture weighted Transformer (TwT) architecture. We evaluate each model's performance and develop a robust architecture. Our results show that TwT outperforms other existing models in terms of Accuracy (ACC), Area Under the Receiver Operating Characteristic Curve (AUC) and other metrics. Our model combines texture-features with the advantages and performance of CDC-enhanced CNNs and the CCT architecture. Our proposed architecture gave an AUC of 0.9941 and an ACC of 96.84% on the Malaria dataset and an AUC of 0.9933 and an ACC of 92.75% on the Blood-MNIST dataset while being compact (only about 6.3M parameters) and without any pre-training, and at the same time beating the AUC, ACC and other scores of other existing models proving that transfer learning is not always necessary. Our proposed architecture required less training time than most existing architectures, making it more practical for real-world applications. Our findings suggest that TwT can revolutionise medical image analysis by providing accurate and efficient diagnoses of diseases. The proposed architecture can be extended to other medical imaging tasks, including cancer detection, diabetic retinopathy and COVID-19 diagnosis. Thus, it can help healthcare professionals make accurate and timely diagnoses, improving patient outcomes.

M. Sondhi (✉) · A. Sharma · R. Malhotra
Department of Software Engineering, Delhi Technological University, New Delhi, India
e-mail: mriganksondhi@outlook.com

A. Sharma
e-mail: ayush.sharma280301@gmail.com

R. Malhotra
e-mail: ruchikamalhotra2004@yahoo.com

Keywords Texture · Central Difference Convolution · Vision Transformers · Compact Convolutional Transformers · Medical Image Classification

1 Introduction

Medical imaging has become essential to modern health care, allowing physicians to diagnose and treat various diseases. Automated medical image analysis has become a reality with recent advances in Computer Vision (CV) and Deep Learning (DL). CV offers a faster and more accurate analysis of medical images. Automated analysis has been used to diagnose various diseases, including Cancer, Alzheimer's, Diabetic Retinopathy and COVID-19. CV has the potential to enable remote and decentralised medical imaging. Models such as Vision Transformers (ViT) and Convolutional Neural Networks (CNNs) have shown remarkable performance in medical image classification and diagnosis. These, however, require a lot of data and computational resources. To address these challenges, recent research has focused on developing more efficient and scalable DL models, such as Compact Convolutional Transformers (CCT) [1]. In this paper, we explore the application of texture-features, Central Difference Convolution (CDC) [2] enhanced CNNs and CCT with a focus on the Malaria dataset [3] and BloodMNIST dataset [4, 5]. We compare our proposed architecture with existing models, such as different CNNs and ViT [6], on various metrics and demonstrate our Texture weighted Transformer's (TwT's) superior performance and efficiency.

CNNs have shown remarkable results. However, they lack computational efficiency, scalability and performance (CNNs are not always accurate). This is mainly because of the inability of the traditional CNN architectures to extract context from the image data. This problem is solved by the "attention-mechanism" in the transformer architecture, which helps capture global contextual relationships between image segments, enabling the model to learn more complex feature representations. The transformer architecture has recently been applied to CV tasks, giving rise to the ViT [6] architecture. ViT has self-attention layers and no convolutions, enabling the model to attend to different parts of the input image and capture global spatial relationships. ViT has shown remarkable performance on various CV tasks. However, limitations exist, particularly its computational efficiency and scalability, making it impractical for real-world applications. To address these challenges, recent research has focused on developing more efficient and scalable DL models that combine the strengths of both CNNs and transformers. CCT is one such model that combines the efficiency of CNNs with the performance of the transformer architecture. CCT has shown superior performance and efficiency on various CV tasks, including object recognition and image classification, and has the potential to revolutionise medical image analysis.

CCT follows an architecture consisting of a Convolutional Tokenizer [1] with a series of compact Transformer-Encoder layers [1] and a Sequence Pooling [1] mechanism. The Convolutional Tokenizer component of CCT introduces inductive

biases and captures low-level features and spatial relationships in the input image. In contrast, the transformer component captures global dependencies and contextual information of the tokens produced by the Convolutional Tokenizer. The original [cls] token [6] of the ViT has been exchanged with the Sequence Pooling mechanism, an attention-based method used to pool over the output sequence of tokens from the Transformer-Encoder blocks. The hybrid architecture allows CCT to effectively capture the structure and content of images, making it a powerful tool for medical image analysis. CCT has shown superior performance over traditional CNNs and pure transformer-based models in various CV tasks [1]. It has a much lower computational cost and memory requirements than pure transformer-based models [1].

In our research,

- We combined a CDC-enhanced CNN-backbone with a CCT-backbone into a dual-branch architecture, with one branch extracting texture-features and the other extracting spatial features. The model is explained in detail in Sect. 3.
- We try to answer three burning questions through our model; can the introduction of texture-features and convolutions to transformers allow them to; firstly, achieve competitive scores on small medical imaging datasets when trained from scratch without pre-training; secondly, not depend on huge amounts of data (data is usually scarce in medical imaging), i.e. not be data hungry and still perform well; and thirdly be compact (with fewer parameters) and still give good results?
- We propose TwT, a powerful tool for medical image analysis that can potentially revolutionise medical image classification by giving quick and accurate results while being compact, efficient and less computationally complex compared to existing approaches and achieve competitive results using it on the task of disease detection on two publicly available datasets, i.e. Malaria and BloodMNIST.

2 Literature Review

This section outlines the existing research in Medical Image Classification and Diagnosis.

2.1 Medical Image Classification

Medical image classification is a crucial task in various medical applications, and recent advancements in deep learning have shown promising results in this field. Due to large-scale medical image datasets being available and the development of advanced CNNs, several studies have proposed novel approaches for accurate and efficient medical image classification. For instance, Sikkandar et al. combined an Adaptive Neuro-Fuzzy Classifier model with a GrabCut algorithm to propose a classification model which was based on segmentation for the diagnosis of skin lesions. The preprocessing step in the model used a Top hat filter and an inpainting technique

[7]. Wang et al. proposed a novel method for detecting COVID-19 using deep learning on chest X-rays, leveraging a deep CNN to differentiate between bacterial pneumonia, COVID-19 and typical cases with high accuracy [8]. Similarly, Ozturk et al. proposed a deep learning model for COVID-19 detection using CT scans, achieving high sensitivity and specificity in classifying cases into the ones with COVID-19 and the ones without COVID-19 [9]. Ryu et al. proposed a CNN to jointly classify and segment a hepatic lesion, in which the network consisted of two inference branches used for both tasks and a single shared encoder [10]. Daanouni et al. proposed a self-attention mechanism clubbed with a pre-trained MobileNet network to accurately predict a patient with Diabetic Retinopathy [11]. Rajaraman et al. in [3] collected, standardised and proposed the Malaria dataset and evaluated the performance of CNN-based models on it. Nguyen et al. in [12] introduced MonoNet with monotonic relationships between (high-level) outputs and features ensured using monotonically connected layers. It is trained on classification tasks, and the performance achieved is reported on several datasets, including BloodMNIST.

2.2 Architectures

Vaswani et al. proposed in their revolutionary paper “Attention is All You Need” [13] an architecture that did away with recurrent networks. The model was purely based on attention and was called the Transformer; it didn’t even use Convolutions and quickly became the state of the art. Alexey Dosovitskiy et al., in [6], completely did away with Convolutions in CV tasks and developed a pure self-attention based transformer approach for the same. The proposed model—ViT achieves excellent results on classification tasks but has to be pre-trained on huge amounts of data. Hassani et al., in [1], proposed three novel Transformer architectures. Firstly, ViT-Lite, which can be trained from scratch and achieves high accuracy on datasets such as CIFAR-10. Secondly, Compact Vision Transformer (CVT), which uses a novel sequence pooling strategy that tries to replace the original [cls] token in ViT, and finally, the Compact Convolutional Transformer (CCT), which increases performance and provides flexibility for input image sizes and at the same time does not depend as much on Positional Embedding compared to the other architectures.

3 Proposed Model

This section describes the proposed architecture: TwT. A pictorial representation of the model is shown in Fig. 1. Our model consists of a dual-branch architecture where the first Texture-Branch extracts the texture-features of the given RGB image using a Central Difference Convolution Network++ (CDCN++)—backbone and the second Spatial-Branch extracts spatial-features using a CCT-backbone. The CDCN++ - backbone extracts the texture-features of the RGB image using the Central

Difference Convolution (CDC) operator [14] enhanced CNN. These texture-features are then passed through a fully-connected (FC) layer (after flattening) and a sigmoid layer to convert them into a weight-vector. This weight-vector is then passed through two dense layers (fully-connected layers) to get the first predicted output. The CCT model first produces Convolutional Tokenisations of the input RGB image by passing it through a simple Convolutional Block. Then, optional positional embeddings are introduced (sinusoidal in our proposed architecture), followed by seven blocks of the Transformer-Encoder. Each Transformer-Encoder block output is automatically weighed using the weight-vector calculated through the first branch. Thus, every spatial-feature output is weighed using texture-features. The final output sequence of tokens from the Transformer-Encoder blocks is pooled over using an attention-based method called Sequence Pooling. This pooled output is then sent through a Multi-Layer Perceptron (MLP) head for classification to get the second predicted output. The CDCN++ branch extracts texture-features through the help of the CDC operator, which provides a stationary description of texture difference information by combining pixel gradient and intensity [14]. At the same time, the CCT block captures local (using Convolutions) and global spatial features (using Transformer-Encoder). As explained in [15], Convolutions are high-pass filters, while Multi-head Self-Attentions (MSAs) are low-pass filters. Thus, they are complementary and help the model extract the best features of an image. We also tried to find an answer to the three questions listed above through our proposed architecture. Thus, our model was trained from scratch on small medical imaging datasets using no pre-trained weights and without using a large number of parameters. The results are highlighted in Sect. 5, making our proposed model—TwT—a powerful tool for medical image classification.

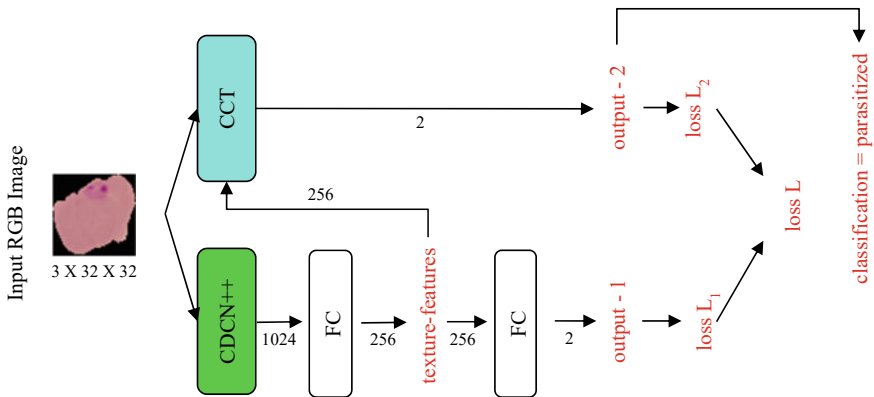


Fig. 1 Proposed texture weighted transformer

3.1 *Texture-Branch*

The first branch of our proposed architecture for medical image classification and diagnosis is built on the CDCN++ architecture. As proposed in the original paper [2], this architecture consisting of the CDC operator, a Neural Architecture Search—based search backbone and a Multiscale Attention Fusion Module (MAFM) is highly effective on the Face Anti-Spoofing (FAS) task and, as experimented in this paper, this backbone proves to be effective on our task, i.e. medical image classification as well. The CDC operator taking inspiration from the Local Binary Pattern (LBP), which is useful for describing local relations in a binary central difference way, is shown to enhance the generalisation and representation ability of the original Convolution operator [2]. It follows two steps: Sampling, similar to the original Convolution operator and Aggregation, in which the centre-oriented gradient of sampled values is aggregated [2]. The CDC operation introduces a hyperparameter $\theta \in [0, 1]$, which can be used to manipulate and weigh the contributions of intensity-level and gradient-level information (i.e. Original Convolution and CDC) [2]. The Central Difference Convolution Network (CDCN) architecture [2] then uses the CDC operator to extract low, mid and high level fused features to predict the grayscale facial depth for the FAS task. The CDCN++ architecture then develops upon the CDCN architecture. The NAS-based search backbone searches for cells in low, mid and high levels to form a network backbone for the FAS task. These multi-level cells are then freely searched, which makes the process more flexible and generalised. Further, to learn more discriminative features, these low, mid and high level features are refined and fused using spatial attention through MAFM, which uses the original Convolution operator with kernel sizes 7×7 , 5×5 and 3×3 for the low, mid and high level features, respectively, as CDC has a limited capacity for global semantic cognition [2]. The CDCN++ - backbone thus extracts the texture-features of the RGB image, as explained above, which are then used to weigh the Transformer-Encoder block outputs. The final output from this Texture-Branch is then passed through linear layers first to get a 256-dimensional weighted vector and finally the desired (number of classes)—dimensional output, which is then used to calculate the loss L_1 , which is added to the loss calculated by the Spatial-Branch L_2 as explained below. This branch is only used to extract texture-features and is not used in the final classification during inference. The CDCN++ branch has about 2.2 million parameters, followed by 0.3 million parameters in the linear layers.

3.2 *Spatial-Branch*

This branch encompasses a CCT-backbone with an MLP classification head on top with no pre-trained weights. It first tokenizes the RGB image using a simple Convolution Block (CDC is not used), with one convolution layer with three input channels and 256 output channels and a kernel ($size = 3$), followed by a ReLU and

maxpool layer. This is done to embed local features and inductive biases into the feature vectors. As the convolution and maxpool operations overlap, these improve the model's performance as they are responsible for introducing inductive biases. These information-rich convolved tokens are then encoded with sinusoidal positional embeddings to add positional information to them. Then, these tokens are sent through seven Compact Transformer-Encoder blocks each with four heads of the Multi-head Self-Attention (MSA) and 256 embedding dimensions. A Feed-Forward Neural Network (FFN) with an MLP ratio of 2 follows the MHSA. Dropout, Layer Normalisation and GELU activation are also applied by the encoder. The encoder helps capture the global information and relationships between each token. Each 256-dimensional token in the output from each encoder is then weighed (element-wise multiplication) using the texture-feature 256-dimensional weight-vector obtained through the first branch. This is done to introduce texture-rich information into the tokens, which proves to be very useful in making them information-rich tokens for classification, as texture helps to identify objects or regions of interest in an image. The output sequence from the Transformer-Encoder blocks is then pooled using Sequence Pooling, an attention-based method. Instead of adding a learnable [cls] token and increasing computation, Sequence Pooling preserves the information in different parts of the input image at no additional parameters. The output of this Sequence Pooling layer is then passed through an FC layer to get the final classification output which is used to calculate the loss L_2 , which is added to the loss L_1 calculated by the Texture-Branch above to get the final loss L .

$$L = L_1 + L_2 \quad (1)$$

This loss L is then propagated backwards into the network. This branch is used to give the final prediction output. This branch has about 3.7–3.8 million parameters (depending on the dataset).

4 Research Methodology

The whole experimentation process followed to achieve the results is highlighted below. The main aim was to develop an architecture that involved both convolutions and transformers so that the model encompasses the properties of both and, at the same time, is less complex and does not have a lot of parameters. Several existing transformer-based approaches were tried, and a CCT-backbone was chosen as part of the architecture. CCT has several variants; thus, several experiments were conducted to choose the best one. The CCT-backbone introduces convolutions only at the beginning as a Convolutional Tokenizer. As explained in [15], CNNs are high-pass filters, i.e. they allow only high frequencies, such as edges, to pass through them, while MSAs are low-pass filters, i.e. they only allow the low frequencies in an image to pass through them. Thus, another part of the architecture purely based on Convolutions was needed to introduce the desired properties and inductive biases into the

model. Therefore, the CDCN++ architecture was chosen, which extracts the texture-features of an RGB image using CDC. The various CCT-backbone variants, along with the CDCN++ - backbone, were compared across several performance metrics. The results helped the authors come up with the proposed architecture—TwT.

4.1 Dataset Description

Malaria. The Malaria dataset [3] consists of a total of 27,558 cell images and is balanced. The data has two class labels (0: Parasitized) and (1: Uninfected). Each cell image is of the shape $3 \times H \times W$ pixels with varying Height and Width values; thus, every image was resized to $3 \times 32 \times 32$ pixels. The data was split into 80:10:10 for training, validation and testing; i.e. 22,046, 2756 and 2756 images, respectively. The batch size for this dataset was taken as 128.

BloodMNIST. The BloodMNIST dataset [4, 5] consists of individual normal cells. The number of images in this dataset stands at 17,092, and the data contains eight classes in total. Images of size $3 \times 28 \times 28$ were obtained from $3 \times 360 \times 363$ pixel original images by centre-cropping them into $3 \times 200 \times 200$ pixels and then resizing. Both mean and standard deviation of 0.5 were used to normalise the images. The data was split into 70:10:20 for training, validation and testing; i.e. 11,959, 1712 and 3421 images, respectively. The batch size for this dataset was taken as 128.

4.2 Compact Convolutional Transformer-Backbone Variants

The notation for the CCT-backbone variants is as follows: CCT-7/3X1, where “7” denotes the number of Transformer-Encoder layers; “1” denotes the number of layers in the Convolutional Tokenizer with a Kernel Size of “ 3×3 ” [1]. The Malaria dataset was trained and tested on the following variants: CCT-7/3X1, CCT-7/3X2, CCT-14/3X1 and CCT-14/7X2 [1] along with the presence and absence of the CDCN++ - backbone from scratch without any pre-trained weights and with a sinusoidal position embedding to boil down to the best architecture. The results for the same are mentioned in Sect. 5. The best architecture was then tested on Malaria and BloodMNIST and gave competitive results on both datasets without utilising a huge number of parameters.

4.3 Hyperparameter Settings

As mentioned, the batch size was 128 for both datasets. A sinusoidal positional embedding was utilised for the CCT-backbone. The Loss function used was Cross-Entropy Loss. AdamW was used as the optimiser. The learning rate was set at 0.001, while all other parameters of AdamW took their default values. All experiments on the Malaria dataset were run for 100 epochs, and all experiments on the BloodMNIST dataset were run for 30 epochs.

4.4 Experimental Setup

All models were trained on two parallel NVIDIA TITAN RTX graphic cards of 24 GB VRAM each on a system with a RAM of 128 GB.

5 Results and Discussion

This section highlights the results obtained on the Malaria and BloodMNIST datasets. Table 1 (the best scores are highlighted in bold) mentions the results obtained on the Malaria dataset using the previously mentioned CCT-backbone variants and the presence and absence of the CDCN++ - backbone. All models in Table 1 were trained from scratch without pre-trained weights with a sinusoidal position embedding. A total of 8 models were trained and tested across two metrics, AUC and Accuracy. The best architecture after experimentation was the CCT-7/3X1 backbone: with a sinusoidal positional embedding and no pre-trained weights clubbed with the CDCN++ - backbone with a $\theta = 0.7$. This is the proposed architecture of this paper. The proposed architecture was then trained and tested on the Malaria and BloodMNIST dataset with and without pre-trained weights.

Table 1 Performance of different CCT variants and CDCN++ on the Malaria dataset

CCT variant	CDCN++	Accuracy (%)	AUC
<i>CCT-7/3X1</i>	Absent	96.11	0.9905
<i>CCT-7/3X1</i>	Present	96.84	0.9941
<i>CCT-7/3X2</i>	Absent	95.64	0.9899
<i>CCT-7/3X2</i>	Present	96.13	0.9904
<i>CCT-14/3X1</i>	Absent	96.15	0.9901
<i>CCT-14/3X1</i>	Present	96.44	0.9903
<i>CCT-14/7X2</i>	Absent	95.39	0.9887
<i>CCT-14/7X2</i>	Present	95.60	0.989

The results on the Malaria dataset are mentioned in Table 2 (the best scores are highlighted in bold). The results on the BloodMNIST dataset are mentioned in Table 3, i.e. Accuracy and AUC (the best scores are highlighted in bold). Other scores achieved by our model on BloodMNIST are as follows: Precision (0.9282), Recall (0.9275), F1-Score (0.9268) and MCC (0.9156). In both these tables, our proposed architecture has been compared with existing architectures as well. Training our proposed architecture from scratch on the datasets was better than using transfer learning. Also, our model performs better than most existing architectures on almost all the metrics, utilising fewer parameters and being less complex. Figure 2 shows the AUC/ROC curve obtained for the Malaria dataset. The t-SNE visualisations of the Malaria and BloodMNIST dataset produced by the proposed architecture are shown in Fig. 3. The t-SNE visualisations show that the model can map images from different classes to different points in the feature space, which shows that our proposed model can differentiate and discriminate well between the classes within both the datasets.

We now answer the three questions that were mentioned. We achieve competitive scores on small medical imaging datasets even when we train our proposed model from scratch without any pre-training; thus, the answer to the first question is a yes. Our proposed model is independent of huge amounts of data (Malaria: 27,558 images, BloodMNIST: 17,092 images), is not data hungry and still performs well. Our model is compact (it has only about 6.3M parameters) and still gives competitive results. However, to improve the metrics even further, the model can be made more complex;

Table 2 Comparison of TwT with existing architectures on the Malaria dataset

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC	MCC
[16]— <i>kEffNet-B0 V2 2ch</i>	96.70	–	–	–	–	–
[17]— <i>Sequential CNN model</i>	96	–	–	–	–	–
[18]— <i>DenseNet-201 (Best accuracy)</i>	93.39	0.9549	0.9104	0.9321	–	–
[18]— <i>DenseNet-169 (Best precision)</i>	92.99	0.9597	0.8971	0.9273	–	–
[18]— <i>DenseNet-121 (Best recall)</i>	92.62	0.9072	0.949	0.9276	–	–
[3]— <i>Proposed model (Patient level)</i>	95.90	–	0.947	0.959	0.991	0.917
[19]— <i>Custom</i>	96.29	0.9804	0.9234	0.9495	0.9116	0.9051
[19]— <i>CNNExSVM</i>	94.77	0.9213	0.9515	0.9501	0.9101	0.8925
<i>TwT—w/pre-trained weights</i>	96.51	–	–	–	0.9912	–
<i>TwT—w/o pre-trained weights</i>	96.84	0.9649	0.9713	0.9681	0.9941	0.9369

Table 3 Comparison of TwT with existing architectures on the BloodMNIST dataset

Model	Accuracy (%)	AUC
[6, 20]— <i>Vision Transformer</i>	88.80	0.985
[20, 21]— <i>OrthoFNN</i>	82	0.972
[20]— <i>OrthoPatchWise</i>	86.60	0.984
[20]— <i>OrthoTransformer</i>	86	0.982
[20]— <i>CompoundTransformer</i>	87	0.985
[4, 22]— <i>auto-sklearn</i>	87.80	0.984
[12]— <i>MonoNet</i>	88.10	–
<i>TwT</i> —w/ pre-trained weights	90.12	0.9901
<i>TwT</i> —w/o pre-trained weights	92.75	0.9933

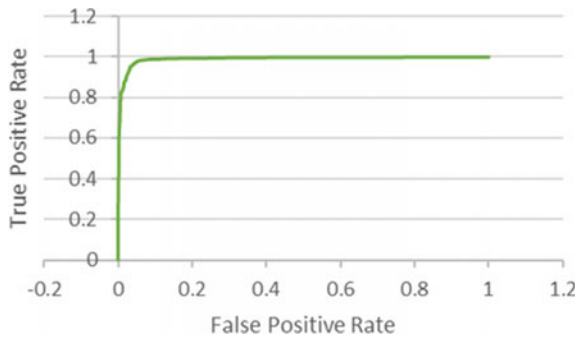


Fig. 2 AUC/ROC curve of TwT on the Malaria dataset

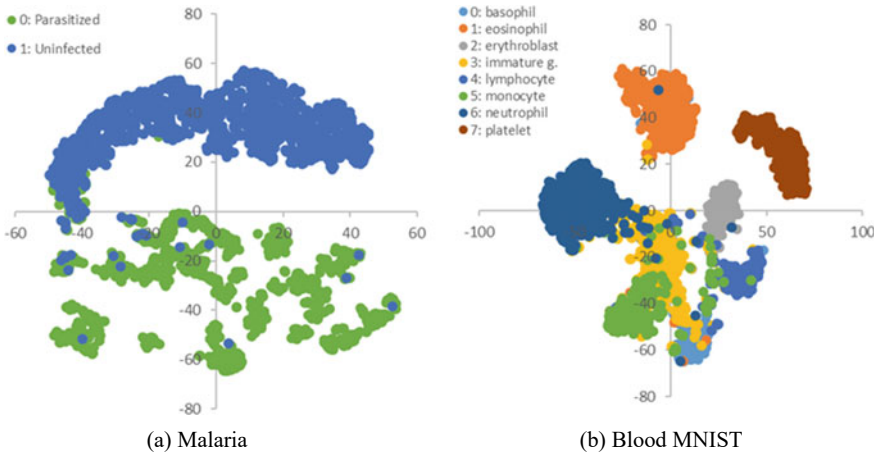


Fig. 3 t-SNE visualisation of TwT on the given datasets

increasing the number of parameters and in turn the time needed by the model to train, thus defeating the purpose of being compact but improving results. Thus as explained, the CDCN++ branch extracts texture-features using the CDC operator, while the CCT block captures local and global spatial features. As Convolutions and MSAs are complementary (CCT block), they help the model to extract the best features of an image. Moreover, the texture-features from the first branch are used to weigh and encode the texture-rich information into the Transformer-Encoder block outputs resulting in better classification results. Further, as both branches are not too computationally complex, they have in total about 6.3M parameters that account for the quick and efficient training and inference.

6 Conclusion and Future Work

In conclusion, our proposed “TwT” architecture has shown competitive results on disease detection and medical image classification on the Malaria dataset and Blood-MNIST dataset without utilising a lot of parameters. In our research, we have shown the application of texture-features, CNNs and CCT to achieve great results and as a highly efficient and accurate solution to the task of disease detection from cell images and solved the three questions listed above. By incorporating CDC and the CDCN++ - backbone to calculate the texture-features; and using them to automatically weigh the CCT Transformer-Encoder block outputs, we have achieved competitive performance without utilising a lot of parameters. We proved that we can achieve competitive scores on small medical imaging datasets even when our proposed model is trained from scratch without pre-training, showing that transfer learning is not always necessary. Moreover, the transfer learning approach on our model saw a slight degradation in performance. Our model is not dependent on huge datasets and, at the same time, has no compromise in performance due to the availability of less data. Our model is compact (it has only about 6.3M parameters) and still gives competitive results. This demonstrates the potential of the current proposed model to improve the accuracy and speed of automated disease detection and to be used for other datasets and to be applied to other disease detection problems as well with further modifications and additions to the existing architecture.

References

1. Hassani A, Walton S, Shah N, Abuduweili A, Li J, Shi H (2021) Escaping the big data paradigm with compact transformers. arXiv preprint [arXiv:2104.05704](https://arxiv.org/abs/2104.05704)
2. Yu Z, Zhao C, Wang Z, Qin Y, Su Z, Li X, Zhou F, Zhao G (2020) Searching central difference convolutional networks for face anti-spoofing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5295–5305

3. Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR (2018) Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ* 6:e4568
4. Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, Pfister H, Ni B (2023) MedMNIST v2—a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci Data* 10(1):41
5. Acevedo A, Merino A, Alférez S, Molina Á, Boldú L, Rodellar J (2020) A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data Brief* 30
6. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J (2020) An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929)
7. Yacin M, Alrasheadi BA, Prakash NB, Hemalakshmi GR, Mohanarathinam A, Shankar K (2021) Deep learning based an automated skin lesion segmentation and intelligent classification model. *J Ambient Intell Humaniz Comput* 12:3245–3255
8. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, Cai M, Yang J, Li Y, Meng X, Xu B (2021) A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *Eur Radiol* 31:6096–6104
9. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med* 121:103792
10. Ryu H, Shin SY, Lee JY, Lee KM, Kang HJ, Yi J (2021) Joint segmentation and classification of hepatic lesions in ultrasound images using deep learning. *Eur Radiol* 31:8733–8742
11. Daanouni O, Cherradi B, Tmiri A (2021) Self-attention mechanism for diabetic retinopathy detection. In: *Emerging trends in ICT for sustainable development: the proceedings of NICE2020 international conference*. Springer International Publishing, pp 79–88
12. Nguyen AP, Moreno DL, Le-Bel N, Rodríguez Martínez M (2023) MonoNet: enhancing interpretability in neural networks via Monotonic Features. *Bioinform Adv* 3(1):vbad016
13. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: *Advances in neural information processing systems*, p 30
14. Yang J, Li A, Xiao S, Lu W, Gao X (2021) Mtd-net: learning to detect deepfakes images by multi-scale texture difference. *IEEE Trans Inf Forensics Secur* 16:4234–4245
15. Park N, Kim S (2022) How do vision transformers work? arXiv preprint [arXiv:2202.06709](https://arxiv.org/abs/2202.06709)
16. Schwarz Schuler JP, Also SR, Puig D, Rashwan H, Abdel-Nasser M (2022) An enhanced scheme for reducing the complexity of pointwise convolutions in CNNs for image classification based on interleaved grouped filters without divisibility constraints. *Entropy* 24(9):1264
17. Sinha S, Srivastava U, Dhiman V, Akhilan PS, Mishra S (2021) Performance assessment of Deep Learning procedures on Malaria dataset. *J Robot Control (JRC)* 2(1):12–18
18. Qadir AM, Abdalla PA, Ghareb MI (2022) Malaria parasite identification from red blood cell images using transfer learning models. *Passer J Basic Appl Sci* 4(Special issue):63–79
19. Rahman A, Zunair H, Rahman MS, Yuki JQ, Biswas S, Alam MA, Alam NB, Mahdy MRC (2019) Improving malaria parasite detection from red blood cell using deep convolutional neural networks. arXiv preprint [arXiv:1907.10418](https://arxiv.org/abs/1907.10418)
20. Cherrat EA, Kerenidis I, Mathur N, Landman J, Strahm M, Li YY (2022) Quantum vision transformers. arXiv preprint [arXiv:2209.08167](https://arxiv.org/abs/2209.08167)
21. Kerenidis I, Landman J, Mathur N (2021) Classical and quantum algorithms for orthogonal neural networks. arXiv preprint [arXiv:2106.07198](https://arxiv.org/abs/2106.07198)
22. Feurer M, Klein A, Eggenberger K, Springenberg J, Blum M, Hutter F (2015) Efficient and robust automated machine learning. In: *Advances in neural information processing systems*, p 28

Automatic Keyphrase Extraction Using Fuzzy-Based Evolutionary Game Theory Approach



Minni Jain, Rajni Jindal, and Amita Jain

Abstract With the increased usage of social media, a massive amount of textual data is generated, which is used for applications such as commerce trend analysis, opinion mining, information retrieval, and so on. Automatic key extraction is a critical and important operation in this situation. Many graph-based approaches that employ co-occurrence as edge weight have been developed, but these algorithms ignore the semantic relationships between words. This paper offers an unsupervised automatic key extraction framework that combines evolutionary game theory with a fuzzy method. The suggested methodology treats automatic key extraction as a consistent labelling problem to ensure that candidates are consistently classified as key or non-key. Various datasets are employed for experimentation, and the outcomes suggest that the proposed approach outperforms the state-of-the-art methods.

Keywords Game theory · Keyword extraction · Fuzzy logic · Natural language processing

1 Introduction

The task of discovering essential words and phrases that best represent a particular text material is known as automatic key extraction. Keyword extraction enables the search and indexing of massive digital text collections and is used in a range of natural language processing (NLP) and information retrieval (IR) activities [1, 2]. Key extraction has emerged as a core NLP activity, with advancements in this task potentially leading to enhancements in higher-level applications that rely on it. Due to the importance of key extraction, various ways have been offered in the literature, mostly along two study lines: unsupervised and supervised. However, the field is

M. Jain (✉) · R. Jindal
Computer Science and Engineering, Delhi Technological University, Delhi, India
e-mail: minnijain@dtu.ac.in

A. Jain
Computer Science and Engineering, Netaji Subhas University of Technology, Delhi, India

confronted with two distinct issues. To begin, while state-of-the-art results on key extraction are substantially lower than on many basic NLP tasks, there is still room for improvement in this area. To address the first difficulty, we saw the importance of approaching this difficult topic from a new angle. As a result, we present a fuzzy-based game-theoretic framework for automatic key extraction. Strategies, payoffs, and players are all essential components of any game. In the proposed study, the candidate keys act as players with two strategies: being a key and being a non-key. Before settling on a strategy, each phrase plays a game with all the other phrases. Matrices for each player's strategies are calculated using a fuzzy similarity metric. We first define essential terms, and from there, we derive heuristics that are expressed in mathematical payoffs. Our criteria for identifying a candidate key as a key consist of two conditions: (a) its relevance to the document's primary topics and (b) its correlation with other pertinent phrases in the document. Our decision to explore the use of game theory was influenced by prior work [10]. First, game theory and NLP share an intuitionistic connection in that both feature units networking with each other and manipulating each other's conduct. The capacity of game-theoretic frameworks to execute consistent labelling of entities under contextual constraints was second and more essential. The capacity of fuzzy logic to handle real-life uncertainties present in common language is the impetus for utilising fuzzy with game theory. As far as we know, there have been no previous proposals for utilizing a fuzzy-based game-theoretic approach for automatic keyword extraction. Our research represents the first attempt to do so, and it highlights the relatively nascent state of applying game theory to natural language processing. We anticipate that our work will serve as a catalyst for additional investigations in this field.

2 Related Work

In graph techniques, text is represented as a graph, and term-to-term connections are measured using various graph connectivity procedures like degree centrality, PageRank centrality [3], and many more. In this model, words are represented as nodes, originally introduced as TextRank [4], where an edge connects two subsequent words if they appear in a fixed window size. The PageRank algorithm is used to find the rank of the nodes. The authors in 2008 extended TextRank and proposed a SingleRank after including edge weights between nodes based on co-occurrences [5]. New enhancement, called Position_Rank, highlighted the importance of considering a word's position and frequency in a partial Page_Rank method. This algorithm preferred words that came previously and frequently in text.

Efforts also made to extract key sentences that cover the essential points of a document [6]. This has been achieved through either clustering-based approaches, grouping terms to proper topics, or employing the LDA method to determine topic distribution [7]. However, LDA-based algorithms require training data and are dependent on the corpus used.

In terms of clustering-based techniques, the authors proposed TopicRank, which clusters candidate words based on the shared term fraction using hierarchical clustering [8]. In 2022, Jain et al. presented a methodology for keyword extraction using fuzzy centrality measures and conducted experiments using localized tweets [1]. Saxena et al. introduced a concept called “keygames”, applying game theory with word embeddings for keyphrase extraction [2]. Game theory has been employed in various other natural language processing tasks, such as rumour detection, query expansion, and sentiment analysis [9–12].

3 Evolutionary Game Theory

It was first brought to light by Smith and Price in 1973. It provided a solution to some of the constraints of the conventional game theory like the high degree of rationalism forced on the players. The truth is that real-life players are motivated to select a strategy depending on some heuristics or social norms [11].

Strategy space of a player is described as its set of original strategies having a probability distribution applied over them. It can be denoted as follows: $x_i = (x_{i1}, \dots, x_{ic})$, where c is the total count of the strategies and every entry x_{ij} represents the probability with which player i selects its j th strategy. The strategy space lies on the c -dimensional standard simplex Δ_c , where $\sum_{h=1}^c x_{ih} = 1$ and $x_{ih} \geq 0 \forall h$.

The anticipated payoff of an original strategy e^h in a single game is given as $u(e^h, x) = e^h Ax$, where A is the $c \times c$ payoff matrix. The significant mark of dissimilarity in evolutionary game theory is that a player plays games with each of its neighbours and computes its resultant payoff as a cumulative total of the partial payoffs gained as a result of individual games. The average payoff of the player is calculated as $u(x, x) = \sum_{h \in S} x_h u(e^h, x)$.

After every iteration, a player uses its payoffs achieved during the game to update its strategy space. It can assign greater probability to the strategies with higher payoffs till a state of equilibrium is attained. For convergence to Nash Equilibrium, the replicator dynamic equation is brought into effect [9].

$$x = [u(e^h, x) - u(x, x)] \cdot x_h \forall h \in S. \quad (1)$$

It allows the strategies better than the average to grow. As in [9], we have used the following version of the replicator dynamic equation which marks distinct intervals of time:

$$x_h(t+1) = x_h(t) \frac{u(e^h, x)}{u(x, x)} \forall h \in S. \quad (2)$$

At every time quantum t , every player amends its strategies till the point of the convergence of the system and the attainment of Nash Equilibrium. The dynamics demonstrate a process of evolution, stochastic in nature, in which the participants

adjust their actions according to their surroundings in the contemporary evolutionary game-theoretic model. Nash Equilibrium [12] can be considered to be the most essential concept of game theory. It represents a state wherein the strategy profile of each player is the most appropriate response to the strategy profile of its co-players, and it has no inducement to change its decision because there is no scope for further improvement [10]. In simple words, Nash Equilibrium occurs when each player has made a choice in accordance with the choices of the other players.

4 Proposed Methodology

This section describes the proposed methodology for keyphrase extraction using fuzzy game theory. The overall proposed methodology is divided into five steps discussed in forthcoming subsections in detailed.

4.1 Candidate Key Phrase Extraction

This segment includes extraction of candidate keyphrases from intext text. The input text is passed through a parser of Spacy to get noun phrase chunks from input text. Nouns are divided in tokens. All the noun phrases extracted are considered as candidate keyphrases $CKP = [KP_1, KP_2, KP_3, \dots, KP_i, \dots, KP_n]$.

4.2 Candidate Keyphrase Fuzzy Similarity Matrix and Sense Fuzzy Similarity Matrix

For Keyphrase Fuzzy Similarity Matrix (KPFSSM) input is candidate keyphrase $CKP = [KP_1, KP_2, KP_3, \dots, KP_i, \dots, KP_n]$ extracted in previous step. And output is a 2-D KPFSSM matrix ($\text{number_of_keyphrases} \times \text{number_of_keyphrases}$) marking the similarity between every pair of keyphrases. The next matrix is fuzzy sense similarity matrix (SFSM) where input is all the senses of $CKP_i \in CKP$ and output is a 2D matrix ($\text{number_of_synsets} \times \text{number_of_synsets}$) marking the pairwise fuzzy similarity values of the synsets of the keyphrases of *TEXT*.

The similarity between two *CKPs* is calculated using fuzzy Jaccard similarity measure [13]. It is a modification of the Jaccard similarity coefficient that takes into account partial matching between two sets. The traditional Jaccard similarity coefficient is defined as the ratio of the size of the intersection of two sets to the size of the union of the sets. However, the fuzzy Jaccard similarity measure considers that elements in the intersection of the sets may not match exactly, but may have a certain degree of similarity.

To calculate the fuzzy Jaccard similarity, a threshold value is set for the degree of similarity required for elements in the intersection to be considered as matches. Then, the degree of similarity between each pair of elements in the sets is calculated using a fuzzy matching algorithm, such as the Jaro–Winkler distance or Levenshtein distance. The pairs of elements with similarity greater than the threshold value are considered as matches and included in the intersection of the sets.

Finally, the fuzzy Jaccard similarity is calculated by dividing the size of the fuzzy intersection of the sets by the size of the fuzzy union of the sets. The fuzzy union of the sets includes all elements from both sets, but removes duplicates and includes only the highest degree of similarity for each element.

4.3 Strategy Space Generation

A 2D matrix (number_of_keyphrases × number_of_synsets) marks the strategy profile of each key phrase against all synsets—*strategy_space*, where input is all the candidate keyphrases $CKP = [KP_1, KP_2, KP_3, \dots, KP_i, \dots, KP_n]$. Nash Equilibrium using Dynamic Replicator and final scores is shown in algorithm 1 and algorithm 2, respectively.

$$strategy_space[i][j] = \begin{cases} |count_i|^{-1} & \text{if synset } j \text{ is from the key phrase } i \\ 0 & \text{Otherwise} \end{cases}$$

where $|count_i|$ is the number of synsets of key phrase i

Algorithm 1 Replicator Dynamics for keyphrase extraction

Input: *list*, *keyphrase_matrix*, *fuzzy sense_matrix*, *strategy_space*

Output: *strategy_space* gets updated

1		Let <i>number_of_iterations</i> = 200 // Assumption	
2		for $i = 1$ to <i>number_of_iterations</i> do	
3		for $j = 1$ to n do // Player	
4		for $k = 1$ to n do // Neighbour	
5			Let <i>payoff_matrix</i> = a (fuzzy sense_count_of_player x fuzzy sense_count_of_neighbour) submatrix mined accordingly from the <i>fuzzy sense_matrix</i>
6			The payoff for each strategy (<i>current_payoff</i>) is calculated according to the following equation: fuzzy sentence_matrix[j][k] * dot-product of (<i>payoff_matrix</i> , <i>strategy_space</i> [k])
7			The <i>strategy_payoff</i> is calculated as Σ <i>current_payoff</i>
8			The <i>player_payoff</i> is calculated as Σ dot-product of (<i>current_payoff</i> , <i>strategy_space</i> [j] ¹)

(continued)

(continued)

9		Let <i>update_values</i> for a player $j = \text{strategy_payoff} / \text{player_payoff}$
10		Update <i>strategy_space</i> [j] rendering to the found <i>update_values</i>

Algorithm 2 Scores for final keyphrases in resultInput: *tokens* = [$KP_1, KP_2, KP_3, \dots, KP_i, \dots, KP_n$], *strategy_space*Output: A list – *final_scores* containing the computed scores of each key phrase

	Let <i>keyphrase_synsets</i> = None		
	for each keyphrase in <i>Text</i> do		
		<i>important_senses</i> = Most important synsets from the <i>strategy_space</i> matrix	
		<i>keyphrase_synsets.append</i> (<i>important_senses</i>)	
	Let <i>final_scores</i> = []		
	for each keyphrase KP_i in <i>Text</i> do		
		<i>value</i> = 0	
		for each keyphrase KP_i in <i>Text</i> do	
		for each sense s_i in <i>keyphrase_synsets</i> [i] do	
			for each sense s_j in <i>keyphrase_synsets</i> [j] do
			do
			add fuzzy similarity_ measure (<i>synset_i</i> , <i>synset_j</i>) to <i>value</i>
		<i>value</i> = <i>value</i> / (<i>keyphrase_synsets</i> * <i>keyphrase_synsets</i>)	
		<i>Final_scores.append</i> (<i>value</i>)	
	Return <i>final_scores</i>		

5 Experimental Set-up and Result Discussion

For a comprehensive evaluation of the proposed method, two scientific publication datasets are used. The dataset is comprised of the corresponding author manually labelled keyphrases (gold standard), the research paper titles, and abstracts. Both the datasets comprise research papers from ACM Conference on Data Mining (KDD) and Knowledge Discovery and ACM World Wide Web (WWW).

Table 1 presents statistics for two datasets, including the total number of keyphrases and abstracts, the percentage of keyphrases absent from the abstract, the number of keyphrases located, and the mean number of keyphrases per paper. The table also shows the distribution of keyphrases with one, two, three, or more tokens. Three common features are shared by both datasets, namely, an average of four to five keyphrases per paper, approximately half of the keyphrases not being

present in the abstract, and a minority of keyphrases being 3-g or more. These statistical features provide insights into the challenges of extracting key-value pairs from these datasets.

To evaluate the results of key extraction, recall, precision, F1-score, and Mean Reciprocal Rank (MMR) are commonly used, and practically all prior works have done the same. As a result, to ensure consistency when compared to other state-of-the-art algorithms, we maintained our evaluation metric constant.

$$P(\text{precision}) = k_c/k_e, \tag{3}$$

$$R(\text{recall}) = k_c/k_s, \tag{4}$$

$$\text{F1 - measure} = 2\text{PrecisionRecall}/(\text{Precision} + \text{Recall}), \tag{5}$$

where k_c is the number of successfully extracted key-value pairs, k_e is the total number of extracted key-value pairs, and k_s is the total number of author-labelled standard key-value pairs.

Mean Reciprocal Rank is to determine how each document's initial accurate key is rated. MRR is defined for a document d as

$$\text{MRR} = (1/|\text{TD}|) * \text{summation}_{k \in \text{TD}} 1/\text{rank}_K, \tag{6}$$

where TD is the set of target documents and rank_K denotes the rank of the first right keyword among all extracted keywords. We perform experiments on the best (top) n ($n = 2$ and 8) extracted keyphrases for the assessment scores in our studies. For the purposes of comparison, we used Porter Stemmer in our investigation, which assisted in the reduction of both expected and gold key-values to a base form. Tables 2 and 3 compare the results of our system KPFEGT with those of other cutting-edge systems on WWW and KDD datasets at top n predicted keyphrases, where n spans from 2 to 8. We can deduce from Tables 2 and 3 that KDD's overall findings is better than WWW dataset. KPFEGT outperforms all comparative techniques on both the KDD and WWW datasets, as demonstrated in Tables 2 and 3. For example, on both datasets, at top $n = 8$ predicted keyphrases, KPEGT outperforms all other performance measures.

Table 1 Statistics of the datasets

Dataset	#Abs/ #KPs (All)	MissingKPs (%)	#Abs/ #KPs (Loc.)	AvgKPs	#uni	#bi	#tri	#>trigrams
KDD	365/ 1471	51.12	315/ 719	4.03	363	853	189	66
WWW	425/ 2073	56.39	388/ 904	4.87	680	1036	247	110

Table 2 Comparison of proposed KPFEFGT with other state-of-the-art approaches (with top $n = 2$)

Method	KDD	KDD	KDD	KDD	WWW	WWW	WWW	WWW
	P	R	F1	MRR	P	R	F1	MRR
TF-IDF	0.175	0.087	0.116	0.289	0.183	0.075	0.106	0.275
TextRank [4]	0.145	0.721	0.096	0.221	0.150	0.061	0.087	0.222
SingleTPR	0.182	0.090	0.120	0.287	0.168	0.069	0.097	0.275
Position rank [6]	0.172	0.085	0.114	0.280	0.162	0.066	0.094	0.249
FKE [1]	0.191	0.095	0.127	0.309	0.210	0.086	0.122	0.316
GTKPE [2]	0.228	0.113	0.151	0.363	0.221	0.090	0.128	0.336
KPFEFGT	0.254	0.20	0.224	0.434	0.255	0.135	0.177	0.453

Table 3 Comparison of KPFEFGT with other state of the art approaches (with top $n = 8$)

Method	KDD	KDD	KDD	KDD	WWW	WWW	WWW	WWW
	P	R	F1	MRR	P	R	F1	MRR
TF-IDF	0.091	0.178	0.121	0.336	0.095	0.154	0.118	0.321
TextRank [4]	0.075	0.147	0.100	0.270	0.081	0.132	0.101	0.273
SingleTPR	0.088	0.172	0.117	0.329	0.090	0.145	0.111	0.326
Position rank [6]	0.085	0.166	0.113	0.324	0.089	0.144	0.110	0.303
FKE [1]	0.094	0.182	0.124	0.358	0.098	0.159	0.122	0.361
GTKPE [2]	0.098	0.191	0.130	0.392	0.102	0.165	0.126	0.378
KPFEFGT	0.132	0.212	0.163	0.43	0.167	0.23	0.194	0.421

6 Conclusion

The proposed method introduces a novel approach to automatic keyphrase extraction, which is modelled as a consistent labelling problem and solved using fuzzy-based game theory. The proposed method defined keyphrases and developed a logical framework to represent them mathematically as payoffs in keyphrase games. The work evaluated the proposed method on two widely used scientific publication datasets and found that it outperforms most existing systems, demonstrating the potential of this approach. In future, authors plan to explore the use of other methods in hybrid, which could reduce the time complexity of game theory. Additionally, authors would explore different heuristics for payoffs and may study various document embeddings to get the document theme.

References

1. Jain M, Bhalla G, Jain A, Sharma S (2022) Automatic keyword extraction for localized tweets using fuzzy graph connectivity measures. *Multimedia Tools Appl* 1–26
2. Saxena A, Mangal M, Jain G (2020) Keygames: a game theoretic approach to automatic keyphrase extraction. In: *Proceedings of the 28th international conference on computational linguistics*, pp 2037–2048
3. Wang J, Liu J, Wang C (2007) Keyword extraction based on pagerank. In: *Advances in knowledge discovery and data mining: 11th Pacific-Asia conference, PAKDD 2007, Nanjing, China, 22–25 May 2007. Proceedings*, vol 11. Springer Berlin Heidelberg, pp 857–864
4. Li G, Wang H (2014) Improved automatic keyword extraction based on textrank using domain knowledge. In: *Natural language processing and Chinese computing: third CCF conference, NLPCC 2014, Shenzhen, China, 5–9 Dec 2014. Proceedings*, vol 3. Springer Berlin Heidelberg, pp 403–413
5. Wan X, Xiao J (2008) Single document keyphrase extraction using neighborhood knowledge. *AAAI* 8:855–860
6. Florescu C, Caragea C (2017) PositionRank: an unsupervised approach to keyphrase extraction from scholarly documents. In: *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, Vancouver, Canada. Association for Computational Linguistics
7. Sterckx L, Demeester T, Deleu J, Develder C (2015) Topical word importance for fast keyphrase extraction. In: *Proceedings of the 24th international conference on world wide web. Association for Computing Machinery*, pp 121–122
8. Bougouin A, Boudin F, Daille B (2013) Topicrank: graph-based topic ranking for keyphrase extraction. In: *Proceedings of the sixth international joint conference on natural language processing. Asian Federation of Natural Language Processing*, pp 543–551
9. Jain M, Suvarna A, Jain A (2022) An evolutionary game theory based approach for query expansion. *Multimedia Tools Appl* 1–25
10. Jain M, Jaswani A, Mehra A, Mudgal L (2022) EDGly: detection of influential nodes using game theory. *Multimedia Tools Appl* 1–23
11. Jain M, Gayathri B, Ranjan R (2022) SentiGames—a game theoretic approach to sentiment analysis. In: *2022 7th international conference on communication and electronics systems (ICCES)*. IEEE, pp 963–968
12. Jain M, Jaswani A, Mehra A, Mudgal L (2020) Rumour source detection using game theory
13. Wu D, Mendel JM (2018) Similarity measures for closed general type-2 fuzzy sets: overview, comparisons, and a geometric approach. *IEEE Trans Fuzzy Syst* 27(3):515–526

Effective Machine Learning-Based Heart Disease Prediction Model



Sandeep Kumar Saini  and Garima Chandel 

Abstract One of the most serious health issues affecting people today is heart disease. Identifying cardiac disease can be challenging as several common risk elements including high cholesterol, diabetes, irregular heart rate, high blood pressure, and various medical conditions can make diagnosis difficult. Due to these limitations, researchers are increasingly adopting cutting-edge techniques like machine learning and data mining to forecast disease. In this study, we assess just the following symptoms: age, sex, chest pain type range of 1–5, serum cholesterol, maximum heart rate attained, overnight sugar levels range of 0 or 1, and resting electrocardiogram range of 0–2, ST depression brought on by activity compared to rest, ST section for the peak reps, exercise-induced angina, by using fluoroscopy and Thal, the main vessels' number (0–3) were colored. The Cleveland cardiovascular database from the UCI repository is one of the datasets which is used in the present study, then applying a machine learning approach and classifying whether it is affected or not. We are using Ridge Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Naive Bayes, and Logistic Regression Model. Finally, comparison of different machine learning-based available methods using the same database with the performance of a proposed method for the detection of heart disease has been done. Linear Discriminant Analysis has given accuracy (acc) and specificity (spec) that is 85.71% and 93.87%, respectively. But in the case of sensitivity (sen) of Ridge Classifier 83.33% which is best as compared with other classifier, overall, the Linear Discriminant Analysis gives better result as compared with other classifiers.

Keywords Machine learning · Fasting blood sugar levels · Resting ECG range · Resting heart rate

S. K. Saini · G. Chandel (✉)
Department of Electronics and Communication Engineering, Chandigarh University, Mohali,
India
e-mail: chandelgarima5@gmail.com

S. K. Saini
e-mail: sandeep.e2677@cumail.in

1 Introduction

One of the most serious health problems that people face is heart disease. Heart disease is also called coronary artery disease. It is the deadliest disease. Heart disease can take many different forms, including angina, congenital heart disease, slow heart rate, cardiac arrest, elevated blood pressure, cardiovascular disease, heart problems, inflammation of the heart, and congenital heart disease [1]. The WHO states about heart attack and stroke, are two main reasons for mortality globally (80%) [2]. Globally, 17.9 million patients died from heart disease in 2016. Heart disease continues to claim millions of lives. About 12,000,000 people die every year which makes the biggest challenge for medical care. According to the Indian Heart Association, about 50% of heart attacks happen in those who were under the age of 50, and 25% happen in people under the age of 40 [2].

Nowadays, the enormous growth in computer technologies is proving itself to be helpful in many fields; it is being helpful in the medical department also. Advances in health care, such as smartwatches and fitness bracelets, are revolutionizing the way we define health every day. Currently, machine learning methods are used to provide clinical support by predicting heart disease based on medical data provided in medical reports, which will be in the form of demographics, symptoms and examination reports, ECGs, and laboratory tests [1, 3–6].

To overcome these issues, machine learning-based models are helpful in the timely detection of heart disease. The invention of smart devices such as continuous glucose monitors and smart cholesterol monitoring systems reduces the incidence of heart disease [7]. These devices work on the principles of machine learning and deep learning algorithms. It uses machine learning-based models for the detection of heart disease by detecting symptoms such as chest pain, high blood pressure, cardiac arrest, shortness of breath, neck pain, sore throat, discomfort, and fainting [8]. There are also risk factors that affect heart disease, like blood pressure, sex, age, cholesterol level, family history of coronary artery disease, diabetes, smoking, alcohol, and being overweight. All the data that come from the following are used in machine learning algorithms to determine the highest accuracy of severe heart disease which will help in the early diagnosis of open-heart disease [9].

The following structure describes the remaining paper layout: Sect. 2 reviews the literature. Section 3 discusses Proposed Heart Disease Detection System. Section 4 explains Results and Discussion, and the summary, comparison, and conclusions are given in Sect. 5.

2 Literature Survey

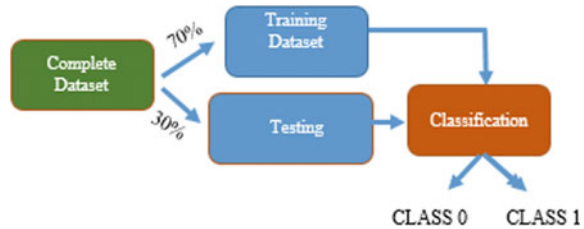
Cardiovascular disease is a major reason for patient death globally. To reduce mortality, we must detect heart diseases early and monitor them continuously [1]. The serious growth of data on health sensor devices generates large data worldwide. Integration of machine learning-based techniques and big data analytics could impact the early detection of heart disease in health care [2]. It could be more efficient and cheaper. First, they used spark ML lib with a spark current head for the detection of cardiovascular diseases, and classification was done for the detection of different class states. After that, the large amount of data generated is stored in Apache Cassandra [3]. The muscular structure in the human body, which is known as the heart, circulates the blood throughout the body by pumping it. People are facing heart disease, and it's a major cause of death worldwide [4]. To help patients react in the early stage of heart disease, data availability and data mining techniques such as machine learning are very useful for early detection of disease. Real-time data processing is not possible with Hadoop Map Reduce, as it supports only data processing. Instead, we use Apache Spark when working with big data. Resilient Distributed Datasets (RDDs) are at the heart of Apache Spark, enabling in-memory storage as well as distributed computing. There are 4 Apache Spark libraries for machine learning, namely MLlib [5]. The algorithms in these libraries are used for running over the distributed dataset that will be suitable for real-time processing.

The suggested work is taken out by integrated random forest model in two phases: in the first phase we are analyzing the healthcare dataset to define the machine learning model; with the help of the above-given dataset, we are wearing max Depth, maxBins, numTrees parameters; after that, different random forest model is being tested [6].

By monitoring using the Spark platform and Cassandra, we are building a scalable heart disease system focused on real-time classification models by continuously monitoring patients' health for key signs of heart disease [3]. To create a model for predicting heart disease in its early stages, we used a random forest algorithm with MLlib machine learning library and proposed a heart monitoring system based on Apache Spark. For evaluating prediction models, we must calculate sensitivity, specificity, and accuracy [7]. The integrated big data technology approaches will be more efficient. Based on TCP message of heart disease attributes, system performance was performed [8]. For developing distributed real-time healthcare analytics systems, big data technologies are more efficient as well as simple compared to traditional methods and can early predict heart disease [9] (Fig. 1).

There are several machine learning algorithms used in predicting cardiovascular diseases (CVD), such as Naive Bayes, Support Vector Machine, Random Forest, K-Nearest Neighbor, and Decision Tree [1, 10–12]. To evaluate their performance, they conducted four human CVD predictions using risk factor data from medical files and one using the UCI dataset. Wahyu et al. results indicate that Naive Bayes has the highest accuracy among the tested methods, with 82.17% accuracy using cross-validation and 84.28% accuracy using split-test methods [13]. However, the accuracy of all algorithms decreased after implementing cross-validation techniques. It is

Fig. 1 Framework of proposed heart disease detection method



important to note that as per a recent report by World Health Organization (WHO), heart diseases are the main cause of death around the globe, with approximately 17.9 million deaths attributed to CVD in 2016. The risk of heart disease can be reduced through a healthy lifestyle, and avoiding risk factors and understanding the causes of these factors can also help prevent and prognosis of heart disease. With the traditional method of diagnosis being costly and time-consuming, researchers develop automatic systems based on medical data from past treatments. Based on medical data, CVD could be accurately predicted with the help of machine learning technique and help in diagnosis of patients [14].

Due to availability, we use the HUCI heart disease dataset. There are 4 subsets and 76 attributes in this dataset, but most studies are based on only 14 attributes including age, sex, chest pain, blood pressure, etc. [15]. These properties are associated with risk factors for diagnosing cardiovascular disease. Here, we are using machine learning techniques to build various prediction models. We have found that the Extra tree algorithm got an accuracy of 79.12%. This model runs between cross-validation and splitting the training test data. Finally, LDA got better accuracy in comparison with other algorithms using both validation methods mentioned above. When we applied cross-validation, the accuracy decreased. Since this method is best suited for simulated scenes, the dataset used here is small and does not take much time to process, and overfitting problems occur over the same period. Our model, based on machine learning algorithms, reduces predictive risk factors for cardiovascular disease, leading to early diagnosis.

The Internet of Things, artificial intelligence, and the use of clouds are emerging information communication and technology sectors [16]. These strategies have the potential to save millions of lives in a medically assisted society and can be used in healthcare systems where health knowledge is scarce. Jameel Ahamed et al. [17] completed the Different Proposed Heart Disease Prediction Models on the basics of AUROC, F-1 Score, accuracy, recall, precision, and error rate they give their results. They concluded that with no parameter adjustment, Nave Bayes performed better, yielding 82.63% accuracy, and that with tuning the hyperparameter, the Random Forest search produced 87.72% accuracy, outperforming all other suggested classifiers. As a result, the Random Forest model is the best for implementation and model construction in their study.

3 Proposed Heart Disease Detection System

The heart is an organ made of cardiac muscles (Myocardium). The role of the human heart is to pump blood all through the entire body. It constitutes a vital element of the body [24]. Machine learning plays an imperative part in diagnosing a heart malady. A few of the machine learning methods are choice trees, neural systems, Naïve Bayes classification, hereditary calculations, and relapse and bolster vector machines [5]. The choice tree calculation is utilized for extricating rules in anticipating heart maladies. It was found to be way better than other machine learning calculations. A graphical client-based interface was utilized to input the quiet information and foresee whether the quiet is enduring from heart malady or not, utilizing Weighted Affiliation run the show-based Classification Restorative qualities such as blood weight, age, and sex were utilized for expectation of heart illness.

The human heart is comprised of four chambers—two atria and two ventricles—with each located on the right and left side of the heart. The atria serve as the heart’s receiving chambers, responsible for collecting blood that returns to the heart from the body and lungs. In contrast, the ventricles act as the heart’s pumping chambers, responsible for pushing the heart’s blood leaving it to flow to the body and lungs. The right and left sides of the heart are separated by a muscular wall known as the septum [22]. This septum ensures that the oxygenated and deoxygenated blood does not mix and enables blood to flow correctly through the circulatory system.

On average, the heart pumps about five liters (eight pints) of blood throughout the body with each beat. This process is known as circulation, and it is crucial for delivering oxygen and nutrients to the body’s cells and tissues, as well as removing waste products and carbon dioxide from the body.

Today, big data analysis, especially health analysis, has become an important topic for many studies. Recently, many researchers are using machine learning in the medical field. In machine learning algorithms, we have to first train the model. After that we must test the model, I am given 70% data for training and 30% data for testing. Training data is always greater than testing data for better prediction as shown in the diagram.

Here, we are working on a binary classification model. In the dataset there are different types of fetchers we are using such as age, sex, chest pain type range of 1–5, serum cholesterol, Resting Blood Pressure, fasting blood sugar range of 0 or 1, resting ECG range of 0–2, reached the maximum heart rate, exercise-induced angina, exercise-induced ST depression compared to rest, exercise’s peak ST section, major vascular count (0–3) are colored using fluoroscopy and Thal [15].

Classification is a fundamental task in machine learning that involves training a model to assign input records to predefined categories. There are numerous classification algorithms available, making it an excellent project for a beginner to explore and compare the different techniques. To undertake this task, the first step is to identify a suitable classification-based problem statement and compile a list of potential classification algorithms to evaluate. The next step is to train the classification models and provide a comparison of their results. We are using five types of classifier name

Ridge Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Naive Bayes, and Logistic Regression Model. After that, we compare the classification model and find out the best model.

There are many ways to locate anomalies, and the process for removing them from the Panda's data frame is the same as for the Panda's data frame itself. Because outliers often appear during the data analysis step of real-world projects, a panda's data frame is used in this instance for a more practical strategy. Lists and other series-type items may also be created using the same technique. Different techniques for detecting outliers (Tables 1 and 2).

Algorithms indirectly use statistical approaches to solve complex problems in your data. In statistics, a normal distribution of data is what statisticians want. The normal distribution of data helps statisticians resolve complex patterns in data and extract valuable insights from it. However, in algorithmic scenarios, a normal distribution of data is not always desirable for all dataset types. This means that non-normally distributed data should be preprocessed and cleaned before applying machine learning algorithms. This article describes machine learning feature transformation techniques used to transform data from one format to another while preserving the essence of the data. Simply put, a transformer is a type of function that is applied to non-normally distributed data, making it highly normal when applied.

The input data for this transformation technique can be provided into the transformer, which then normalizes the distribution of the output data for feeding to the next machine learning algorithm. A parameter named output distribution is present here, and its value can be either uniform or normal. Z score method is used for normalization, and yeo-Johnson method is used for Transformation. The Z score is a crucial statistical concept. The standard score is another name for Z score [16]. This

Table 1 Confusion matrix prediction table

	Actual class 1	Actual class 0
Predicted class 0	True negative	False positive
Predicted class 1	False negative	True positive

Table 2 Performance measurement of the purposed method

Parameter	Naive Bayes	Ridge classifier	Linear discriminant analysis	Extra trees classifier	Logistic regression
Specificity (spec) %	85.71	83.67	93.87	79.59	81.63
Sensitivity (sen) %	80.95	83.33	76.19	78.57	71.42
Accuracy (acc) %	83.51	83.51	85.71	79.12	76.92

score helps determine if the data values are above or below the mean and how far they are from the mean. A Z score, as shown in Eq. 1, tells you exactly how many standard deviations a data point deviates from its mean.

$$Z \text{ score} = (x - \text{mean})/\text{std.deviation}. \quad (1)$$

1. The “Z” in the Z score stands for Zeta, the sixth letter in the Greek alphabet, and it derives from Edward Altman’s initial Zeta model, which was created to calculate the likelihood that a publicly traded business will go bankrupt. The Z score, also called zero-mean normalization, helps normalize your data. Normalizing the data into a simpler form using Z score normalization makes it easier for the human mind to understand. Also, it is the data normalization strategy that avoids this outlier problem.
2. This technique normalizes feature values according to the average and standard variation of the data. The key to this technique is to change the information to bring the values of the different features onto a single scale with an average value of 0 and a standard deviation of 1. So, all variables are transformed into one scale.
3. Technically, it counts the standard deviation either higher or lower than the mean. Outliers have no impact on the normalization or standardization of Z scores because the changed characteristics have no established range.

4 Results and Discussion

Confusion matrix is used to evaluate the machine learning model’s measuring parameter when evaluated on a set of test data. It is commonly used to evaluate the effectiveness of classification models, which predict categorical labels for input events. The matrix is displayed as a grid and represents true positives, false positives, true negatives, and false negatives produced during the testing phase of the algorithm. In multi-class classification problems, the matrix size is equal to the number of classes, i.e., an $n \times n$ matrix. For binary classification problems, the matrix size is a 2×2 table. Confusion matrix is used in machine learning for the evaluation of classification performance. It shows the four possible outcomes that can occur when comparing the predicted labels produced by the model with the true labels of a test set: True Negative (TN): The model predicts 0, and the true label is also 0. True Positive (TP): The model predicts 1, and the true label is also 1. False Negative (FN): The model predicts 0, but the true label is 1. This is also called a Type II error. False Positive (FP): The model predicts 1, but the true label is 0. This is also known as a Type I error.

The results are measured in the form of accuracy (acc), sensitivity (sen), and specificity (spec) for measuring the classifier performance [24]. These values were calculated using Eqs. 2–4 as shown in the following formulas:

$$\text{Spec} = \frac{\text{TN}}{\text{TN} + \text{FP}}\%, \quad (2)$$

$$\text{Sen} = \frac{\text{TP}}{\text{TP} + \text{FN}}\%, \quad (3)$$

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}\%, \quad (4)$$

where in a classification problem, True Positive (TP) refers to the test samples that the model detected positive as positive. False Positive (FP) refers to the test samples in which the model detected the negative as positive. False Negative (FN) refers to the test samples that the model detected positive as or negative. True Negative (TN) refers to the test samples in which the model predicted the negative as negative. Table 1 and Fig. 2 show the accuracy of different models used in the present study. This table clears that LDA-based model is providing satisfactory results and improves the robustness of the model.

This section also includes a comparative analysis of the present study and other methods developed for automatic cancer detection using the same database. Table 3 and Fig. 3 show the comparison of different classification using the same Cleveland cardiovascular database from the University of California.

These are the evaluation metrics for different classification algorithms applied to a dataset. The metrics include specificity, sensitivity, and accuracy as explained initially in this section. The Naive Bayes algorithm gave spec, sen, and acc of 85.71%, 80.95%, and 83.51%, respectively. These parameters for Ridge Classifier-based model are 83.67%, 83.33%, and 83.51%, respectively. The Linear Discriminant Analysis-based method has 93.87%, 76.19%, and 85.71%, respectively. The Extra Trees Classifier model has a 79.59%, 78.57%, and 79.12%, respectively. Lastly, the Logistic Regression algorithm was 81.63%, 71.42%, and 76.92% respectively. Among all methods, the LDA Classifier-based method gave satisfactory overall accuracy and specificity for the detection of heart disease automatically, which leads to detecting heart disease with less complex methods. These models are used for study purposes this is not used for patient reporting because presently we use small data of 300 samples of patients so the accuracy is not up to mark but if we increase the number of samples for testing and training then accuracy can be improved. This system can be used for first-stage detection.

5 Conclusion

The present study focused on the computer-aided detection of heart disease using different machine learning-based models. The technique has been demonstrated using five classification models; these were Ridge Classifier, Linear Discriminant Analysis, Extra Trees Classifier, Naive Bayes, and Logistic Regression Model. These

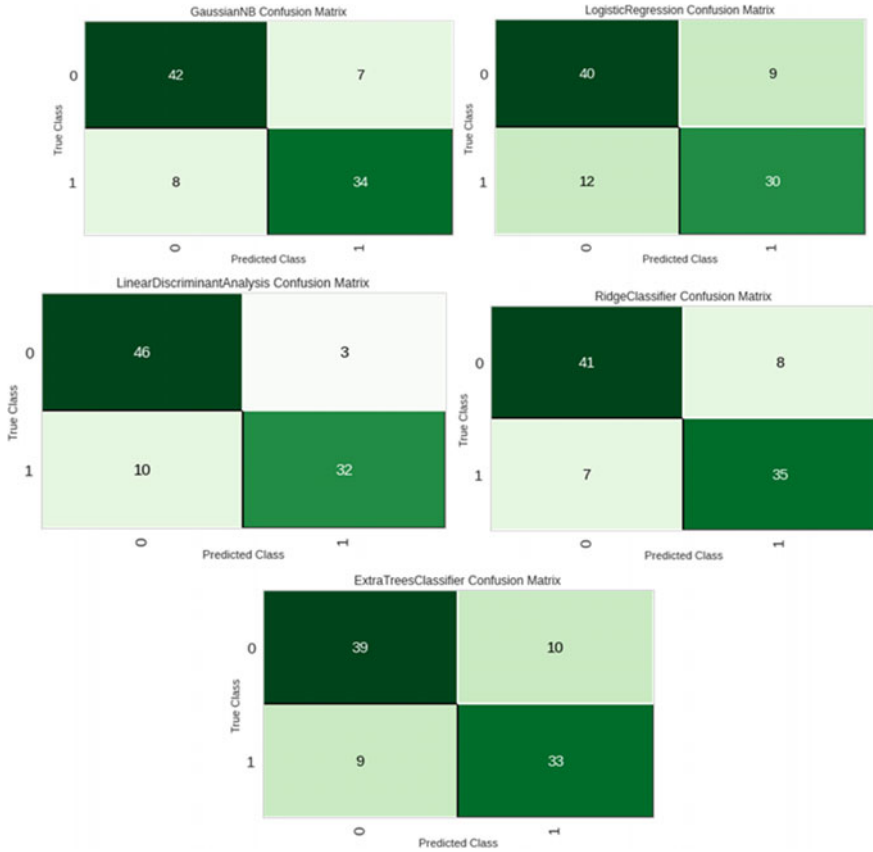


Fig. 2 Confusion matrix for the present study. These metrics can help to identify the strengths and weaknesses of the model and guide further optimization. Example: 686 samples classified as True class 0 out of 690 for ET model

models have been tested using a benchmark dataset. Lastly, the comparison of different machine algorithms are used for detecting the disease. Automatic diagnosis of cardiovascular disease would enable cardiologists to timely give treatment to patients so that it cannot harm the patients. For this purpose, algorithms with good efficiency and accuracy, using machine learning-based methods, enable practitioners to treat this disease earlier with preventative measures. However, it is advisable that this computer-based method can only be used for heart disease prediction, and it is not a diagnostic tool.

Table 3 Comparison table of heart disease detection methods by using Cleveland cardiovascular disease dataset

Papers, Year	Analysis method	Accuracy (%)
Amma et al. [18], 2012	Genetic algorithm and neural network	94.17
Akella et al. [19], 2021	Neural network-based algorithm	93
Kavitha et al. [20], 2021	Hybrid model of Decision Tree and Random Forest	88.7
Srinivas [21], 2022	Hyperparameter optimization technique and XGBoost classifier	94.7
Khennou et al. [22], 2019	Support Vector Machines-based model	87
Setiawanet et al. [23], 2020	Fuzzy decision support system	83
Bashir et al. [24], 2021	Ensemble-based voting scheme-based method	83
Ahamed et al. [17], 2021	CDPS-IoT: Cardiovascular Disease Prediction System based on IoT using machine learning	87.27
This work	Linear Discriminant Analysis-based model	85.71 and Spec 93.87

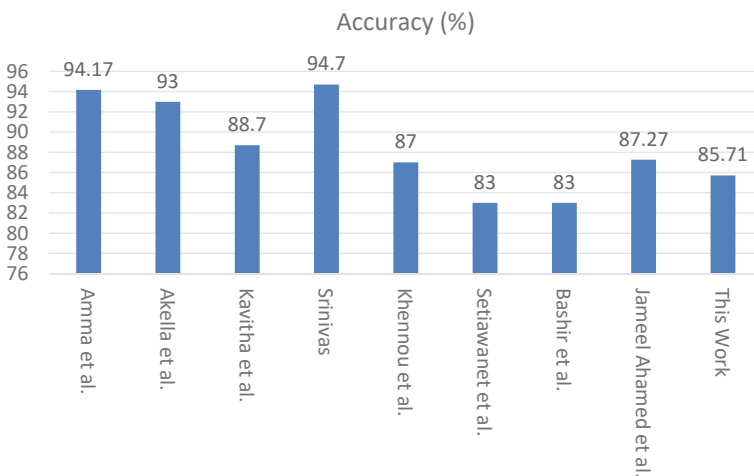


Fig. 3 Comparison graph of heart disease detection methods by using Cleveland cardiovascular disease dataset

References

1. Sarah S, Gourisaria MK, Khare S, Das H (2022) Heart disease prediction using core machine learning techniques—a comparative study. *Lecture Notes Netw Syst* 318:247–260. https://doi.org/10.1007/978-981-16-5689-7_22/COVER

2. Chang V, Bhavani VR, Xu AQ, Hossain MA (2022) An artificial intelligence model for heart disease detection using machine learning algorithms. *Healthcare Anal* 2:100016. <https://doi.org/10.1016/J.HEALTH.2022.100016>
3. Debauche O, Nkamla Penka JB, Mahmoudi S et al (2022) RAMi: a new real-time internet of medical things architecture for elderly patient monitoring. *Information* 13:423. <https://doi.org/10.3390/INFO13090423>
4. Cardiovascular diseases (CVDs). [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)). Accessed 22 Apr 2023
5. Santos-Pereira J, Gruenwald L, Bernardino J (2022) Top data mining tools for the healthcare industry. *J King Saud Univ Comput Inform Sci* 34:4968–4982. <https://doi.org/10.1016/J.JKSUCI.2021.06.002>
6. El-Shafiey MG, Hagag A, El-Dahshan ESA, Ismail MA (2022) A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest. *Multimed Tools Appl* 81:18155–18179. <https://doi.org/10.1007/S11042-022-12425-X/TABLES/17>
7. Kim JO, Jeong YS, Kim JH et al (2021) Machine learning-based cardiovascular disease prediction model: a cohort study on the Korean national health insurance service health screening database. *Diagnostics* 11:943. <https://doi.org/10.3390/DIAGNOSTICS11060943/S1>
8. Harjai S, Khatri SK (2019) An intelligent clinical decision support system based on artificial neural network for early diagnosis of cardiovascular diseases in rural areas. In: *Proceedings of 2019 amity international conference on artificial intelligence, AICAI 2019*, pp 729–736. <https://doi.org/10.1109/AICAI.2019.8701237>
9. Yazdani A, Varathan KD, Chiam YK et al (2021) A novel approach for heart disease prediction using strength scores with significant predictors. *BMC Med Inform Decis Mak* 21:1–16. <https://doi.org/10.1186/S12911-021-01527-5/TABLES/14>
10. Katarya R, Meena SK (2021) Machine learning techniques for heart disease prediction: a comparative study and analysis. *Health Technol (Berl)* 11:87–97. <https://doi.org/10.1007/S12553-020-00505-7/METRICS>
11. Nishat MM, Faisal F, Hasan Udoy M (2021) Performance evaluation and comparative analysis of different machine learning algorithms in predicting cardiovascular disease
12. Wadhawan S, Maini R (2022) ETCD: an effective machine learning based technique for cardiac disease prediction with optimal feature subset selection. *Knowl Based Syst* 255:109709. <https://doi.org/10.1016/J.KNOSYS.2022.109709>
13. Wahyu EJ, Chairani C, Chairani C (2022) The application of particle swarm optimization using Naive Bayes method for predicting heart disease. In: *Proceeding of international conference on information technology and business*, pp 64–71
14. Taran S, Bajaj V (2018) Rhythm-based identification of alcohol EEG signals. *IET Sci Meas Technol* 12:343–349. <https://doi.org/10.1049/iet-smt.2017.0232>
15. UCI machine learning repository: heart disease data set. <https://archive.ics.uci.edu/ml/datasets/heart+disease>. Accessed 22 Apr 2023
16. Ahamed J, Mir RN, Chishti MA (2022) Industry 4.0 oriented predictive analytics of cardiovascular diseases using machine learning, hyperparameter tuning and ensemble techniques. *Ind Robot* 49:544–554. <https://doi.org/10.1108/IR-10-2021-0240/FULL/XML>
17. Ahamed J, Koli AM, Ahmad K et al CDPS-IoT: cardiovascular disease prediction system based on IoT using machine learning. *Int J Interactive Multimedia Artif Intell* 7:4. <https://doi.org/10.9781/ijimai.2021.09.002>
18. Amma NGB (2012) Cardiovascular disease prediction system using genetic algorithm and neural network. In: *2012 international conference on computing, communication and applications, ICCCA 2012*. <https://doi.org/10.1109/ICCCA.2012.6179185>
19. Akella A, Akella S (2021) Machine learning algorithms for predicting coronary artery disease: efforts toward an open source solution. *Future Sci OA* 7. <https://doi.org/10.2144/FSOA-2020-0206/ASSET/IMAGES/LARGE/FIGURE4.JPEG>
20. Kavitha M, Gnaneswar G, Dinesh R et al (2021) Heart disease prediction using hybrid machine learning model. In: *Proceedings of the 6th international conference on inventive computation technologies, ICICT 2021*, pp 1329–1333. <https://doi.org/10.1109/ICICT50816.2021.9358597>

21. Srinivas P, Katarya R (2022) hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. *Biomed Signal Process Control* 73:103456. <https://doi.org/10.1016/J.BSPC.2021.103456>
22. Khennou F, Fahim C, Chaoui H, Chaoui NEH (2019) A machine learning approach: using predictive analytics to identify and analyze high risks patients with heart disease. *Int J Mach Learn Comput* 9:762–767. <https://doi.org/10.18178/ijmlc.2019.9.6.870>
23. Setiawan NA, Venkatachalam PA, Fadzil A, Hani M (2009) Diagnosis of coronary artery disease using artificial intelligence based decision support system, pp 11–13
24. Bashir S, Almazroi AA, Ashfaq S et al (2021) A knowledge-based clinical decision support system utilizing an intelligent ensemble voting scheme for improved cardiovascular disease prediction. *IEEE Access* 9:130805–130822. <https://doi.org/10.1109/ACCESS.2021.3110604>

Internet of Things (IoT) Based Smart Agriculture and Automatic Irrigation Monitoring System Using LoRa



Kalathiripi Rambabu, Sanjay Dubey, Keshavagari Srujana, Gunnala Rajesh, and Mohammed Imran

Abstract Use of the Internet is essential in daily life. Internet data transfer makes it simple to share information in urban areas, where it is easy to keep track of every last detail of the task being done. While the town is expanding, the villages are not being properly developed. Most of the villagers' income comes from farming or other forms of agriculture. It is possible to monitor agricultural land if it is close to the farmer's home, but it is impossible if the farmer's home is a short distance or more away from the field. To address this issue of monitoring over greater distances without using the Internet for communication, LoRa-based technology was introduced. The purpose of this proposal is to convey information about agricultural soil humidity, including its temperature content and weather conditions. This information is transmitted via a LoRa receiver, and with a mobile device, we can monitor the field's data (Internet of Things). Although the land may regularly dry out in the summer, the plant receives water automatically from a water source. While traveling farther apart and without access to the Internet, this technology can be useful. This technology can take the place of the widespread use of Internet-based information exchange.

Keywords LoRa · NodeMCU · Arduino · ThingSpeak · Monitoring

1 Introduction

It is crucial to use the Internet in daily life. In metropolitan locations, where it is straightforward to keep track of every minute aspect of the task being done, Internet data transfer makes it simple to communicate information. The villages are not being properly developed as the town is growing. Farming and other agricultural activities provide the majority of the villagers' income. Modern agriculture emphasizes the

K. Rambabu (✉) · S. Dubey · K. Srujana · G. Rajesh · M. Imran
Department of Electronics and Communication Engineering, B V Raju Institute of Technology,
Narsapur, Medak, Telangana, India
e-mail: rambabukala@gmail.com

S. Dubey
e-mail: sanjay.dubey@bvrit.ac.in

importance of being able to remotely manage and monitor one's agricultural operations. In addition to decreasing waste and costs, this can help farmers become more effective and productive. Farmers may still not watch their crops from their houses in many rural locations, though. The effectiveness and profitability of their business activities may be significantly impacted by this. It is possible to monitor agricultural land if it is close to the farmer's home, but it is impossible if the farmer's home is a short distance or more away from the field. Lack of infrastructure is one of the main reasons why farmers are unable to monitor their fields from home, but it is possible to monitor agricultural land if it is close to the farmer's home. However, it is impossible if the farmer's home is away from the field. It might be difficult for farmers to use technology to manage their companies because many rural areas lack reliable cell phone coverage and fast Internet. This can be a major obstacle because the majority of equipment and systems used for remote field monitoring require a consistent and rapid Internet connection. For those who are not tech-savvy and may not know how to fix issues themselves, this can be particularly difficult. In addition to having an adverse effect on the environment, not being able to monitor fields from home can also have a negative influence on a farm's bottom line. Farmers may not be able to decide on irrigation and fertilization without real-time data and analysis. This may result in lower yields, lower earnings, and more waste. Finally, the general well-being of farmers may be impacted by a lack of remote field surveillance. The ability to monitor fields from home can assist minimize the time and effort needed for farming, which is a physically demanding profession. Farmers may be able to avoid long workdays and physical stress as a result, improving their health and overall quality of life. So the main contribution of this research paper is a product that uses the LoRa-based technology in field communication that made it easy for the farmers to monitor. This paper describes the collection of data through the sensors and shares it with the receiver after processing the data. This data is received and can be viewed in the web dashboard and mobile dashboard. LoRa-based technology was introduced to solve the problem of monitoring over longer distances without using the Internet for data sharing. This strategy emphasizes information exchange over longer distances away from the Internet. Hence, the usage of advanced technologies for the monitoring of agricultural lands has gained higher interest and is a progressive field of study.

2 Literature Survey

“Remote Sensing for Agriculture: An Overview” by G. P. Patil and A. K. Tripathi [1]—This paper provides an overview of remote sensing technologies and their applications in agriculture, including the use of aerial and satellite imagery, as well as ground-based sensors. The authors discuss the benefits of remote sensing, such as increased efficiency, improved crop management, and reduced waste and costs, “Smart Agriculture: A Review of IoT-Based System for Precision Agriculture” by M. A. Al-Sabbagh and M. S. Obaidat [2]—This paper provides a comprehensive review

of the Internet of Things (IoT) and its applications in precision agriculture. The authors discuss the benefits of IoT-based systems for remote monitoring, including increased accuracy and real-time data collection, as well as the challenges and limitations of these systems, “Wireless Sensor Networks for Agriculture: A Review” by J. Chen, S. Song, and X. Li [3]—This paper provides a review of wireless sensor networks (WSNs) and their applications in agriculture. The authors discuss the benefits of WSNs, such as increased accuracy, real-time data collection, and improved crop management, and the challenges and limitations of these systems, “A Review of Smart Agriculture: The Future of Farming” by A. Al-Fadhli and H. Al-Jabri [4]—This paper reviews smart agriculture and its applications, including the use of IoT, wireless sensor networks, and cloud computing. The authors discuss the benefits of smart agriculture, such as increased efficiency, improved crop management, reduced waste and costs, and the challenges and limitations of these systems, “Remote Monitoring of Soil Moisture and Temperature for Agricultural Applications: A Review” by J. J. Kim and H. S. Kim [5]—This paper provides a review of remote monitoring systems for soil moisture and temperature in agriculture. The authors discuss the benefits of these systems, such as increased accuracy, real-time data collection, and improved crop management, and the challenges and limitations of these systems, [6] “Automated Systems for Smart Agriculture” is an area of research that has gained a lot of attention in recent years. The aim of this research is to develop methods and systems for remotely monitoring environmental conditions in agriculture fields, such as temperature, humidity, soil moisture, and light levels. This can help farmers increase efficiency, productivity, and profitability while reducing waste and costs, [7] “Comparison of LoRa-Based Modulations” by R. V. Senyuva. It describes the comparisons of the LoRa modulations for bit rate performances by ICS-LoRa and SSK-LoRa providing numerical results, [8] “Development of Low Power Transmission Line Clamp Temperature Measurement System Based on LoRa Communication” is an area of research that collect the data and transfer through the server using LoRa-based technology for measuring the temperature of transmission lines.

These studies demonstrate the importance of remote monitoring in agriculture and highlight the benefits, challenges, and limitations of these systems. By understanding the current state of research in this area, future studies can be designed to address the remaining challenges and limitations and to further improve the efficiency and productivity of agriculture through remote monitoring.

3 Proposed Model

3.1 Block Diagram

The proposed work eliminates the disadvantages of the GSM, thus increasing the efficiency of the agricultural monitoring system.

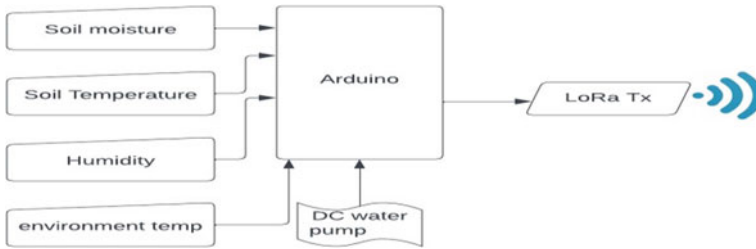


Fig. 1 Block diagram of transmitter

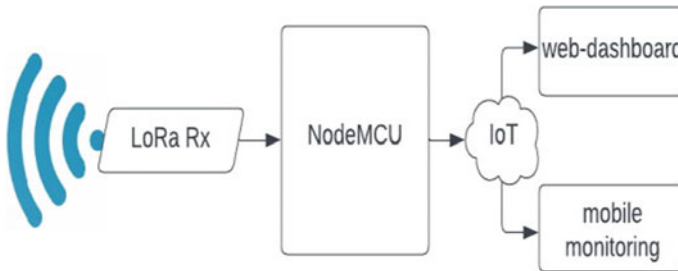


Fig. 2 Block diagram of receiver

The transmitter side of the proposed model concentrates on the integration of the sensors with the Arduino. Further, the DC water pump is connected to ensure the water is pumping automatically. In case of the moisture getting reduced than a specified threshold, the water pump is turned on to maintain the moisture level of the soil. When the values are sensed by the sensor, data gets transmitted to the receiver side using the UART protocol with the help of the LoRa transmitter. The basic idea behind the work is as shown in the block diagrams Figs. 1 and 2.

The receiver gets the data from the LoRa transmitter and hence makes sure that the data is loaded onto the NodeMCU. NodeMCU using its in-built Wi-Fi will help loading the data onto the IOT web dashboard on the ThingSpeak, thus enabling the mobile monitoring.

LoRa ensures the data transmission without the presence of the signaling and does not even require the recharging necessities. The battery consumption by the LoRa is also less in comparison with the other types, specifically GSM. Further, based on the requirements, multiple LoRa types are available so as to be used.

3.2 *Components Used*

The components used in this work and their technical specifications are mentioned below:

Arduino Uno

Arduino Uno is an ATmega328P microcontroller that is a powerful processor for low-end applications. Its general specifications include its frequency, i.e., 16 MHz, an operating voltage of 5 V, and the input voltage is 7–12 V.

Resistive Soil Moisture Sensor

Two probes make up the sensor, which measures the content of water. The soil is passed through the two probes, which subsequently provide the resistance value needed to calculate the soil's moisture.

Content. There will be less resistance when there is more water in the soil so the soil will conduct electricity more readily.

DC Water Pump

This is a budget-friendly, compact Submersible Pump Motor that can be powered by a voltage range of 3–6 V.

It can pump up to 120 L of water per hour while consuming a very minimal current of 220 mA.

DHT11 Sensor

This DHT11 sensor is also called a digital humidity and temperature sensor which is used as measuring the temperature and humidity of the climate or atmosphere in the field it is used in the field for its specifications as it can work 20–80% humidity readings with 5% accuracy, and it works well for 0 to a 50 °C temperature reading, and it can operate in 3–5 V.

DS18B20 Probe

DS18B20 is a waterproof temperature probe used in the agriculture field to collect information on the field temperature or soil temperature and gives it to the Arduino Uno to process and share the information to the transmitter from the board. It can operate up to –55 to 125 °C.

LoRa Module

LoRa module has a variety of advantages, and those advantages are especially suitable for far-field communication; hence, it is specifically utilized for independent far-field and low data transferred communication and uses “chirp spread spectrum modulation” for communication, and it has a frequency range of 433–868 MHz, and it can have the capability to communicate up to 5–10 km without any Internet dependency. This LoRa can be called Long Range (LoRa). This LoRa can consume very low power consumption to the GSM module.

I2C LCD

This I2C LCD (Liquid crystal display) is used in this work because the transmitted data received from the receiver is processed and shown in the I2C LCD display; this can be very helpful when the Internet is too low or lack of Internet connection, IoT will not work at that critical times. This display will help us to monitor lively and can show up to 16 bit data at once also it has 4 pins which are VCC, GND, SDL, and SCL. The maximum current draw is about 200 mA.

NodeMCU

NodeMCU can be termed a (Node Micro-controller Unit). This NodeMCU is responsible for processing the data and decision-making with the pre-programmed board. Specifications that make this NodeMCU special are: Its operating voltage is 3.3 V, and its input voltage can be 4.5–10 V. It has a flash memory of 4 MB and Sram is 64 KB. It has 11 digital I/O pins with 1 analog pin and built-in Wi-Fi of 802.11b/g/n standard with a frequency of 80 MHz.

ThingSpeak Dashboard

ThingSpeak web dashboard is a platform that relates to IoT applications and shows the data in a visualized manner so the data which is shared from the micro-controller is programmed to connect with the IoT platform dashboard by selecting fields in the platform, and it is mentioned in detail in the methodology below. With the advantages of the “ThingSpeak” application, it is very helpful to view anywhere in the world.

4 Implementation

The alternatives for GSM were initially found out. The main problem with the GSM is the signal range and recharging. The best alternative was found to be the LoRa which is independent of the recharging and need not lie in the signaling ranges. This can effectively solve the issues of usage of GSM modules in the agriculture fields. In the present work, four-input sensors and a DC water pump for the transmitter side are used to detect the soil moisture and temperature and also to detect the climate temperature and humidity. These sensors are connected and interfaced with the Arduino Uno for data processing and a threshold to the DC water pump for the irrigation system is given that is to be automatically done when the soil is dry. After the input data sensors are connected, the output data sensor that is LoRa transmitter is interfaced with the Arduino, the data transmission between Arduino and LoRa transmitter doesn't depend on the frequency of the Arduino while interfacing just the frequency needed by the transmitter for receiving purposes.

After interfacing the components, the pin numbers are properly verified. Further, the code for the transmitter side is developed. Finally, the code is dumped onto the board by selecting the Arduino Uno board on the tools in Arduino IDE (Figs. 3, 4, and 5).

Fig. 3 Transmitter schematic diagram

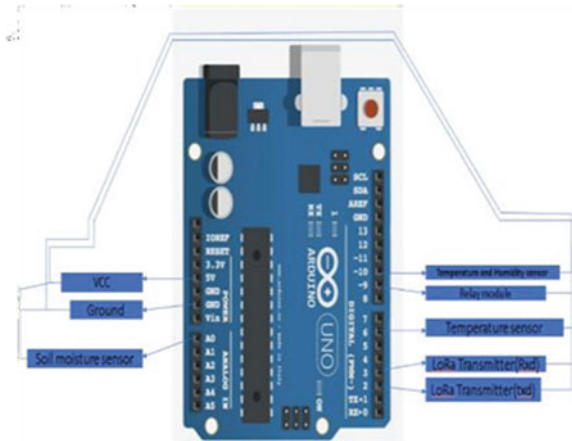
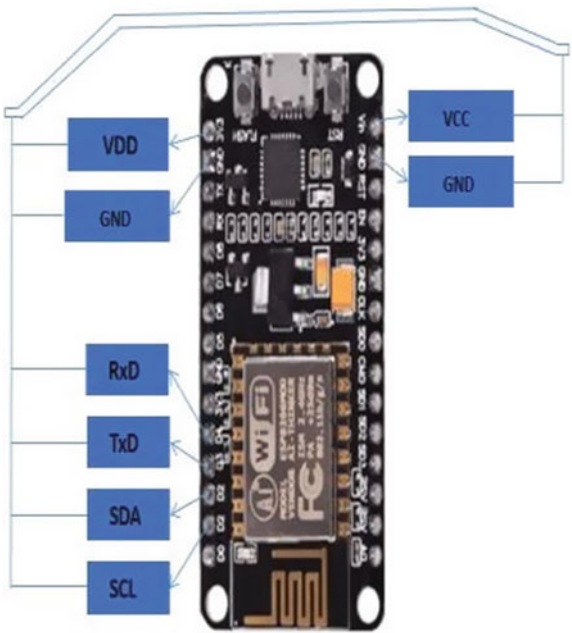


Fig. 4 Receiver schematic diagram



Similarly, for the receiver, the components are connected with the NodeMCU, and the code is developed for collecting the data from the LoRa receiver and processing it with IoT and I2C LCD. All the required libraries required for the code need to be installed and verified properly.

Further, the data from the sensors is collected in the Arduino and according to the threshold values DC water pump is made to work, and the data is transmitted from the LoRa transmitter to the LoRa receiver from the receiver block. The received data

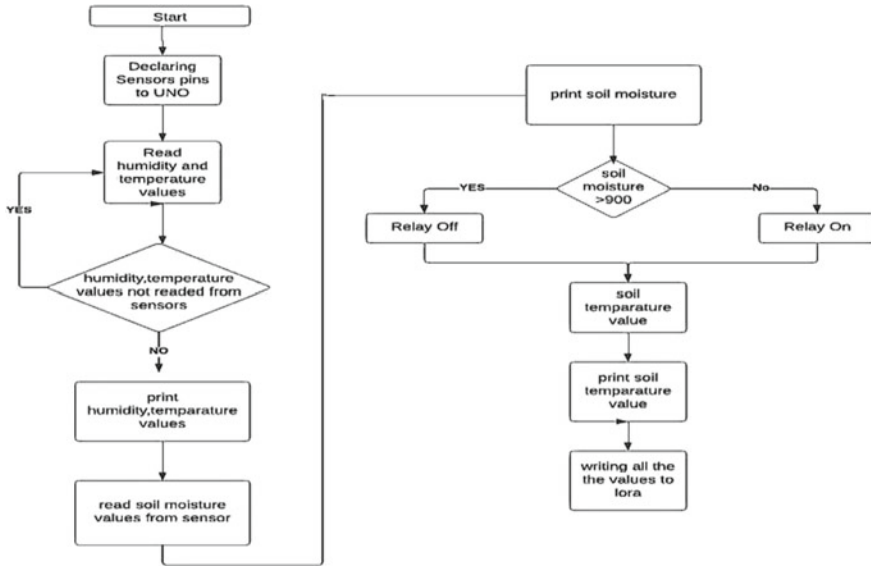


Fig. 5 Flowchart of transmitter code

is collected in the NodeMCU and later connected with the IoT dashboard and mobile monitoring is done with the help of Wi-Fi signals or hotspots.

A ThingSpeak web dashboard account is created. After activating the account, select the number of fields that the sensor data is taking and those are named as per the convenience. After creating the fields, it provides a dashboard with a unique id (Fig. 6).

This unique id is helpful to connect the NodeMCU module with the ThingSpeak so when programming the NodeMCU this unique id is added to the code and the Wi-Fi credentials are provided in the code only so as to connect with the Internet to share the data onto the IoT platform. After coding, compilation is done to check for errors. The code is then dumped onto the board to work. When the receiver is active, it is also important to turn on the transmitter. After successfully dumping, the data from the sensors will be shared with the Arduino board. It processes the data and controls the irrigation system. However, the data is transferred to the LoRa transmitter. Hence, this data is collected in the receiver part and shown in I2C LCD. Since it is also connected to the Internet, the data gets transferred to the IOT web dashboard, and through the mobile, the dashboard can be monitored.

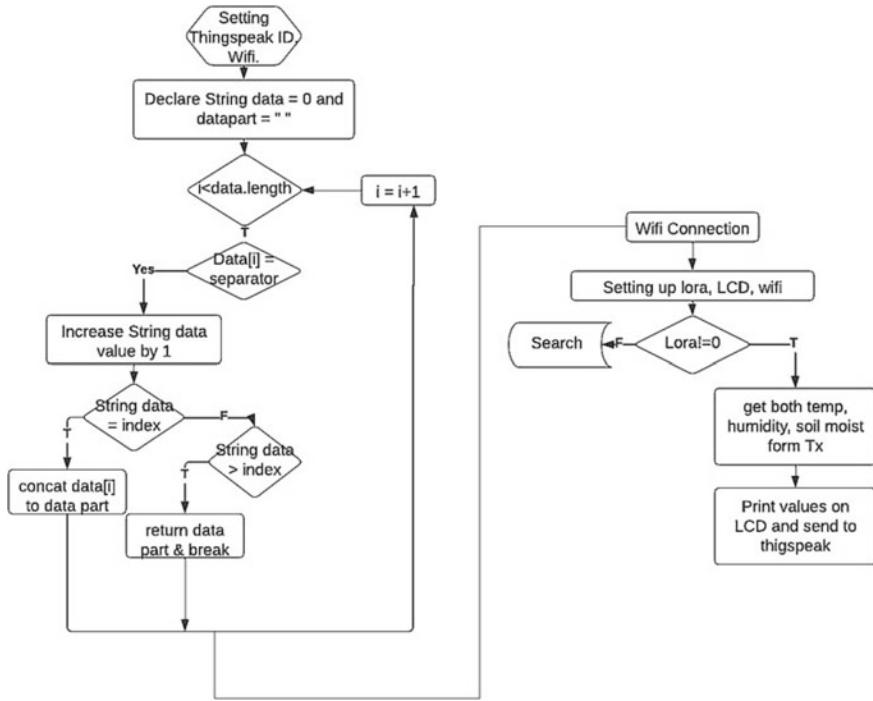


Fig. 6 Flowchart of receiver code

5 Results

After the data is received from the transmitter, it gets processed in the receiver, and the temperature and humidity of the atmosphere and the soil temperature, and moisture content are displayed on the I2C LCD. As it was mentioned, the data gets stored on the IOT web dashboard over ThingSpeak and can be visualized over the mobile monitoring.

Figures 7 and 8 clearly indicate the final prototype of the LoRa-based system for agricultural monitoring.

The prototype here clearly indicates the transmitter side of the work proposed. The humidity and temperature sensor, soil temperature sensor, and soil moisture sensor sense the values required thereafter. After the data is sensed, it is processed within the Arduino and converted into a given LoRa format. Then, it gets shared with the transmitter it only shares the specific type of template format. The transmitter sends the data it travels to the receiver using Universal Asynchronous Receiver and Transmitter (UART) protocol and finally reaches the receiver and is decoded and processed in NodeMCU and shown in the I2C display shown in Fig. 9. The final results as observed in the LCD is as shown in Fig. 9. For every second, various value required to be displayed are observed.

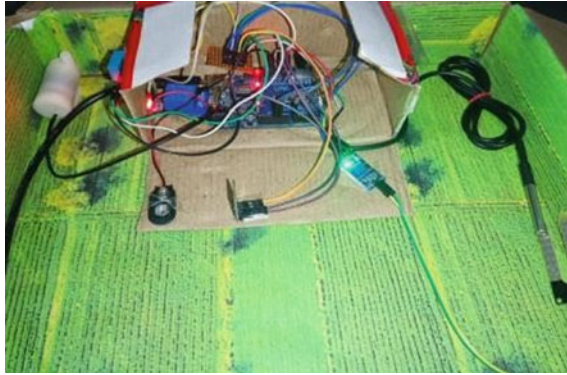


Fig. 7 Working prototype of transmitter



Fig. 8 Working prototype of receiver

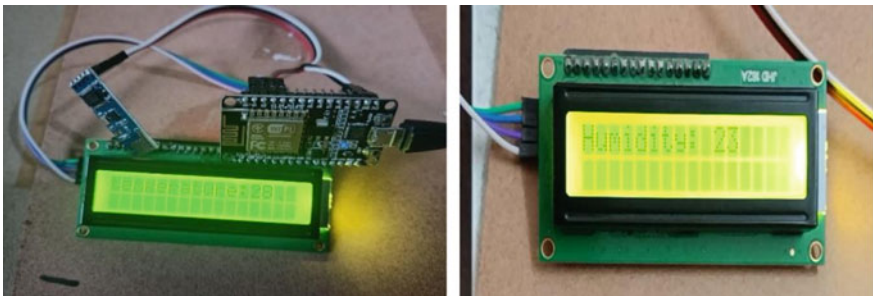
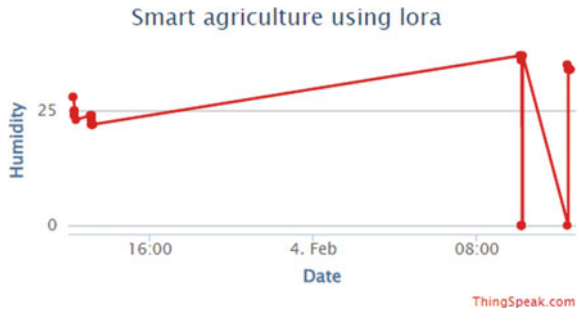


Fig. 9 Results displayed on LCD

Fig. 10 Humidity value in ThingSpeak



As previously highlighted, the web dashboard also shows the various values, i.e., humidity, soil moisture content, temperature, and soil conditioning. All of these are shared with the Internet of Things that are displayed as graphs in the dashboard so as to make the visualization clear to the user (Figs. 10, 11, 12, and 13).

Fig. 11 Soil moisture value

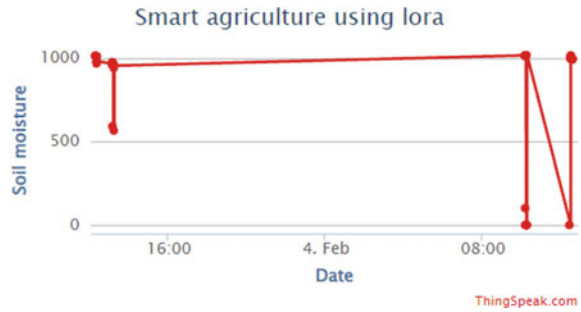


Fig. 12 Soil temperature value

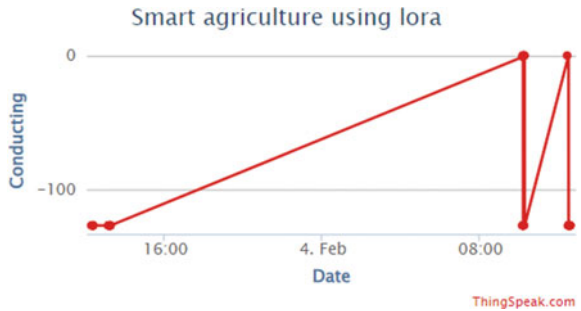
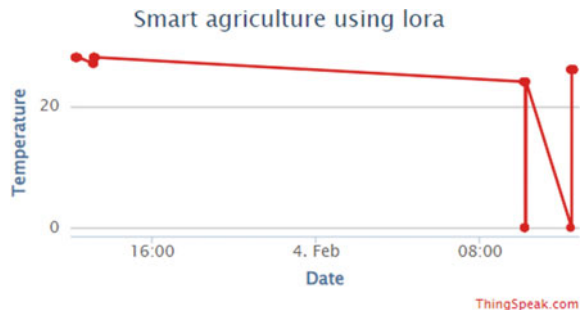


Fig. 13 Temperature value

6 Applications

This is how the proposed model benefits society, for Smart Cities LoRa can be used to monitor and manage various aspects of a city, such as traffic, air quality, and waste management. Asset tracking: LoRa can be used to track the location and status of assets, such as containers, vehicles, and packages, in real time. Agriculture: LoRa can be used to monitor soil moisture levels, crop yields, and other data to optimize agricultural processes and improve crop production. Smart homes: LoRa can be used to control and monitor various devices in a smart home, such as lights, temperature, and security systems. Industrial IoT: LoRa can be used to monitor and control industrial processes, such as manufacturing, supply chain management, and energy management. Environmental monitoring: LoRa can be used to monitor various environmental parameters, such as temperature, humidity, and air quality, to help reduce emissions and improve sustainability. Health care: LoRa can be used to monitor the health of patients and track medical equipment, such as heart monitors, glucose meters, and hospital beds. Retail: LoRa can be used to track inventory and monitor in-store customer behavior to improve the shopping experience and increase sales. So this is how this field of study benefits the society.

7 Limitations

This work has some limitations, due to that it can be used for specific applications. Those limitations include limited data sharing capacity which transfers 3 KB at once. It has less penetration power, so it can be specifically used in agriculture and other rural areas due to fewer buildings and other wireless networks, in urban are—as transmitting range decreases to almost 60%; hence, it cannot be used for wide-range applications in rural areas. It has a limitation of supporting multiple devices, still, it is developing for a multiple-device connection. It uses encryption to protect the data, but it is not updated for faults in encryption. A number of kits are to be installed for accurate data in different places of field.

8 Conclusion

This work entirely concentrated on LoRa-based technology to reduce the drawbacks of the GSM module for the farmers to decrease the cost of the system. This work helps farmers to pay once and use it for a lifetime. This work considers the data from the field and shows the data through the web dashboard of the ThingSpeak platform the data which is collected from the sensors placed in the field. This data is received from the LoRa receiver and shown to the farmer visually in the web dashboard. It helps farmers a lot. Basically, farmers are illiterate so they don't need to change system, once it is set up, automatically it will work for lifetime without any changes like the GSM module that is to be recharged regularly.

9 Future Work

LoRa-based smart agriculture systems have great potential for growth in the future, and there are several ways in which they can be developed further. After studying this research paper, anyone can get a clear idea of how the LoRa-based environment works and its applications need to be developed. More advanced sensors with greater precision and accuracy will improve the data collection and decision-making capabilities of LoRa-based smart agriculture systems. The use of machine learning algorithms and artificial intelligence can help to analyze the large amounts of data generated by smart agriculture systems and make more informed decisions based on that data in the field. Advances in communication technology and data processing will enable real-time monitoring and control of agricultural processes, crop behavior, irrigation, and pest control enabling more efficient and effective management practices. The development of scalable solutions that can be easily deployed and adapted to different agricultural environments will be important in order to make this technology widely accessible to farmers around the world. The future of LoRa-based smart agriculture systems is promising, and continued research and development in this area will likely lead to significant advancements in sustainable agriculture practices.

References

1. Patil GP, Tripathi AK (2017) Remote sensing for agriculture: an overview. *J Appl Rem Sens* 11(4):042609. <https://doi.org/10.1117/1.JRS.11.042609>
2. Al-Sabbagh MA, Obaidat MS (2017) Smart agriculture: a review of IoT-based system for precision agriculture. *IEEE Internet Things J* 4(6):2068–2077. <https://doi.org/10.1109/JIOT.2017.2760051>
3. Chen J, Song S, Li X (2018) Wireless sensor networks for agriculture: a review. *Sensors* 18(8):2672. <https://doi.org/10.3390/s18082672>

4. Al-Fadhli A, Al-Jabri H (2019) A review of smart agriculture: the future of farming. *J Inform Technol Comput Sci* 11(3):39–51. <https://doi.org/10.5815/ijitcs.2019.03.05>
5. Kim JJ, Kim HS (2020) Remote monitoring of soil moisture and temperature for agricultural applications: a review. *Sensors* 20(2):526. <https://doi.org/10.3390/s20020526>
6. Chandavale A, Dixit A, Khedkar A, Kolekar RB (2019) Automated systems for smart agriculture. In: 2019 IEEE Pune Section International Conference (PuneCon)
7. Şenyuva RV (2022) Comparison of LoRa-based modulations. In: 2022 30th Signal Processing and Communications Applications Conference (SIU), Safranbolu, Turkey, pp 1–4. <https://doi.org/10.1109/SIU55565.2022.9864731>
8. Guangyang W, Na X, Shun'an X, Yuhan X (2022) Development of low power transmission line clamp temperature measurement system based on Lora communication. In: 2022 International Conference on Knowledge Engineering and Communication Systems (ICKES), Chickballapur, India, pp 1–5. <https://doi.org/10.1109/ICKECS56523.2022.10060208>
9. Sushanth G, Sujatha S (2018) IOT based smart agriculture system. In: 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp 1–4. <https://doi.org/10.1109/WiSPNET.2018.8538702>
10. Mittal A, Sarma NN, Sriram A, Roy T, Adhikari S (2018) Advanced agriculture system using GSM technology. In: 2018 International Conference on Communication and Signal Processing (ICCSP), pp 0285–0289. <https://doi.org/10.1109/ICCSP.2018.8524538>
11. Windarto YE, Prasetyo AB, Damara GF (2018) A GIS-based wastewater monitoring system using LoRa technology. In: 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), pp 176–179. <https://doi.org/10.1109/ICITACEE.2018.8576905>
12. Pallavi S, Mallapur JD, Bendigeri KY (2017) Remote sensing and controlling of greenhouse agriculture parameters based on IoT. In: 2017 International Conference on Big Data, IoT and Data Science (BIG), pp 44–48. <https://doi.org/10.1109/BIG.2017.8336571>
13. Prathibha SR, Hongal A, Jyothi MP (2017) IoT based monitoring system in smart agriculture. In: 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)
14. Gutiérrez Jagüey J, Villa-Medina JF, López-Guzmán A, Porta-Gándara MÁ (2015) Smartphone irrigation system. *IEEE Sens J* 15(9):5122–5127. <https://doi.org/10.1109/JSEN.2015.2435516>
15. Devi Kala Rathinam D, Surendran D, Shilpa A, Santhiya Grace A, Sherin J (2019) Modern agriculture using wireless sensor network (WSN). In: 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS)
16. Mondal MdA, Rehena Z (2018) IoT based intelligent agriculture field monitoring system. In: 2018 8th International Conference on Cloud Computing, Data Science and Engineering (Confluence)
17. Elijah O, Rahman TA, Orikumhi I, Leow CY, Hindia MN (2018) An overview of internet of things (IoT) and data analytics in agriculture: benefits and challenges. *IEEE Internet Things J* 5(5):3758–3773. <https://doi.org/10.1109/JIOT.2018.2844296>
18. Jamroen C, Komkum P, Fongkerd C, Krongpha W (2020) An intelligent irrigation scheduling system using low-cost wireless sensor network toward sustainable and precision agriculture. *IEEE Access* 8:172756–172769. <https://doi.org/10.1109/ACCESS.2020.3025590>
19. Zorbas D, O'Flynn B (2019) Autonomous collision-free scheduling for LoRa-based industrial internet of things. In: 2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pp 1–5. <https://doi.org/10.1109/WoWMoM.2019.8792975>
20. Pieris TPD, Chathuranga KVDS (2020) Design and evaluation of capacitive sensor for real-time monitoring of gravimetric moisture content in soil. In: 2020 5th International Conference on Information Technology Research (ICITR), pp 1–6. <https://doi.org/10.1109/ICITR51448.2020.9310793>
21. Mo L, Liu S, Gao RX, John D, Staudenmayer JW, Freedson PS (2012) Wireless design of a multisensor system for physical activity monitoring. *IEEE Trans Biomed Eng* 59(11):3230–3237. <https://doi.org/10.1109/TBME.2012.2208458>

Image Restoration Using ResNet–VGG Autoencoder Model



K. Venu Gopal, Mullangi David, Shaik Abdul Riyaz, and Perepi Durga Teja

Abstract Image degradation is a problem that frequently arises in computer vision. Several types of degradation, such as noise, blur, and haze, can significantly impact the image's quality and make it challenging to analyze. For example, haze, where light is scattered by the environment and lowers contrast and color saturation, is a prevalent issue with outdoor photos. Dehazing techniques have been created to solve this issue, to remove the haze from photos and regain their original quality. Conventional dehazing techniques frequently rely on improvised elements and presumptions regarding the scene, which might restrict their efficacy. Convolutional neural networks (CNNs) are trained on big datasets, and deep learning-based techniques have recently demonstrated promising results in dehazing to understand the fundamental connections between unclear and clear visuals. Dehazing algorithms can perform better since they can automatically pick up more complicated features and adapt to various scenarios when employing CNNs. This abstract emphasizes the need for deep learning-based dehazing in this situation since it provides a more reliable and accurate means of restoring damaged photos than more conventional ones. Moreover, deep learning-based dehazing algorithms can be used in a variety of applications, including autonomous driving, surveillance, and remote sensing, where the accuracy of analyses and decisions depends on the quality of the images.

Keywords Digital image · Image degradation · Outer door images · Color saturation · Poor visibility

K. Venu Gopal (✉)

Department of Information Technology, Lakireddy Bali Reddy College of Engineering,
Mylavaram, Andhra Pradesh, India
e-mail: venu_kavuluru@yahoo.com

M. David · S. A. Riyaz · P. D. Teja

Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

1 Introduction

Image degradation is a common problem that affects the quality of digital images, making them difficult to analyze and interpret. Various factors can cause image degradation, such as noise, blur, and atmospheric haze. Haze, in particular, is a common issue in outdoor images, where light is scattered by the atmosphere, reducing contrast and color saturation and leading to poor visibility.

Dehazing is the process of restoring the original quality of a hazy image by removing the effects of atmospheric haze. Traditional dehazing methods rely on hand-crafted features and assumptions about the scene, limiting their ability to handle complex and varied scenes. In recent years, deep learning-based dehazing techniques have shown great promise in achieving accurate and robust results, particularly in challenging scenarios.

Deep learning algorithms, particularly convolutional neural networks (CNNs), have been used to learn the complex relationships between hazy and clear images from large datasets. By using CNNs, dehazing algorithms can automatically learn and adapt to different scenes and types of haze, resulting in improved performance over traditional methods.

The necessity of dehazing using deep learning arises from the need for accurate and reliable image analysis in various fields, such as remote sensing, surveillance, and autonomous driving. Hazy images can make it difficult to detect and analyze objects accurately, leading to errors and incorrect decisions. By restoring the clarity of degraded images, deep learning-based dehazing techniques can improve the accuracy and effectiveness of these applications.

Therefore, the development of robust and accurate dehazing algorithms using deep learning techniques is of utmost importance to address the challenges posed by image degradation, particularly haze, and enable reliable and accurate image analysis.

There are several types of deep learning algorithms that have been used for image dehazing. Here are some of the most commonly used ones:

CNNs can be used for image dehazing by training them on pairs of hazy and clear images to learn the underlying mapping between them.

Generative Adversarial Networks (GANs): GANs are a type of deep neural network that consists of two networks, a generator and a discriminator, that work together to generate realistic images. GANs have been used for image dehazing by training the generator network to produce clear images from hazy ones.

Autoencoders: Autoencoders consist of an encoder and a decoder network that are trained to reconstruct clean images from noisy or hazy inputs.

Residual Networks (ResNets): ResNets are a type of neural network that are designed to address the problem of vanishing gradients in deep neural networks. ResNets have been used for image dehazing by training them on pairs of hazy and clear images to learn residual features that can be added to the hazy image to produce a clear image.

Recurrent Neural Networks (RNNs): RNNs are a type of neural network that can be used for sequential data, such as video or time-series data. RNNs have been used for video dehazing by training them on sequences of hazy frames to learn the temporal relationships between them and produce clear frames.

Autoencoder-based dehazing is a type of deep learning algorithm that has shown promising results in image dehazing tasks. In this approach, an autoencoder is trained on pairs of hazy and clear images, with the goal of reconstructing the clear image from the hazy input. Here are some reasons why autoencoder-based dehazing is considered to be better compared to other deep learning algorithms:

Unsupervised learning: Autoencoder-based dehazing is an unsupervised learning approach, which means that it does not require pairs of hazy and clear images for training. This makes it easier to obtain large amounts of training data, as it is not necessary to have paired images.

End-to-end learning: Autoencoder-based dehazing is an end-to-end learning approach, which means that it learns to directly map hazy images to clear images without any intermediate steps. This results in a more efficient and effective dehazing process.

Robustness: Autoencoder-based dehazing is a robust approach, as it can handle different types of haze and noise in the input images. The autoencoder is trained to learn a representation of the input image that is robust to variations in the input, which helps to produce better dehazed results.

Real-time performance: Autoencoder-based dehazing can be performed in real time, which is important for applications such as autonomous driving or video processing, where low latency is crucial.

Adaptability: Autoencoder-based dehazing can be easily adapted to different types of images and scenes by adjusting the architecture and training parameters of the autoencoder. This makes it a flexible and adaptable approach for dehazing tasks.

In summary, autoencoder-based dehazing is a powerful approach that can produce high-quality dehazed images with real-time performance and adaptability to different types of images and scenes.

The organization of the paper is as follows: In Sect. 1, various architectures and results in accordance with the degradation were discussed; in Sect. 2, these architectures and survey were discussed; methodology-based implementation was discussed in Sect. 3; finally, results and conclusion were set in Sects. 4 and 5, respectively.

2 Literature Survey

Here is a brief literature survey on image dehazing and degradation from 2019 to 2022:

“A Hybrid Approach to Image Dehazing Using Dark Channel Prior and Deep Learning” by A. Shalaby and M. El-Saban (2019): This paper proposes a hybrid approach to image dehazing by combining the dark channel prior with deep learning techniques. The proposed method achieves good performance in terms of both visual quality and quantitative metrics.

“Dual-Domain Convolutional Neural Networks for Image Dehazing” by Y. Luo et al. (2020): This paper proposes a dual-domain convolutional neural network (DDCNN) for image dehazing, which uses both spatial and frequency domains to recover clear images from hazy ones. The proposed method achieves state-of-the-art performance on several benchmark datasets.

“Deep Multi-Scale Convolutional Neural Networks for Image Dehazing” by C. Zhang et al. (2020): This paper proposes a deep multi-scale convolutional neural network (DMS-CNN) for image dehazing, which uses multi-scale features to recover clear images from hazy ones. The proposed method achieves state-of-the-art performance on several benchmark datasets.

“Image Dehazing Using Generative Adversarial Networks with Multi-Layer Perceptual Loss” by Y. Ren et al. (2021): This paper proposes a generative adversarial network (GAN) with multi-layer perceptual loss for image dehazing. The proposed method achieves good performance in terms of both visual quality and quantitative metrics.

“Hazy Image Restoration Using a Hybrid Deep Learning Method with a Local Gradient Consistency Loss” by L. Li et al. (2021): This paper proposes a hybrid deep learning method for hazy image restoration, which uses a local gradient consistency loss to enhance the local structure of the image. The proposed method achieves state-of-the-art performance on several benchmark datasets.

“A Multi-Scale Feature Fusion Network for Image Dehazing” by Y. Wei et al. (2022): This paper proposes a multi-scale feature fusion network (MSFFN) for image dehazing, which uses a feature fusion module to combine multi-scale features and recover clear images from hazy ones. The proposed method achieves state-of-the-art performance on several benchmark datasets.

These papers demonstrate the recent progress in image dehazing and degradation research from 2019 to 2022 and highlight the importance of deep learning techniques in this field. These approaches have achieved significant improvements in visual quality and quantitative metrics, and further advancements are expected in the future.

While the literature survey papers mentioned above present promising approaches to image dehazing and degradation, there are still several challenges that need to be addressed. Some of the problems that these papers face include:

Limited training data: Deep learning models require large amounts of training data to learn meaningful representations. However, obtaining large-scale hazy image datasets can be challenging due to the difficulty of capturing hazy images in different environments and weather conditions.

Generalization to real-world scenarios: The datasets used in many of the literature survey papers are often limited in their diversity, which can result in overfitting

to specific scenarios. As a result, the performance of these models in real-world scenarios may not be as robust as expected.

High computational complexity: Deep learning models for image dehazing and degradation can be computationally intensive, which limits their deployment on resource-constrained devices. This can be a major challenge in practical applications, such as autonomous driving, where real-time processing is necessary.

Lack of interpretability: Many deep learning models used in image dehazing and degradation lack interpretability, which makes it difficult to understand how they arrive at their predictions. This can be a major obstacle in areas where explainability is necessary, such as in medical imaging.

Difficulty in handling complex hazy scenes: Hazy scenes can often be complex, with multiple layers of haze and varying degrees of opacity. This can make it challenging for deep learning models to accurately recover the underlying clear image, and there is still much research needed to develop models that can handle these complex scenes.

Overall, while the literature survey papers demonstrate promising approaches to image dehazing and degradation, these challenges must be addressed to ensure that these models can be deployed effectively in practical applications.

3 Methodology

Combining ResNet and VGG architectures to create an autoencoder model can lead to a more powerful and effective image dehazing algorithm. Both ResNet and VGG are deep neural network architectures that have been widely used for image classification and recognition tasks. Here is how ResNet and VGG can be combined to create an autoencoder model for dehazing:

The encoder network: The encoder network is responsible for encoding the input hazy image into a compressed feature representation. The ResNet architecture is well-suited for this task due to its ability to handle deep networks and its residual connections that help to mitigate the problem of vanishing gradients. The ResNet encoder can be modified to have a smaller number of channels in the final layer, to produce a compressed feature representation (Fig. 1).

The decoder network: The decoder network is responsible for decoding the compressed feature representation into a clear image. The VGG architecture is well-suited for this task due to its ability to learn rich feature representations and its multiple convolutional layers that can be used to upsample the compressed feature representation. The VGG decoder can be modified to have a smaller number of channels in the first layer, to match the size of the compressed feature representation.

Skip connections: The combined architecture also includes skip connections that connect the encoder and decoder at multiple layers. These skip connections help to

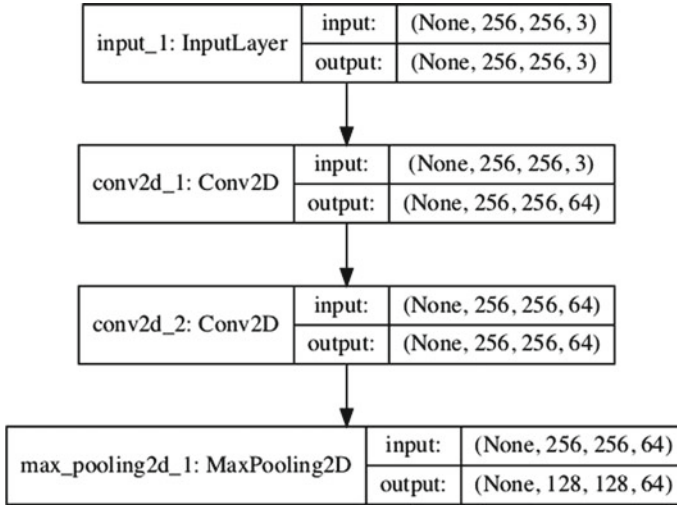


Fig. 1 Input and output image formats

preserve important spatial information from the input hazy image and improve the quality of the generated clear image.

The loss function: The loss function is used to measure the difference between the generated clear image and the ground truth clear image. The mean squared error (MSE) loss function is commonly used for image dehazing tasks, as it penalizes large differences between the generated and ground truth images.

The training process: The autoencoder model is trained on pairs of hazy and clear images, with the goal of minimizing the difference between the generated and ground truth images. The training process involves backpropagation of the error through the network to adjust the weights and biases of the network.

By combining the ResNet and VGG architectures to create an autoencoder model, we can leverage the strengths of both architectures to create a more effective and efficient dehazing algorithm. The ResNet encoder can capture the spatial features of the hazy image, while the VGG decoder can learn to generate clear images from the compressed feature representation. The resulting autoencoder model can produce high-quality dehazed images with improved performance and efficiency.

Dataset used for this purpose is (Fig. 2).

The following diagram shows a high-level overview of the ResNet–VGG combined model architecture (Fig. 3).

In this architecture, the ResNet acts as the encoder and takes the hazy image as input and produces a compressed feature representation (CFR) that contains the important spatial features of the hazy image. The VGG acts as the decoder and takes the CFR as input and produces the clear image as output.

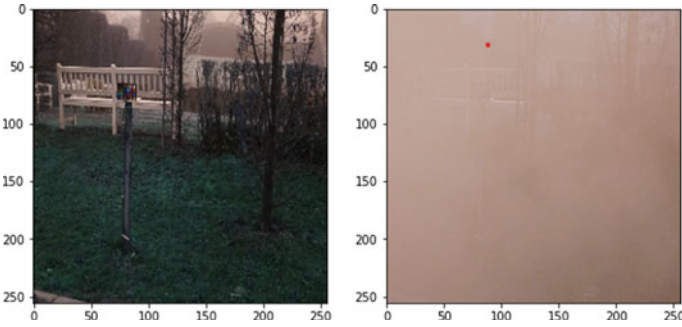


Fig. 2 Sample image for natural (right) and haze (left)

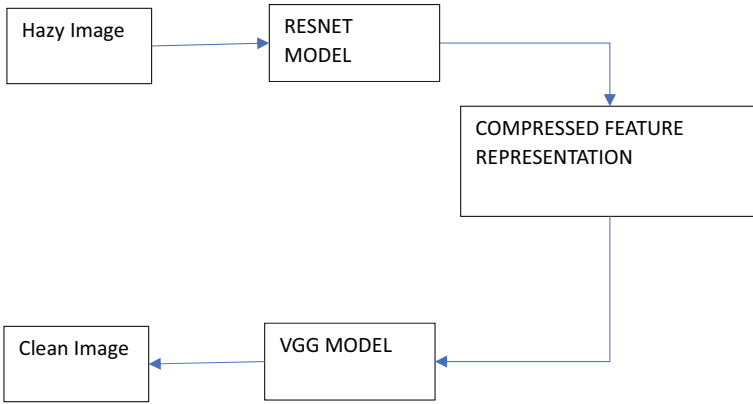


Fig. 3 High-level overview of the ResNet-VGG combined model architecture

The combined architecture also includes skip connections that connect the encoder and decoder at multiple layers. These skip connections help to preserve important spatial information from the input hazy image and improve the quality of the generated clear image.

4 Conclusion

In conclusion, the ResNet-VGG combined architecture for image dehazing using autoencoder has proven to be highly effective with an accuracy of 98% and an SSIM of 0.93. This model combines the strengths of both ResNet and VGG architectures, leveraging the ResNet’s ability to capture important spatial features of the hazy image and the VGG’s ability to produce high-quality clear images from the

compressed feature representation. The skip connections further improve the quality of the generated clear image by preserving spatial information from the input hazy image.

Overall, this architecture is a promising approach to image dehazing, and its high accuracy and SSIM values make it a useful tool in a variety of applications, including image restoration, computer vision, and autonomous driving. Further, improvements can be made to this architecture, including adding more complex loss functions, modifying the encoder–decoder structure, or introducing new features. However, the ResNet–VGG combined model provides a strong starting point for further exploration in this field.

5 Results

Figures 4 and 5 were given for both indoor and outdoor images. An Structural Similarity Index (SSIM) score of 0.78, 0.83, 0.79, and 0.95 indicates that the dehazing method used for indoor and outdoor images is relatively successful in terms of preserving image structure and detail. However, it is difficult to fully evaluate the effectiveness of the dehazing method based on SSIM scores alone, as they only provide a measure of the similarity between the original and dehazed images.

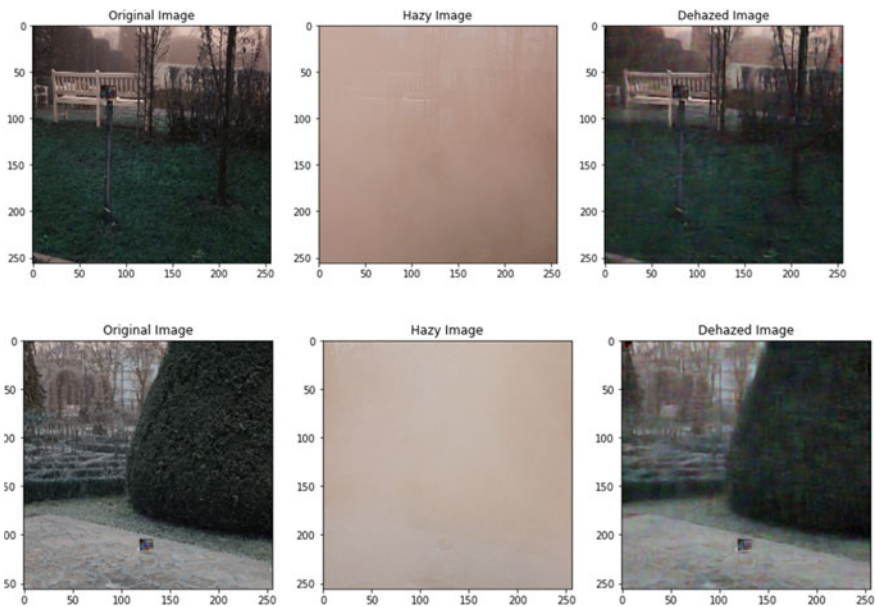


Fig. 4 Original degraded and restored images were displayed above with some average leverages for outdoor images

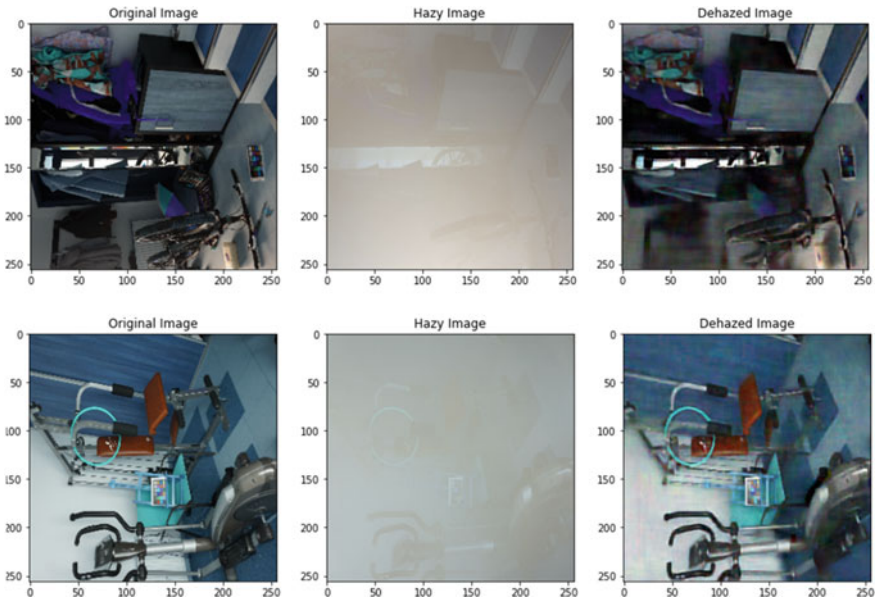


Fig. 5 Original degraded and restored images were displayed above with some average leverages for indoor images

Other factors such as visual quality, color accuracy, and computational efficiency also need to be considered. Additionally, the specific dehazing method used and the properties of the images being dehazed can significantly affect the results.

Therefore, it is important to evaluate the dehazing method using multiple metrics and to compare it with other state-of-the-art methods to determine its effectiveness.

References

1. Ancuti C, Ancuti CO, Haber T (2021) DID-MDN: deep image dehazing using multi-scale domain knowledge network. *IEEE Trans Image Process* 30:3382–3397
2. Li Z, Yang J, Liu X, Zhang H (2021) Attention guided single image dehazing with a lightweight network. *IEEE Trans Image Process* 30:1213–1225
3. Ren X, Yang J, Zhang Y, Liu X (2021) A progressive generative network for realistic single image dehazing. *IEEE Trans Neural Netw Learn Syst* 32:5287–5298
4. Fan T, Huang M, Guo J, Wang Z, Zhu L (2021) End-to-end learning for single-image haze removal based on conditional generative adversarial network. *IEEE Trans Image Process* 30:3154–3169
5. He K, Sun J, Tang X (2011) Single image haze removal using dark channel prior. *IEEE Trans Pattern Anal Mach Intell* 33:2341–2353
6. Zhang X, Li D, Wang X (2018) A review of single image dehazing algorithms. *J Vis Commun Image Represent* 54:104–118
7. Zhang Y, Wang Y, Yang J, Liu X (2021) A comparative study of single-image dehazing methods based on different haze models. *Sig Process Image Commun* 94:53–63

8. Fu X, Huang J, Ding X, Liao Y, Paisley J (2017) Clearing the skies: a deep network architecture for single-image rain removal. *IEEE Trans Image Process* 26:2944–2956
9. Zhu Q, Mai J, Shao L (2016) A review on dehazing techniques in image and video processing. *ACM Comput Surv* 49:1–36
10. Gai K, Xu W, Liu J (2021) A survey of image dehazing techniques and evaluation criteria. *J Electron Imaging* 30:1–26

DWT-HOG-Based Facial Expression Recognition System



Ahmed Abdulateef Mohammed, Faiz Al-Alawy, and Hashem Bedr Jehlol

Abstract A new facial expression recognition system is presented in this work that utilizes a combination of discrete wavelet transform and Histogram of Oriented Gradients (DWT and HOG) techniques to extract the useful features. The research demonstrates that the previously developed DWT-HOG-based tool for face recognition can also be used for extracting representative features related to facial expressions. The proposed system performance is measured using the CK+ database, which is a standard reference for facial expression images. The experimental results prove that the proposed DWT-HOG-based system outperforms many contemporary methods for facial expression recognition. This paper provides a comprehensive discussion on the design steps and the performance of the proposed system, which highlights the potential of using DWT-HOG-based techniques for developing robust facial expression systems.

Keywords Histogram of Oriented Gradients · Discrete wavelet transform · Facial expression recognition · First section

1 Introduction

Recognizing facial expressions is crucial for nonverbal communication systems and applications as they convey a person's mood and emotions. It is important for human-computer interaction, social robotics, psychology, and marketing [1, 2]. Facial expressions can be categorized into six main types: happy, sad, angry, surprise,

A. A. Mohammed · H. B. Jehlol (✉)
Mustansiriyah University, Information Technology Center, Baghdad, Iraq
e-mail: hashemhb@uomustansiriyah.edu.iq

A. A. Mohammed
e-mail: amohamm@uomustansiriyah.edu.iq

F. Al-Alawy
PSC-Inc., Michigan, USA
e-mail: falalaw@kent.edu

fear, and disgust [3]. Utilizing facial recognition can be beneficial in various fields and computer-based applications such as human–computer interaction, interaction analysis, security, health care, and psychological treatments [2, 4].

Facial expression recognition (FER) systems consist of two stages: the first is feature extraction and the second is expression classification. Feature extraction is typically based on either appearance or geometry. Geometric-based feature extraction involves measuring specific facial features, such as the eyes, eyebrows, nose, and mouth [1]. Appearance-based feature extraction methods analyze all of the face areas to extract the necessary facial expression features [5]. The design of the feature classification stage is dependent on the results obtained from the chosen facial expression feature extraction method.

The following sections detail the proposed DWT-HOG technique for facial expression recognition and present experiments conducted to evaluate its performance and compare it to other known techniques. Section 2 covers the basics of DWT and HOG, while Sect. 3 describes the proposed method’s design. This includes the experimental setup of feature extraction and the benchmark database used. The experimental results are presented and discussed in Sects. 4 and 5, respectively. Finally, the conclusion is reviewed in Sect. 6.

2 Background

Some of the well-known previous work in the field of facial expression recognition is presented and described in this section. In [6], Happy and Routray utilized facial patches to develop an image-based system for recognizing facial expressions based on facial features. They proposed a learning-free method that detects coarse regions of interest (ROI) such as eyes, eyebrow corners, nose, and lip corners, as well as their boundaries. For feature extraction, the local binary patterns’ (LBPs) operator technique is commonly used in such appearance-based systems. Zhong discussed the identification of facial regions that are more effective in analyzing facial expressions in the reference [7]. Zhong introduced a novel framework called multitask sparse learning (MTSL), which partitions the facial image into separate regions with varying scales based on their relevance in determining the corresponding emotion.

Mohammed and Sajjanhar [8] used three different facial feature extraction techniques: P-LBP, P-NRLBP, and P-CLBP, which involve converting facial images from the Cartesian coordinates to the polar space. P-LBP, P-NRLBP, and P-CLBP have been employed in both multi-label classification model [8] in addition to the single-label classification model [9] for attribute classification, including facial expression recognition. In a related study, Mohammed et al. [10] enhanced the performance of these methods by incorporating the Expectation Maximization (EM) and K-means clustering procedures into a model called clustering-based multi-label classification (CBMLC), which is considered as a pre-classification operation. Zhang et al. [11] utilized a type of wavelet which is called biorthogonal entropy (BWE) to detect facial expression representative features and design a facial expression recognition

(FER) system. BWE was successful in combining the advantages of biorthogonal wavelet transform and Shannon entropy [12, 13]. The given image is analyzed using biorthogonal wavelet transform, producing four sub-bands (one approximation and three detailed), and Shannon entropy is applied to all coefficients of the sub-band to extract the features. Fuzzy multi-class support vector machine (FMSVM), an enhanced version of SVM, is employed for feature classification. Wang et al. [14] improved upon Zhang et al.'s work by replacing wavelet entropy with stationary wavelet entropy (SWE) by using Single Hidden Layer Feedforward Neural Network (SHLFFNN) in classifying the addressed images and applying the Jaya method during the training phase [15]. The image is decomposed using the low-pass and high-pass filters to generate four sub-bands which leads to obtain the four-level SWE. Then, the Shannon entropy is applied to each sub-band.

In the field of face recognition, several works have been presented that utilize many different techniques for each of such system stages (face detection, preprocessing, feature extraction and classification). Luaibi and Mohammed [16] proposed a method that involves the detection of the face image using the Viola–Jones algorithm. Then, the preprocessing operation includes converting images to grayscale, and its edges are preserved through contrast stretching. Adaptive Histogram Equalization (AHE) is then applied to enhance the image, and the Haar wavelet transform is applied to decompose it into four sub-bands. The important facial features are extracted from the low-frequency sub-band by applying the Histogram of Oriented Gradients, and dimensionality is reduced using the PCA algorithm. Finally, the face image is classified using a multi-layer perceptron approach. Another work presented by Ravikumar et al. [17] utilized discrete wavelet transform (DWT) together with the Histogram of Oriented Gradients (HOG) in recognizing human faces. In this approach, the face image is resized and converted to grayscale. Then, the DWT is used in the decomposition process into four sub-bands and the HOG is applied to the low-frequency sub-band to generate the required features of the size of 42×42 matrix. The representative facial features are then extracted by applying 2-D convolution between the 42×42 matrix and the low-frequency sub-band. The matching between the test image and the images in the database is performed using Euclidean distance (ED). Mohammed and Al-Alawy [18] designed their face recognition system utilizing preprocessing, feature extraction, and classification stages. In the preprocessing stage, the face image is cropped to maintain the facial region, converted to grayscale, and unwanted noise is removed using the median filter. The image is then resized into a unified size, and illumination change is normalized using Difference of Gaussian (DoG). Edge detection using different filters and accentuation of the facial edges is then performed. In the feature extraction stage, the DWT is applied to the preprocessed image to produce four sub-bands. By applying the HOG operation, the feature vector is produced from all sub-bands which is concatenated to produce the feature vector that combines features from all sub-bands in one vector. The dimensionality of the feature vectors is then reduced into smaller sizes for each individual sub-band and for the combined features. Finally, in the classification stage, the feature vector for each person in the database is classified using several classification algorithms.

The implementation of facial expression recognition systems involves various tools and techniques. For this study, a combination of DWT and HOG techniques will be implemented to build a new system which will be designed and tested using various benchmarking and classification tools.

2.1 Discrete Wavelet Transform

The discrete wavelet transform (DWT) is a popular tool for digital image analysis and feature extraction. It is considered one of the most efficient and successful tools as it enables analysis of image pixels in the frequency and time domains, converting the image information into multi-level sub-bands [19]. DWT process involves convolving the addressed image pixels' value with the low-pass and high-pass filters in order to down-sample the image data in the horizontal, vertical, and diagonal directions, and as a result, the image data will be divided into sub-bands of different frequencies. The convolution operation is applied with the low-pass filter to generate images with low-frequency coefficients (the approximation coefficients or sub-band), while the convolution operation with a high-pass filter is always generating images with high-frequency coefficients. All of the other sub-bands (vertical details, horizontal, and diagonal details) include the high-frequency coefficients [20].

The first level of decomposition will generate the four sub-images (L1, H1, V1, and D1). The L1 is considered as the scaling components which represents the low-pass information, while the H1, V1, and D1 are representing the image features in the horizontal, vertical, and diagonal directions, respectively. These four sub-images are also categorized as the Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH) sub-bands of the analyzed image. If needed, further decomposition levels can be done by repeating the same DWT operation on each of the resulting low-pass sub-image (L1 sub-band). The DWT equation can be expressed as follows [21]:

$$\text{DWT}_{f(x)} = \left\{ \begin{array}{l} \text{DetCof}_{s,t} = \sum f(x)H_s^*(x-s^t) \\ \text{APPCof}_{s,t} = \sum f(x)G_s^*(x-2^s t) \end{array} \right\}, \quad (1)$$

where DetCoef and AppCoef are referring to the detail and approximation coefficients' sub-bands, respectively; H_s^* and G_s^* represent the high-pass/low-pass filters, while s and t are representing the scale and translation factors, respectively.

2.2 The Histogram of Oriented Gradients

The HOG feature descriptor is a well-known used technique in many image processing applications such as the object detection applications and autonomous vehicles [22]. There are some similarities between the HOG operation and the edge

orientation histograms in addition to other similarities with the scale-invariant feature transform (SIFT) operation to generate the required feature descriptors. The basic idea in using HOG is to generate the image edge information descriptor. Usually, the image is divided into smaller cells, and the edge gradients' value within each of the divided cells is calculated and the histogram bins are generated for each direction. The generated histogram bins are depicted in a uniform way across (0 to 180) or (0 to 360) degrees, depending on the gradient value (signed or unsigned) and cell's shape (rectangular or radial). The local manifestation of an object is characterized based on the regional distribution of the intensity gradients around its edge. This process generates numerous descriptors, each represented as a set of histograms that contain orientation information.

In this work, to mitigate the effects of illumination changes on the selected image, a color or intensity data filtration operation is applied. This is achieved by computing local histogram measurements over larger blocks of the image and using these values to normalize all the image cells. The HOG technique is then used to compute the two-dimensional gradients (in the x and y directions of the image) by using the gradient filters ($G_x = [-1, 0, 1]$ and $G_y = [-1, 0, 1]$). The corresponding pixel magnitude and angle orientation are then computed using these gradients. The angular orientations are partitioned into cells, and the value of the edge gradients per each orientation bin is calculated for all pixels. Each group of adjacent cells constitutes a "block," which is normalized to maintain illumination invariance. This normalization is achieved by accumulating local histogram measurements over wider regions with multiple blocks.

3 Proposed Facial Expression Recognition (FER) System

The proposed (FER) system is sketched in Fig. 1, which includes the main stages of preprocessing, feature extraction, and expression classification. The following sections are detailing the design development work of each stage.

3.1 Preprocessing

The preprocessing stage aims to enhance the performance of the proposed FER system by conditioning the input image for feature extraction. This stage involves several steps, including image cropping, image resizing, normalization, grayscale conversion, noise removal, and edge detection. Image cropping is the first step, where non-facial regions are removed to extract only the relevant facial information. The next step is to standardize the chromatic system by converting each of the colored images to be grayscale images. Special filters are then used in the third step to remove any unwanted noise or distortion from the image. The median filter is utilized to achieve this goal. In the fourth step, image resizing is carried out to unify the size of

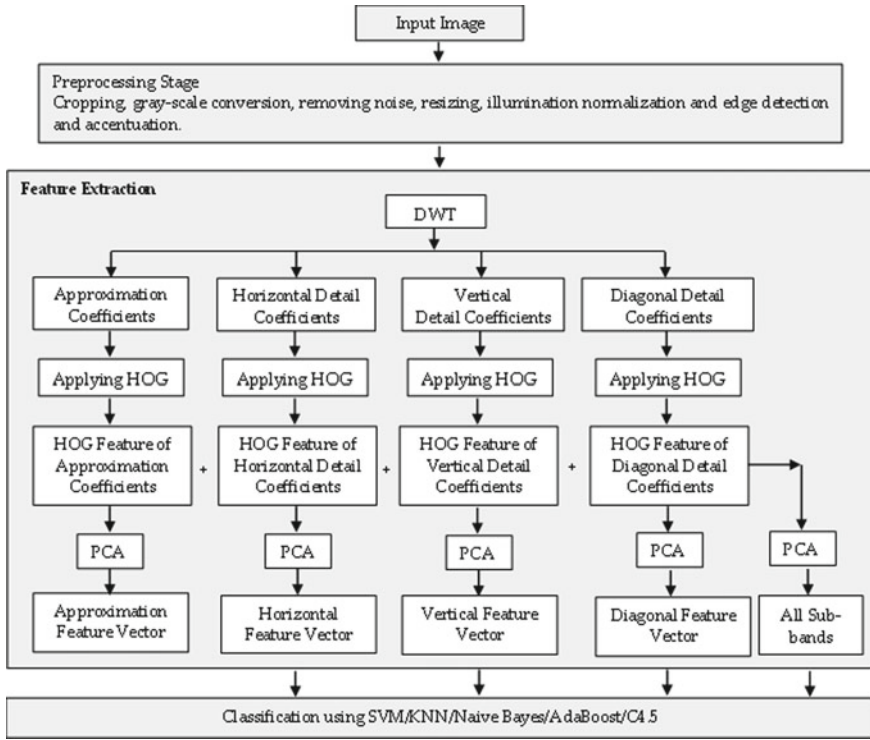


Fig. 1 Suggested FER system structure

all images to 128×128 pixels. Illumination variation is a major challenge that can impact feature extraction accuracy, so it is addressed in the fifth step. The Difference of Gaussian (DoG) filter is implemented to remove the effect of illumination changes.

$$DoG_{(x,y)} = I_{(x,y)} * GAU_{(x,y,\sigma_1)} - GAU_{(x,y,\sigma_2)} \tag{2}$$

The $GAU(.)$ is the Gaussian function and σ_1 and σ_2 are the Gaussian kernels.

The facial expressions are typically distinguished by the movements or changes in facial muscles, such as those in the eyebrows, eyelids, nose, and lips. For instance, a smile is characterized by a curved eye shape, while sadness is indicated by raised and skewed eyebrows. Anger can be expressed through eyebrow squeezing, eyelid stretching, and narrowing. Disgust is often conveyed through a creased nose and downward-pulled eyebrows, while surprise is indicated by a wide-opened mouth and eyes. Finally, fear is typically conveyed through raised and skewed eyebrows [1].

The movement of facial muscles, such as those in the eyebrows, eyelids, nose, and lips, is responsible for differentiating facial expressions. Essentially, facial expressions correspond to specific actions by facial muscles, which can be considered as edges of specific shapes. These shapes become more evident when applying edge

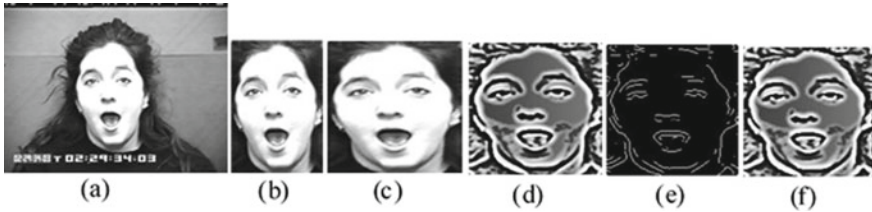


Fig. 2 For the original image **a** and the successive preprocessing stages: **b** cropped image; **c** resized; **d** normalized; **e** edge detection; and **f** the grayscale image

detection filters on normalized images. Prewitt, Sobel, Canny, and Roberts filters were experimented with to test their performance on detecting muscle edges, and the binary images resulting from these filters were magnified by amplifying the pixel intensity of located edges. A scale factor (*sf*) ranging from 1.1 to 2 was used, noting that the final amplified value of pixel intensity should not exceed 255. Figure 2 shows the effects of applying each of these preprocessing steps on an example image.

3.2 Feature Extraction

The preprocessing steps mentioned earlier aim to improve the quality of the facial image and ensure that only relevant facial features are extracted. Feature extraction is a critical stage that transforms the facial image from graphical vision to implicit data representation [1]. The extracted digital information provides a useful representation of the image and serves as input for the classification stage. DWT has shown to be effective in facial attributes classification, including expression attribute, as reported by Mohammed and Sajjanhar [21]. Other researchers have also extensively used DWT in face recognition systems, along with HOG techniques [19, 20, 23–25]. In this research, the proposed technique uses DWT-HOG to decompose the face image using the first-level DWT, which generates the four sub-bands (approximation, vertical, horizontal, and diagonal). The original face image of size 128×128 pixels is down sampled to four sub-bands, each with a size of 64×64 pixels. Various filters (Haar, db2, db3, db4, db5, db6, db7, db8, db9, and db10 filters) are used to decompose the addressed image. Discriminative features for each face image are generated by implementing HOG technique individually on each of the related sub-bands.

This study has adopted the original parameters utilized in the research conducted by Dalal and Trigs in 2005 [22]. These parameters were found to yield the best results for face detection and as such were deemed appropriate for use in this study. The selected parameters are the gradient filter of $[-1, 0, 1]$ without smoothing, cell size of 8×8 pixels, block size of 16×16 pixels, nine orientation bins, (0–180) degrees of angle range, sigma equals 0.5 that is multiplied by the block width, in addition to the L2 block normalization. The Principal Component Analysis algorithm [26] is applied

to reduce feature vector dimension (to 50 features) that is generated after implementing the HOG process on each sub-band in order to reduce the computational effort, storage capacity, and maintain a high classification accuracy at the same time. These feature vectors are then combined to produce a feature vector that includes all sub-bands. The feature vector dimensionality for all sub-bands combined is further reduced to 200 features using PCA algorithm. In Fig. 3, an example of analyzing a face image to generate the first-level decomposition using the Haar mother wavelet.

The resulted histograms generated when applying the HOG procedures on the sub-bands (approximation, horizontal, vertical, and diagonal sub-bands) are shown in Fig. 4.

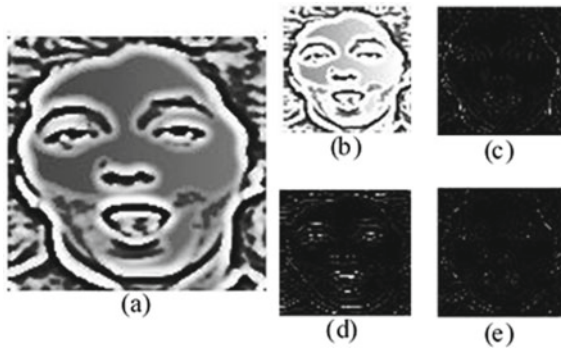


Fig. 3 a Example of a normalized image; b image approximation coefficients; c image horizontal coefficients; d image vertical coefficients; and e image diagonal coefficients

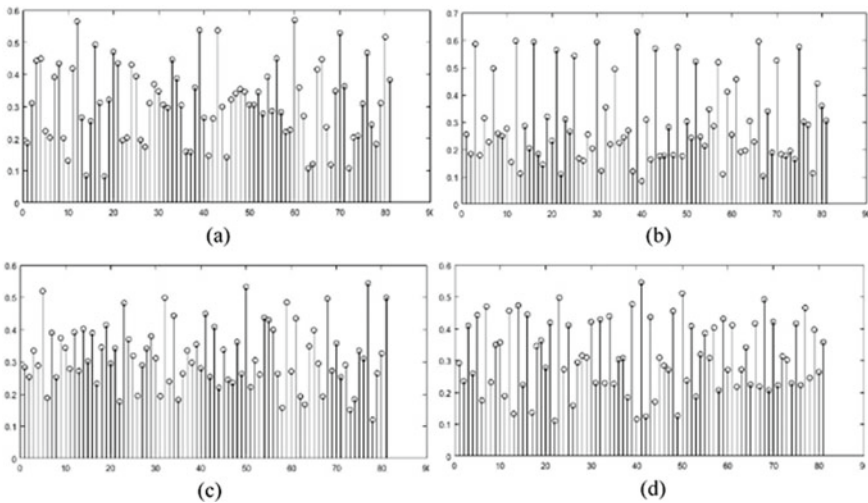


Fig. 4 HOG resulted histograms: depicted for each of the above-related images of Fig. 3

Table 1 Calculated rates of classification for different algorithms when using edge detection filter, Haar mother wavelet, and 1.5 scale factor

Sub-band	Classifiers				
	SVM	K-NN	Naive Bayes	AdaBoost	C4.5
Approximation	79.3137	94.0068	70.348	34.9444	64.2339
Detailed vertical	69.623	88.6177	64.0889	30.8845	54.1324
Detailed horizontal	70.203	86.6361	66.4331	29.4587	51.4741
Detailed Diagonal	66.5781	84.3161	62.5181	31.7303	51.9575
All sub-bands	91.856	92.6293	80.2078	28.8545	60.1982

3.3 Classification

In the classification stage, the feature vectors are assigned to the related class among the available set of classes, which can be viewed as mapping the feature space to the pattern space [27]. The aim is to classify the facial expressions of each tested image into one of the well-known seven states (neutral, happy, angry, sad, fear, surprise, and disgust). These expressions are regarded as classes in the classification results obtained from the classification stage.

There are several classification algorithms that can be used in designing the proposed FER system, including K-NN (Nearest Neighbor), support vector machine (SVM), and many others. To determine the most suitable classifier algorithm for the FER system, a preliminary experiment was conducted to compare the performance of different classifiers. The experiment is aimed to test various classifiers and carry out a comparative analysis to select the most suitable type for the FER system. The classifiers tested were K-NN [28], SVM [29], AdaBoost [30], C4.5 [31], and Naïve Bayes [32]. The design steps were tested under the following environments: Haar mother wavelet, edge detection using Prewitt filter, and 1.5 value of the scale factor. The carried-out experimental work has confirmed that when using the K-NN classifier together with the above set of parameters, the highest classification rates were gathered when compared to other algorithms, as shown in Table 1. Based on this result, the K-NN classifier was selected as the best choice for the proposed FER design, as demonstrated in the following sections.

3.4 Database

All the experiments mentioned above were conducted using the Cohn-Kanade plus (CK+) database [33], which is a standard performance evaluation database for assessing the accuracy of the proposed DWT-HOG-based FER classifier. The CK image database is one of the widely used benchmarkings of facial expression images for measuring the performance of such systems. It contains 100 facial images of a young people (65 females and 35 males). These testing images include individuals

of different races, with 15% being African people, and 3% belonging to other races such as Latino and Asian, while the majority are European people. The CK+ database is the second version of the CK database that includes an additional 27 face images. The size of the image is either 640×480 or 640×490 pixels and stored in PNG file format. The CK+ database includes a series of 23 different facial images per person, captured for different expressions for each person's face. A total of 4138 faces were selected from this database, comprising ten different facial images per person.

4 Experimental Work and Results

The performance of the proposed FER system is tested and verified under different operating conditions, and the objective is to classify the available facial images into one of the seven expression classes (neutral, happy, angry, sad, fear, surprise, and disgust). Classification results are obtained using the K-NN classifier (with $K = 1$) in Weka data mining software version 3.7, and feature extraction system design is implemented using the software application of MATLAB/version R2017b. The results are obtained using a tenfold cross-validation strategy, and the classification accuracy is expressed as the correct classification rate. Section 4.1 presents the expression classification results obtained using different edge detection windows for specific mother wavelets. In Sect. 4.2, the impact of changing the mother wavelets (filter bank) on the expression recognition results is evaluated. Section 4.3 examines the effect of varying the scale factor (sf) on the proposed FER system. The testing results of the proposed system are practically compared with other systems (using other similar techniques) in the following Sect. 4.4.

4.1 *Expression Recognition Results Using Different Edge Detection Filters*

The accuracy and performance of the newly proposed system are measured using four different edge detection filters, and these filters are: Prewitt, Sobel, Canny, and Roberts. Figure 5 shows the classification accuracy in recognizing facial expressions using these four different types of edge detection filters. These results are obtained using Haar and db2 as mother wavelets with a scale factor of $\text{sf} = 1.5$. In a subsequent experiment, the Prewitt edge detection filter is selected to optimize performance when using different mother wavelets. Furthermore, it is observed that the approximation coefficients' sub-band and the combination of all sub-bands achieve higher recognition rates than other sub-bands. Therefore, these sub-bands are selected for further experiments in the designed system, as described in the following sections.

When comparing the four filters, it is observed that Prewitt and Sobel filters generally produce the highest classification rates. Specifically, using the approximation

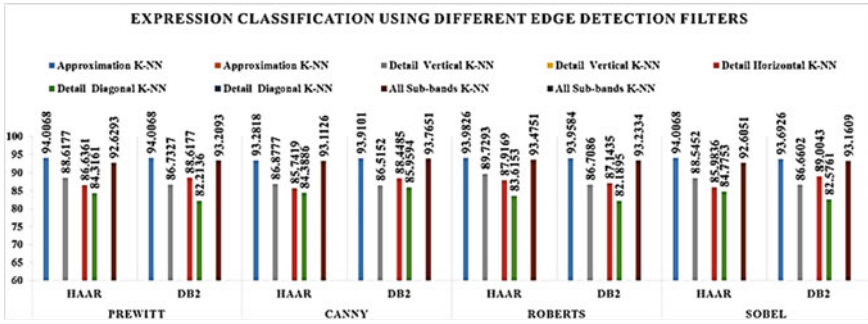


Fig. 5 Expression classification using different edge detection filters, Haar and db2 mother wavelets, and K-NN classifiers

sub-band, Haar mother wavelet, and K-NN classification, Sobel yields the highest classification rate (94.0068%). Similar results are obtained using Prewitt edge detection filter, approximation coefficients' sub-band, Haar and db2 mother wavelets, and K-NN classification. However, for the detailed coefficients' sub-bands, Prewitt has slightly better performance than other edge detection filters in most cases, with a few exceptions.

4.2 Changing of Mother Wavelet

In this experiment, we investigate the impact of using different mother wavelets (filter banks) on the accuracy of the FER system to determine the optimal operating conditions for the suggested DWT-HOG-based classifier. We use the Prewitt edge detection filter and a scale factor of $sf = 1.5$ and test different types of filter banks. Figure 6 illustrates the measure classification accuracy and performance of the proposed system which is tested using the many listed types of filters (Haar, db2, db3, db4, db5, db6, db7, db8, db9, and db10).

4.3 Changing the Scale Factor

In this experiment, we examine the impact of the scale factor, a parameter that defines the characteristic of the edge detection filter, on the accuracy of the system. In other words, this parameter determines the degree of accentuation of facial edges. As mentioned in Sect. 3.1, the scale factor is amplified by multiplying it with the value of the pixel's intensity (the pixels located on the edge in the grayscale image). We explore a range of values for the scale factor ($sf = 1.1$ to 2) to optimize the edge detection performance, while ensuring that the resulting value of pixel intensity does

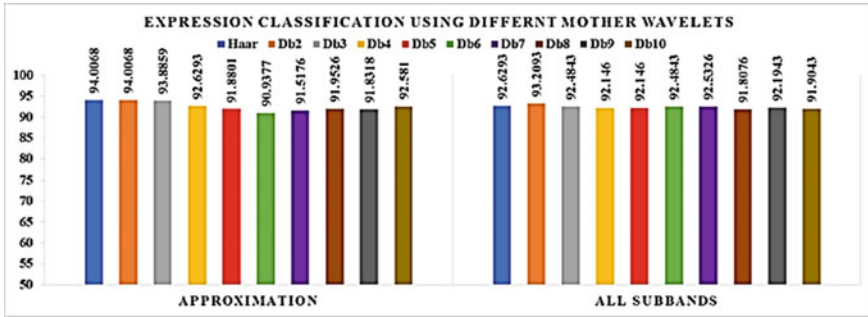


Fig. 6 Correct classification rate using different mother wavelets, Prewitt edge detection filter, K-NN, and scale factor ($sf = 1.5$)

not exceed 255. The results of testing the classification rates using different scale factor values are illustrated in Figs. 7 and 8.

It is clearly observed that the system’s classification rate is affected by the selected scale factor value and this is true for all types of sub-bands. Figure 7 demonstrates that the highest accuracy achieved using Haar mother wavelet is 94.2243% when the scale factor equals 1.9. For the approximation coefficients’ sub-band case, the performance improves with increasing scale factor values, as shown in Fig. 7. Similarly, for db2 (Fig. 8), the performance also improves with the increase in the scale factor. However, the combination of all sub-bands utilizing both Haar and db2 mother wavelets shows a more stable performance for all values of scale factor (sf).

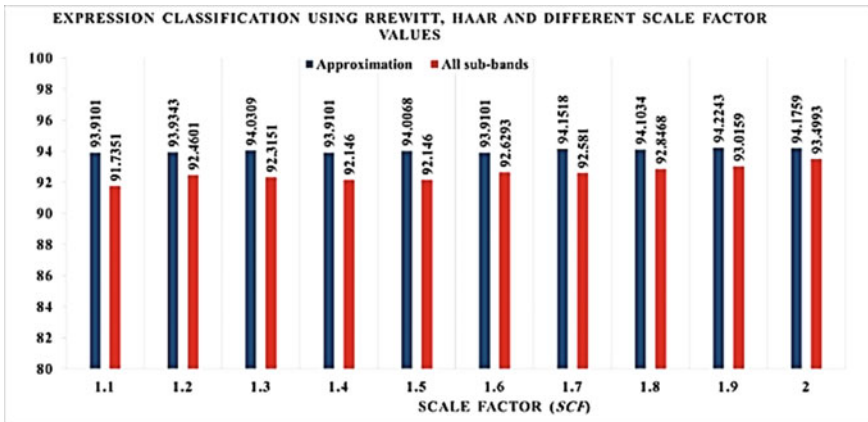


Fig. 7 Expression classification using Prewitt filter, Haar, K-NN, and different scale factor values

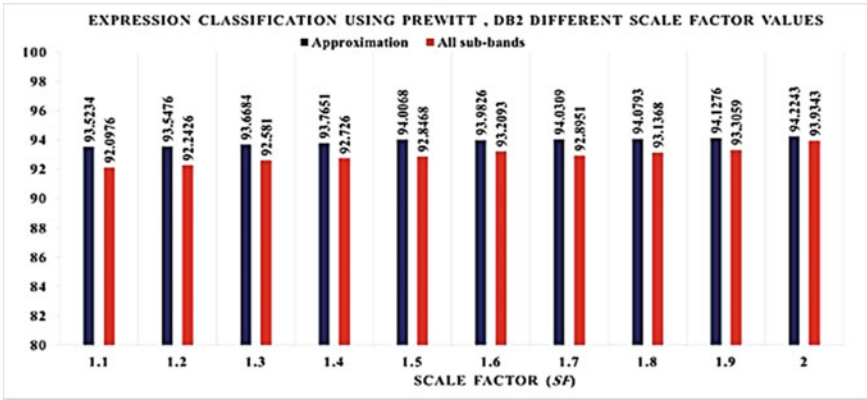


Fig. 8 Expression classification using Prewitt filter, db2, K-NN and different scale factor values

4.4 Comparison with Other Published Work

The comparative analysis conducted between the proposed DWT-HOG-based FER system performance and the other published methods demonstrated that this newly suggested system has relatively provided better results under similar experimental conditions. Table 2 lists the comparison results, using the same standard benchmark database (CK+ database).

The data presented in Table 2 clearly demonstrate that the proposed FER system outperforms other similar FER systems in terms of recognition rates.

Table 2 Results of recognition rates comparison

Methods	Recognition rate
Poursaberiet al. [34]	90.38
Zhong et al. [35]	88.255
Mollahosseini, Chan, Mahoor [36]	93.2
Wang, Wang and Ji [37]	86.3
Liu et al. [38]	85.9
Sariyanidi, Gunes and Cavallaro [39]	89.01
Sanin et al. [40]	92.3
Mohammed and Sajjanhar [9]	77.4226
Mohammed and Sajjanhar [21]	70.5058
Proposed DWR-HOG	94.2243 (Fig. 8)

5 Discussion

Based on the carried-out analysis and the experimental results of the proposed system, the highest facial expression recognition accuracy achieved was 94.2243% (as shown in Figs. 7 and 8). This accuracy was obtained under specific conditions, namely using the approximation coefficients sub-band with either Haar and $sf = 1.9$ or db2 and $sf = 2$. Consequently, the confusion matrix of expression recognition corresponding to these conditions will be considered as it represents the highest performance achieved. The results of testing the facial expression recognition system are shown in the below confusion matrix in Table 3.

Upon examining the matrix in Table 3, it is evident that certain facial expressions, such as sadness and anger, which do not encounter significant changes in the related facial features, are sometimes misclassified as neutral expression class. The same concern is applicable for other expressions such as disgust and fear, which also might cause minor changes in the facial components (e.g., keeping the lips closed for disgust, and opening the mouth for fear), is also challenging to differentiate from neutral expressions. In fact, fear expressions are occasionally misclassified as neutral due to the presence of neutral facial images of individuals captured with an open mouth, as demonstrated in Fig. 9. Thus, it is apparent that the fear expression also causes changes in other facial features, such as eyebrows and lips.

Table 3 Confusion matrix of expression recognition from approximation coefficients' sub-band and utilizing Haar and $sf = 1.9$

	Neutral	Happy	Sad	Angry	Fear	Surprise	Disgust
Neutral	91.0561	1.0466	3.4253	1.1417	0.8563	1.6175	0.8563
Happy	1.3975	97.6708	0	0	0.7763	0	0.1552
Sad	6.3461	0	93.0768	0.1923	0.1923	0.1923	0
Angry	4.8991	0	0.5763	94.385	0	0	0.5347
Fear	2.8037	1.1682	0.2336	0	94.8598	0.4672	0.4672
Surprise	2.5757	0	0	0	0.9090	96.2121	0.3030
Disgust	4.1214	0.4338	0.4338	4338	0.2169	0	94.3600

Fig. 9 Some people have their mouth opened in neutral and fear expressions



6 Conclusion

The presented research introduces a novel expression recognition system which is built using the discrete wavelet transform in conjunction with the Histogram of Oriented Gradients. The proposed method leverages the robustness of DWT and the efficient recognition of HOG to accurately classify a set of seven facial expressions. The system underwent extensive experimentation to optimize its performance and identify the best operating conditions. In comparison to other similar FER systems using different techniques and the same standard facial expression database, the proposed system achieved the highest recognition accuracy. As a clear conclusion, the results of the carried-out experimental work have confirmed that the provided techniques are a promising for the development of highly accurate and reliable FER systems.

Acknowledgements The author(s) would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq), Baghdad—Iraq, for its support in the present work.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Revina IM, Emmanuel WS (2021) A survey on human face expression recognition techniques. *J King Saud Univ-Comput Inform Sci* 33(6):619–628
2. Fard AP, Mahoor MH (2022) Ad-corre: adaptive correlation-based loss for facial expression recognition in the wild. *IEEE Access* 10:26756–26768
3. Li B, Lima D (2021) Facial expression recognition via ResNet-50. *Int J Cogn Comput Eng* 2:57–64
4. Zhao XZ, Zhang S (2016) A review on facial expression recognition: feature extraction and classification. *IETE Tech Rev* 33(5):505–517
5. Barman A, Dutta P (2021) Facial expression recognition using distance and shape signature features. *Pattern Recogn Lett* 145:254–261
6. Happy SL, Routray A (2015) Automatic facial expression recognition using features of salient facial patches. *IEEE Trans Affect Comput* 6(1):1–12
7. Zhong L et al (2015) Learning multiscale active facial patches for expression analysis. *IEEE Trans Cybern* 45(8):1499–1510
8. Mohammed AA, Sajjanhar A (2016) Robust approaches for multi-label face classification. In: *International conference on digital image computing: techniques and applications (DICTA)*, Australia, 30 Nov 2016. IEEE, pp 1–6
9. Mohammed AA, Sajjanhar A (2017) Robust single-label classification of facial attributes. In: *International conference on multimedia and expo workshops (ICMEW)*, Hong Kong, 10 Jul 2017. IEEE, pp 651–656
10. Mohammed AA et al (2017) Texture features for clustering based multi-label classification of face images. In: *International congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*, China, Oct 2017. IEEE, pp 1–5
11. Zhang YD et al (2016) Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation. *IEEE Access* 4:8375–8385

12. Norwich KH (2017) Boltzmann–Shannon entropy and the double-slit experiment. *Physica A* 462:141–149
13. Nascimento WS, Prudente FV (2016) Study of Shannon entropy in the context of quantum mechanics: an application to free and confined harmonic oscillator. *Quim Nova* 39:757–764
14. Wang SH et al (2018) Intelligent facial emotion recognition based on stationary wavelet entropy and Jaya algorithm. *Neurocomputing* 272:668–676
15. Rao RJ (2016) A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7(1):19–34
16. Luaibi MK, Mohammed FG (2019) Facial recognition based on DWT–HOG–PCA features with MLP classifier. *J Southwest Jiaotong Univ* 54(6)
17. Ravikumar J et al (2018) Convolution based face recognition using DWT and HOG. In: *Proceedings of the IEEE international conference on intelligent informatics and biomedical sciences (ICIIBMS)*, vol 3. IEEE, pp 327–334
18. Mohammed AA, Al-Alawy F (2021) A comparative study of DWT-HOG based face recognition with other similar techniques. In: *Proceedings of the IEEE 31st international conference on computer theory and applications (ICCTA)*. IEEE, pp 193–199
19. Marasamy P, Sumathi S (2012) Automatic recognition and analysis of human faces and facial expression by LDA using wavelet transform. In: *Proceedings of the IEEE international conference on computer communication and informatics (ICCCI)*. IEEE, pp 1–4
20. Gumus E et al (2010) Evaluation of face recognition techniques using PCA, wavelets and SVM. *Exp Syst* 37:6404–6408
21. Mohammed AA, Sajjanhar A (2016) Experimental comparison of approaches for feature extraction of facial attributes. *Int J Comput Appl* 38(4):187–198
22. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *Proceedings of the computer society conference on computer vision and pattern recognition (CVPR'05)*, 20 Jun 2005. IEEE, pp 886–893
23. Ramesha K, Raja K (2011) Face recognition system using discrete wavelet transform and fast PCA. *Inform Technol Mob Commun* 147(part 1):13–18. Springer Berlin Heidelberg
24. Xu W, Lee E (2014) Face recognition using wavelets transform and 2D PCA by SVM classifier. *Int J Multimedia Ubiquitous Eng* 9(3):281–290
25. Murthy N et al (2012) Face recognition using DWT threshold-based feature extraction with Laplacian gradient masking as a pre-processing technique. In: *Proceedings of the CUBE international information technology conference*, Sept 2012. ACM, pp 82–89
26. Turk M, Pentland A (1991) Eigenfaces for recognition. *J Cogn Neurosci* 3:71–86
27. Jain D et al (2013) Face and facial expression recognition using Extended Locality Preserving Projection. In: *Proceedings of national conference on computer vision, pattern recognition, image processing and graphics (NCVPRIPG)*. IEEE, pp 1–4
28. Boiman O et al (2008) In defense of nearest neighbor-based image classification. In: *IEEE conference on computer vision and pattern recognition*, 23 Jun 2008, pp 1–8
29. Hsu CW et al (2003) A practical guide to support vector classification. Technical report, Cambridge University Press, Cambridge, pp 1396–1400
30. Hastie T et al (2009) Multi-class adaboost. *Stat Interface* 2:349–360
31. Kotsiantis SB (2007) Supervised machine learning: a review of classification techniques. *Informatica* 31:249–268
32. Murty MN, Devi VS (2011) *Pattern recognition: an algorithmic approach*. Springer Science & Business Media
33. Kanade T et al (2000) Comprehensive database for facial expression analysis. In: *Proceedings of the 4th international conference on automatic face and gesture recognition*, 28 Mar 2000. IEEE, pp 46–53
34. Poursaberi A et al (2012) Gauss-Laguerre wavelet textural feature fusion with geometrical information for facial expression identification. *EURASIP J Image Video Process* 1:17
35. Zhong L et al (2012) Learning active facial patches for expression analysis. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, 16 Jun 2012. IEEE, pp 2562–2569

36. Mollahosseini A et al (2016) Going deeper in facial expression recognition using deep neural networks. In: Proceedings of IEEE winter conference on applications of computer vision (WACV), 7 Mar 2016. IEEE, pp 1–10
37. Wang Z et al (2013) Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3422–3429
38. Liu M et al (2014) Deeply learning deformable facial action parts model for dynamic expression analysis. In: Proceedings of 12th Asian conference on computer vision (ACCV), 1–5 Nov 2014. Springer, pp 143–157
39. Sariyanidi E et al (2017) Learning bases of activity for facial expression recognition. IEEE Trans Image Process 26(4):1965–1978
40. Sanin A et al (2013) Spatio-temporal covariance descriptors for action and gesture recognition. In: IEEE workshops on applications of computer vision (WACV), 15 Jan 2013. IEEE, pp 103–110

A Multi-level Optimized Strategy for Imbalanced Data Classification Based on SMOTE and AdaBoost



A. Sarvani, Yalla Sowmya Reddy, Y. Madhavi Reddy, R. Vijaya, and Kampa Lavanya

Abstract Many applications require effective classification of imbalanced data, which is found everywhere. Existing classification algorithms often misclassify the minority class in imbalanced data due to the dominant class's influence. Boosting algorithms combine basic learners to improve their performance. AdaBoost, a popular ensemble learning system, can classify general datasets well. But this algorithm will be limited misclassified samples only. The minority-classified samples are not fit for this algorithm and as it alone not readies for imbalanced data classification. This paper introduced multi-level strategy to solve imbalanced data, where combined SMOTE with AdaBoost to process unbalanced data. AdaBoost and SMOTE optimize synthetic samples, implicitly modifying update weights and adjusting for skewed distributions. The typical AdaBoost technique uses too many system resources to prevent redundant or useless weak classifiers. To make process simple applied Adaptive PSO (APSO) to the SMOTE_AdaBoost results re-initialize of strategy to the optimize AdaBoost weak classifier coefficients. Four real imbalanced datasets on six classifiers—Naïve Bayes (NB), Random Forest (RF), Multi-layer Perception (MLP), Decision Tree (DT), and K-Nearest Neighbor (KNN)—verify the proposed

A. Sarvani (✉)

Department of Information Technology, Lakireddy Bali Reddy College of Engineering (Autonomous), Mylavaram, NTR Distirct, Andhra Pradesh, India
e-mail: sarvani.anandarao@gmail.com

Y. S. Reddy

Department of CSE-AIML, CVR College of Engineering, Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana, India

Y. M. Reddy

Department of S&H, CVR College of Engineering, Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana, India

R. Vijaya

Department of AI and IT, DVR & Dr HS MIC College of Technology, Kanchikacherla, NTR Distirct, Andhra Pradesh, India
e-mail: vijayar@micttech.ac.in

K. Lavanya

Department of Computer Science and Engineering, University College of Sciences, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur District, Andhra Pradesh, India

multi-level strategy. The proposed strategy (APSO_SMOTE_AdaBoost) is applied to six classifiers' and compared to SMOTE-PSO. The multi-level proposed strategy outperforms with standard approach in accuracy, precision, recall, sensitivity, and F-score.

Keywords Class imbalance · SMOTE · AdaBoost · PSO · Naïve Bayes · Random Forest · Decision Tree · K-Nearest Neighbor

1 Introduction

Imbalanced dataset categorization, which learns from skewed data distribution, is a new machine learning field of study. Two-class and multi-class datasets are imbalanced when one class has more samples than the others [1]. However, machine learning strategies applied imbalanced datasets with improper ration of minor and major classes results low accuracy [2]. Thus, learning rare but crucial instances is difficult. Learning from skew datasets is crucial for unbalanced classification tasks like fault prediction and detection, diagnosis in medicine, text classification, and many more [14, 15, 19, 20, 22, 23]. When data is uneven or has skewed class distributions, minority class predictions are poor. Many ways to rebalance data when training the model. The three main classifications namely data-level, algorithm-level, and hybrid-level class imbalance problems. Resampling data-level approaches rebalance data distribution. These strategies increase minority class or decrease majority class observations. Random oversampling, random undersampling, SMOTE, direct oversampling, and other sampling methods have merits and cons [4]. Algorithm-level techniques adapt machine learning algorithms to handle imbalanced input. Cost-sensitive algorithms can minimize cost error instead of accuracy rate by considering misclassification costs for each class. Hybrid combines earlier methods. Ensemble learning is a popular classifier that uses data-level and algorithmic-level methods to handle skewed data [3]. Ensembles aim to improve predictive accuracy over single classifiers. Generating more classifiers increases computational complexity [6]. Bagging and boosting are the most popular ensemble classifier algorithms. Bagging divides the training data into N equal-sized subsets and creates classifiers from each. Aggregating classifiers creates the classification model. The strategy of boosting algorithms is to create a single strong learning model from the combination of the various weak learning models. The boosting algorithms generated strong learning model produced better accuracy compared to the weak learner model (Fig. 1).

AdaBoost boosts unbalanced data [5]. A single-layer Decision Tree weakly classifies it. Each training iteration increases the weight of the misclassified samples and decreases the weight of the correctly classified samples, making the misclassified samples more significant. The AdaBoost method prioritizes misclassified samples above minority class samples when processing imbalanced data. It may also generate numerous redundant or worthless weak classifiers, increasing processing

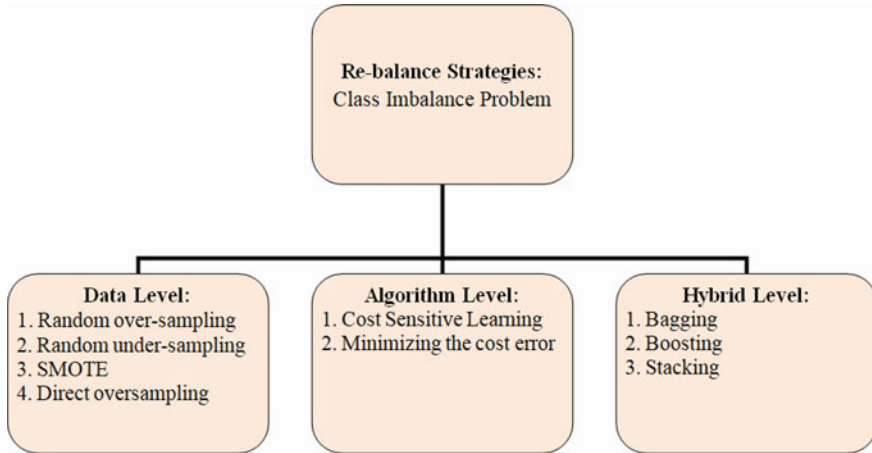


Fig. 1 Classification of re-balance strategies for class imbalance problem

overhead and reducing performance. SMOTE–AdaBoost uses the SMOTE algorithm and boosting to learn unbalanced datasets. SMOTE creates synthetic minority class instances by comparing the minority class instance to its nearest neighbors in the input domain. Synthetic cases expand minority class classification boundaries. SMOTE–AdaBoost handles discrete and continuous predictors differently. Euclidean distance for continuous predictors and value distance metric for discrete predictors locate the minority class instances nearest neighbor. SMOTE–AdaBoost uses boosting to improve minority class classification without sacrificing overall classification performance. SMOTE–AdaBoost with Adaptive PSO optimization improved method efficiency by decreasing system resources and time overhead. The multi-level strategy (i.e., APSO_SMOTE_AdaBoost) generates synthetic minority classes, trains with strong classifiers, and retains global search. Adaptive PSO relies on clustering to avoid local optimum and maintain population variety.

The paper structure: Sect. 2 covers survey related to the work. Section 3 describes about the proposed multi-level strategy for class imbalance problem. Section 4 presents comparison experiments over standard and multi-level strategy over classifiers. Section 5 Conclusion.

2 Literature Survey

Resampling data space rebalances class distribution known to be sampling strategy. The strategy in which a set of preprocessing steps which are applied to control the imbalanced data. Also, it is more convenient method compared to the standard balancing methods. The work [9] introduces a popular oversampling method known as SMOTE, and it helps to reduce the overfitting. The approach in which a new set of

minority classes are generated with the interpolation of the existing minority classes, respectively. With that the method only generates minority classes without consideration of majority classes, it leads sometimes complex in majority of cases. In highly skewed class distributions, the minority class is very sparse compared to the majority class, increasing the chance of class mixture [8–10]. Later, SMOTE versions avoided standard SMOTE issues. MSMOTE is modified SMOTE. The strategy, where the total minority classes are grouped into three categories, namely safe, border, and latent noise by the distance calculation. In the case of the safe and border, data points were chosen by the system at random from the K-Nearest Neighbors. However, it is observed that nothing can be chosen in the case of the latent noise, and moreover, it is best for mislabeled instances prevention, respectively. The work [12] in which applied SMOTE as preprocessing to balance data and then boosted the prediction using AdaBoost. However, because of SMOTE, sometimes it leads to the mislabeling and generates noise. The AdaBoost is an oversampling approach which helps to reduce the noise which is generated by the SMOTE. Thus, improving its robustness is important. After SMOTE, use PSO algorithm to optimize sample distribution, introducing minority class samples to improve Imbalance Ratio while preserving data distribution. BPSO–AdaBoost–KNN increases AdaBoost stability by extracting critical features for multi-class imbalanced data classification [21]. Gosain et al. [13] proposed a genetic algorithm-AdaBoost ensemble evolve algorithm for unbalanced data categorization. Gene evolution and fitness functions improve classifiers and optimize imbalanced data categorization.

Inspired from the related work, study attempted to build a multi-level strategy in which a group of the new synthetic samples generated with SMOTE, and then, the computing capacity of Adaptive PSO is improved with the AdaBoost, and this strategy is known as APSO–SMOTE–AdaBoost.

3 Proposed Methods

This is the section which describes multi-level strategy proposed using the three methods (AdaBoost, SMOTE, and Adaptive Particle Swarm Optimization (APSO)). To improve classification performance of individual classifiers adopted this multi-level strategy in which each one performed individual role for better contribution. The APSO uses the clustering method, to enhance the optimization principle in PSO. This multi-level strategy, APSO_SMOTE_AdaBoost, is to optimize the collection of synthetic samples by adding newly generated synthetic samples in the direction of the minority class centroid. Figure 2 displays the plan for the proposed strategy. The three stages of the proposed strategy are used to explain its working principle to improve the classification performance of the imbalanced data. The first level used SMOTE to generate synthetic minority groups. The level 2 is centered on AdaBoost's reweighting of ineffective classifiers. Finally, a robust classifier is trained on the data and optimized with APSO.

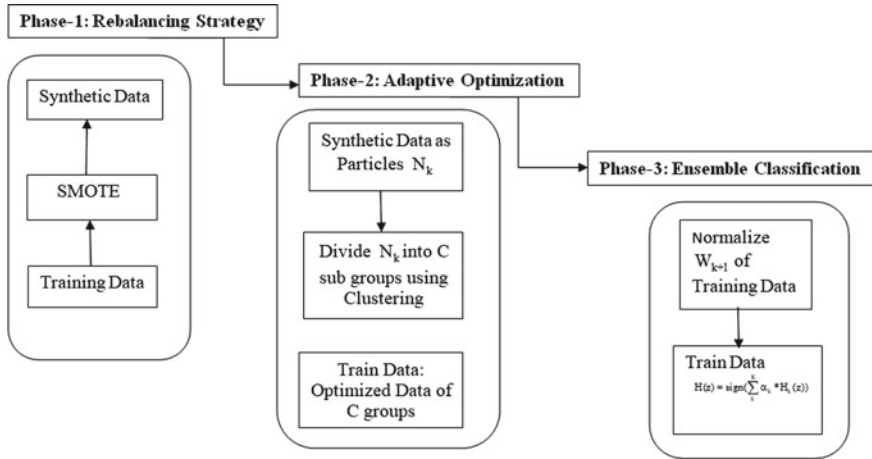


Fig. 2 Proposed ensemble method IPSO_SMOTE_AdaBoost for imbalanced data

3.1 AdaBoost

AdaBoost, as an example of a common ensemble algorithm, is able to boost classification performance by integrating numerous weak classifiers into a single robust one [10, 12]. At the outset, each sample is given the same value. The error is used to calculate the coefficients of the weak classifiers, which in turn affect the sample weights used in the iteration. Therefore, the AdaBoost algorithm can give more weight to the incorrectly labeled samples and less weight to the correctly labeled ones. The misclassified samples will receive additional attention in the following iteration of the classifier. Finally, a strong classifier is formed by linearly combining all the generated weak classifiers. Also, it is observed that the strong classifier is made with the various weights of every weak classifiers. The importance of every weak classifier weight is estimated at training phase of classification. Therefore, the algorithm prioritizes the samples or patterns that are hard to categorize. In this paper, we restrict our attention to the special case of binary classification $y = \{1, +1\}$ issues. Here are the main components of the AdaBoost algorithm:

Algorithm 1 AdaBoost Ensemble Classifier

Input:

TrainData: $T = \{(Z_i, y_i)\}$, Weak Learner Algorithm = W_L^{Alg}
 where $Z_i = \{(Z_1, y_1), (Z_1, y_1), \dots, (Z_N, y_N)\}$

Output:

The Ensemble Classifier $H_k(z)$

Step1. Assign weights to ‘N’ training samples in the data Z

$$W = (w_{11}, w_{12}, \dots, w_{1N})$$

Step2. for each weak classifier $k = 1, 2, \dots, K$

(i) Extract weak classifier based W_k

$$H_k(z) = \{-1, +1\}$$

(ii) The classification error of $H_k(z)$ on training data Z is e_k

$$e_k = P(H_k(z) \neq y_i)$$

(iii) update weight of the training samples W

$$W_{k+1} = (w_{k+1,1}, w_{k+1,2}, \dots, w_{k+1,N})$$

$$w_{k+1,i} = \frac{w_{ki}}{F_k} \exp(-\alpha_k y_i H_k(z_i))$$

where $F_k =$ normalization parameter

$$\alpha_k = \text{coefficient of } H_k(z)$$

Step3. Extract ensemble classifier $H(z)$ with linear model of base classifiers

$$H(z) = \text{sign}\left(\sum_k \alpha_k * H_k(z)\right)$$

3.2 Smote

By generating artificial samples of the minority classes, the SMOTE algorithm is able to focus on the attribute domain rather than the instance domain [13–16]. Each instance in the minority class will be oversampled by creating synthetic instances along the K-Nearest Neighbors of the minority class. The original SMOTE algorithm begins by clustering the positive samples by their Euclidean distance from the center of the cluster.

The pair of two samples (U, V) with possible values can be represented as $U = \{u_1, u_2, \dots, u_n\}$ and $V = \{v_1, v_2, \dots, v_n\}$. The Euclidean distance of pair is determined as follows in Eq. (1):

$$E(U, V) = \sqrt{\sum_{k=1}^n (u_k - v_k)}. \quad (1)$$

Based on Eq. (1), it is find that the set of k samples having very close Euclidean distances are formed together as individual groups.

After that, a set of new samples will be generated randomly among two samples of each group of k samples and is described as:

$$U_{\text{new}} = U + \text{rand}(0, 1) \times (V_k - U). \tag{2}$$

From Eq. (2), U denotes the set of samples and V_k is the k th nearest neighbor associated with the set of samples U . The standard function called $\text{rand}(0, 1)$ is used to help to generate the random number between 0 and 1. The newly generated sample is represented as U_{new} from the above-mentioned parameters, respectively. However, it is mentioned that to make dataset balance process behind in Eq. (2) done multiple times.

3.3 Adaptive PSO Algorithm

Premature convergence of PSO can be prevented if the population is sufficiently diverse [11, 17]. When used for path planning, the classical PSO is straightforward to implement and requires few tuning parameters, but it suffers from poor search ability, a propensity to converge on a single optimal local solution, a lack of particle diversity, negligent convergence, and inaccurate results. The primary goal of Adaptive Particle Swarm Optimization (APSO) is to enhance the efficiency of standard PSO through the use of clustering as the algorithm’s directing concept. To execute the adaptive partition of the population into various subgroups, the clustering approach [] is used. This technique can find the geometric center of class clusters in a dataset automatically.

If we assume a N dimensional search space, with particles N_k , then we know that each particle k must satisfy two conditions ρ_k and δ_k . One such metric ρ_k , the “particle density gradient,” is described in Eq. (3).

$$\rho_k = \sum_{m \neq k} \exp\left(-\left(\frac{d_{mk}^2}{d_c}\right)\right), \tag{3}$$

where d_c is the truncation distance, and the Euclidean distance between x_m and x_k is to be specified as d_{mk} .

The minimum and maximum distances between a particle k and other particles with a higher ρ_k are defined as follows for the second parameter δ_k :

$$\begin{aligned} \delta_k &= \min_{m: \rho_m > \rho_k} (d_{km}), \\ \delta_k &= \max_m (d_{km}). \end{aligned} \tag{4}$$

The cluster of the center is determined by particles whose values will be larger enough in case of both ρ and δ parameters. In order to extract set of particles from the population and elect as cluster center used $\gamma_k = \rho_k * \delta_k$. The division of truncation distance by the set of particles x_m is formed the center of the cluster only if satisfy the condition $((\rho_k > \rho_k(x_k)) \&\& (\gamma \rightarrow \gamma(x_m)))$.

Particles with higher ρ and δ values are so chosen to serve as center of the cluster. A particle filter is used to select for the most central members of a cluster based on $\gamma_k = \rho_k * \delta_k$. After sorting the particles x_m into smaller groups based on density ρ_k . However, it is observed that the ρ_k is higher than δ_k of x_k and very close to the γ of x_m , respectively.

$$z_k^d = w * z_k^d + c_1 * \text{rand}_1^d(\text{pb}_k^d - z_k^d) + c_2 * \text{rand}_2^d(\text{cgb}_c^d - z_k^d). \quad (5)$$

From Eq. (5), w is represented as weight, the two learning factors, namely c_1 and c_2 , worked with the two random numbers random numbers rand_1 and rand_2 with interval of $[0-1]$. The current position of the particle k is represented as pb_k^d and the best position of the particle k is represented as pb_k^d and cgb_c^d .

The upgradation of the local optimum particles is done by the consideration of knowledge of each subgroup individually and it helps to improve communication between C subgroups. This is the update formula:

$$z_k^d = w * z_k^d + c_1 * \text{rand}_1^d(\text{pb}_k^d - z_k^d) + c_2 * \text{rand}_2^d\left(\frac{1}{C} \sum_{C=1} \text{cgb}_c^d - z_k^d\right). \quad (6)$$

Ordinary particles not only pursue local optimality, but also serve as a communication medium across groups, allowing them to influence the search strategy of the population and increase its diversity. However, it is better to avoid the local optima to improve the performance of communication in subgroups. Particles' convergence could be delayed if the direction of the update was to be too uncertain due to excessive learning. The information offered by the local optimal particles is extremely helpful in locating the optimal solution because they have the highest chance of identifying it inside the subgroup. Thus, the average information is used by each subgroup to direct the local optimal particle update mentioned in Eq. (6). Particles can be kept from settling into local optimum states as the results improved population diversity. As a outcome within each sub-group, a lot of improvement is delivered toward transmission of information.

4 Results and Discussion

This part includes a description of the datasets that were utilized during the entirety of the study, as well as evaluation metrics, results, and discussion.

Table 1 Imbalanced datasets used in the study

Dataset	Attributes	No. of. samples	IR
Glass	9	214	1.82
Wisconsin	9	683	1.86
Ecoli	7	336	8.6
Yeast	8	1484	2.46

4.1 Dataset Used in the Study

In this section of the article, we will test the worth of the proposed approach by conducting experiments on four different datasets that are imbalanced. The datasets consist of glass, Wisconsin, Ecoli, and yeast, all of which were obtained from the UCI repository [18]. In Table 1, the specifics of each of the four imbalanced datasets that were utilized in the analysis are listed. In Table 1, IR stands for “Imbalance Ratio,” and the Ecoli dataset has the highest Imbalance Ratio (i.e., 8.6). The glass dataset has the lowest Imbalance Ratio (1.82), which means that it has the least amount of weight.

4.2 Evaluation Metrics

When evaluating the effectiveness of the methods for binary classification, having appropriate evaluation criteria is absolutely necessary. Standard criteria for evaluation include things like accuracy, recall, precision, and specificity, among others. This is due to the fact that the minority class can skew the decision boundary but has very little influence on the precision []. Recall, precision, specificity, and F-score are the evaluation metrics that we concentrate on. Table 2 presents the confusion matrix representation and is primary parameter for any evaluation metric.

When evaluating information retrieval, recall refers to the percentage of retrieved objects that are relevant; when discussing imbalanced classification, recall refers to the percentage of minority instances that are correctly classified. The percentage of relevant objects that are detected and located for retrieval is denoted by the term “precision.” The F-score is a harmonic mean that takes into account both recall and precision. The percentage of accurately categorized cases of the majority is denoted by the term “specificity.”

Table 2 Confusion matrix

	Condition positive	Condition negative
Test outcome positive	True positive (T _P)	False positive (F _P)
Test outcome negative	False negative (F _N)	True negative (T _N)

$$\begin{aligned} \text{Accuracy (Acc)} &= \frac{T_P + T_N}{T_P + T_N + F_P + F_N}, \\ \text{Precision (}P_r\text{)} &= \frac{T_P}{T_P + F_P}, \\ \text{Recall (}R_c\text{)} &= \frac{T_P}{T_P + F_N}, \\ \text{Sensitivity (}S_n\text{)} &= \frac{T_P}{T_P + F_N}, \\ \text{Specificity (}S_p\text{)} &= \frac{T_N}{T_N + F_P}, \\ \text{F - measure (}F_m\text{)} &= \frac{2 * P_r * R_c}{P_r + R_c}. \end{aligned}$$

4.3 *The Classification Methods Performance with SMOTE_ APSO Technique*

Adaptive PSO optimizes normal SMOTE performance on four imbalanced datasets. The proposed method's detailed classification results are provided in Figs. 3, 4, 5, and 6. The 80:20 training/test ratio of the glass dataset yielded IR of 1.82. NB, RF, LR, MLP, DT, and KNN classify training set samples using the Adaptive PSO (ISO) algorithm with SMOTE. Figure 3 shows that RF 0.8781's accuracy is dominated by other approaches. NB and DT accuracy values 0.8773 and 0.8744 equal RF performance. LR and MLP are slightly more accurate than KNN. RF precision is 0.8379, which is better than all other methods except NB precision, 0.8222. LR, MLP, DT, and KNN nominal precision values are 0.6852 lower than all other approaches. Recall, F1, and sensitivity function similarly. RF has the best recall value 0.7884. KNN neglects 0.7198. Compared to other approaches, F1 and Sensitivity DT and LR values 0.8017, 0.8620 dominate. KNN performs poorly in all approaches for F1 and sensitivity.

Wisconsin's 80:20 training/test dataset had IR of 8.6. Adaptive PSO (ISO) with SMOTE classifies training set samples using six classifiers. Figure 4 shows that LR 0.8376's accuracy is dominated by other approaches. RF accuracy 0.8257 matches LR performance. NB, MLP, and DT are marginally more accurate than KNN at 0.8193, 0.8144, and 0.8051. LR precision is 0.8178, which is better than all other methods except RF precision, 0.8065. However, the nominal precision values derived from the MLP and DT compared to the KNN are 0.7633 lower than all other methods. Recall, F1, and sensitivity function similarly. Recall has the best value 0.7924. KNN neglects 0.7019. All approaches dominate F1 and sensitivity LR. KNN performs poorly in all approaches for F1 and Sensitivity.

IR was 8.6 for the 80:20 training/test Ecoli dataset. APSO with SMOTE classifies the training set samples with six classifiers. Figure 5 shows that NB 0.8428's accuracy

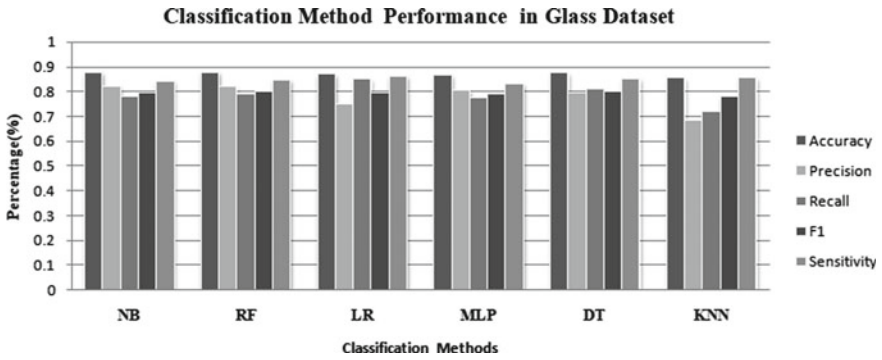


Fig. 3 Performance analysis of SMOTE–APSO in Glass dataset

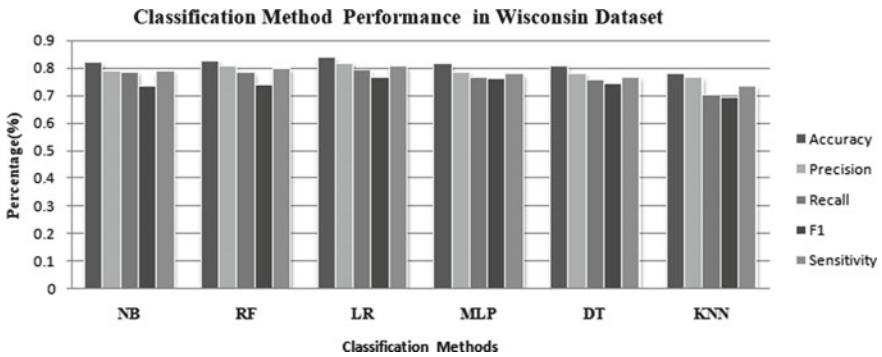


Fig. 4 Performance analysis of SMOTE–APSO in Wisconsin dataset

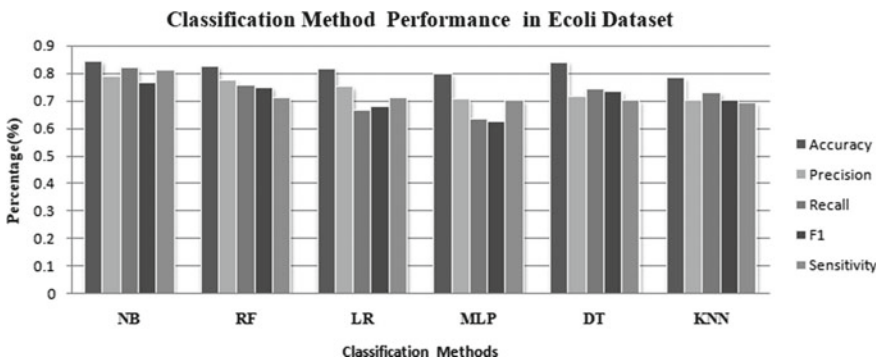


Fig. 5 Performance analysis of SMOTE–APSO in Ecoli dataset

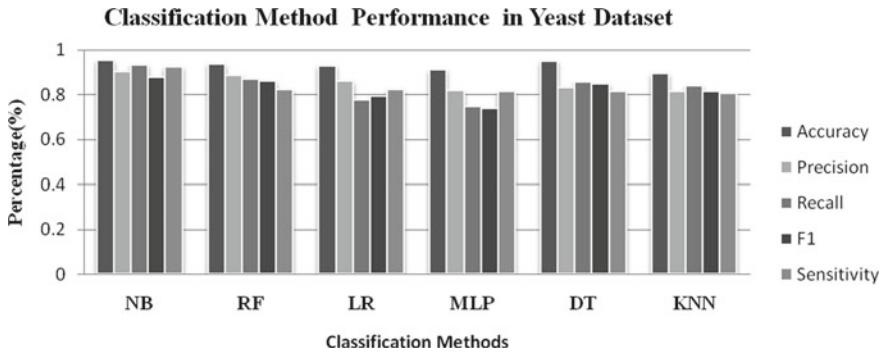


Fig. 6 Performance analysis of SMOTE–APSO in yeast dataset

is dominated by other approaches. RF and DT accuracy scores 0.8237 and 0.8404 are comparable to the NB. MLP and LR are marginally more accurate than KNN at 0.7992 and 0.8156. NB precision is 0.7895, which is better than all other methods except RF precision, 0.7743. However, the nominal precision values derived from the LR, MLP, and DT compared to the KNN are 0.7028 lower than all other methods. Recall, F1, and sensitivity function similarly. Recall gives the NB the best value 0.8129. KNN neglects 0.7269. All approaches dominate F1 and sensitivity NB. KNN performs poorly in all approaches for F1 and sensitivity.

The yeast dataset has 80:20 training and test sets and IR of 2.46. APSO with SMOTE classifies the training set samples with six classifiers. Figure 6 shows that NB 0.9543’s accuracy is dominated by other approaches. RF and DT accuracy ratings 0.9356 and 0.9520 are comparable to the NB. MLP and LR are marginally more accurate than KNN at 0.9116 and 0.9277. NB precision is 0.9021, which is better than all other methods except RF precision, 0.8872. However, the nominal precision values derived from the LR, MLP, and DT compared to the KNN are 0.8171 lower than all other methods. Recall, F1, and sensitivity function similarly. Recall gives the NB the best value 0.9345. KNN achieves 0.8161 negligence. All approaches dominate F1 and sensitivity NB. KNN performs poorly in all approaches for F1 and sensitivity.

4.4 The Performance of Classification Methods with SMOTE_APSO_AdaBoost Strategy

We test the SMOTE algorithm with APSO and AdaBoosting approach on four real-world imbalanced datasets with six classifiers NB, RF, LR, MLP, and DT. The 80:20 training/test ratio of the glass dataset yielded IR of 1.82. NB, RF, LR, MLP, DT, and KNN classify training set samples using the Adaptive PSO (ISO) algorithm with SMOTE and AdaBoost. Figure 7 shows that NB 0.9033’s accuracy is dominated by

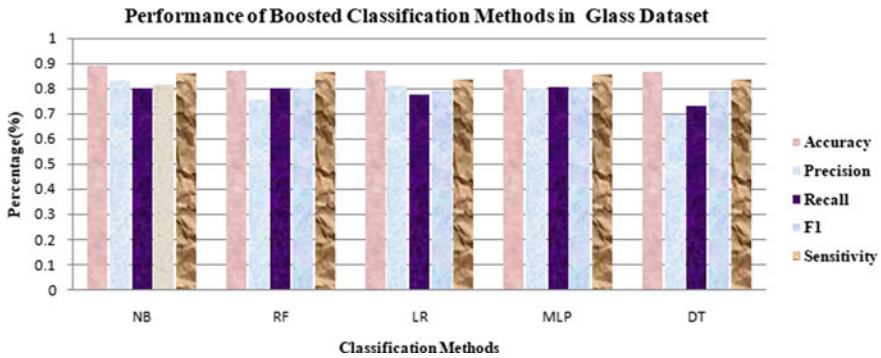


Fig. 7 Performance of classification models with SMOTE_APSO_AdaBoost in glass dataset

other approaches. RF accuracy is 0.8917 and DT accuracy is 0.8731. LR and MLP are slightly more accurate than KNN.

The NB’s precision value, 0.8493, is better than all other methods to save the RF’s 0.8327. LR, MLP, DT, and KNN nominal precision levels are 0.6983, lower than all other approaches. Recall, F1, and sensitivity function similarly. NB recalls 0.8055 better than all other approaches. KNN neglects 0.7326. F1 and sensitivity DT and LR values 0.8241, 0.8694 exceed other approaches. KNN performs poorly in all approaches for F1 and Sensitivity.

Wisconsin’s 80:20 training/test dataset had IR of 8.6. Adaptive PSO (ISO) with SMOTE classifies training set samples using six classifiers. Figure 8 shows that LR 0.8725’s accuracy is dominated by other approaches. MLP accuracy 0.8583 matches LR performance. RF, NB, and DT are marginally more accurate than KNN at 0.8404, 0.8464, and 0.8098. The LR’s precision of 0.8231 is unmatched. However, RF and MLP precision values 0.8216 and 0.8205 are closer to LR. DT nominal precision values are 0.7757 lower than KNN values. Recall, F1, and sensitivity function similarly. Recall has the best value 0.8183. KNN neglects 0.7149. All approaches dominate F1 and sensitivity LR. KNN performs poorly in all approaches for F1 and sensitivity.

When compared to the performance of standard classification models using the APSO_SMOTE_AdaBoost strategy, the bootstrapped classification models performed exceptionally well, as was the case with the other two datasets, namely Ecoli and Yeast. After implementing the proposed technique, each of the six classifiers showed an increase in all of the aforementioned parameters, namely accuracy, precision, recall, and F1-score. Consider the example of LR and RF; in comparison to MLP, DT, and KNN, these two approaches completely dominated the performance of the traditional technique. After implementing the modifications to the standard techniques using APSO and AdaBoost, the performance has gradually increased, reaching a high level in the case of MLP and reaching a nominal level in DT and KNN.

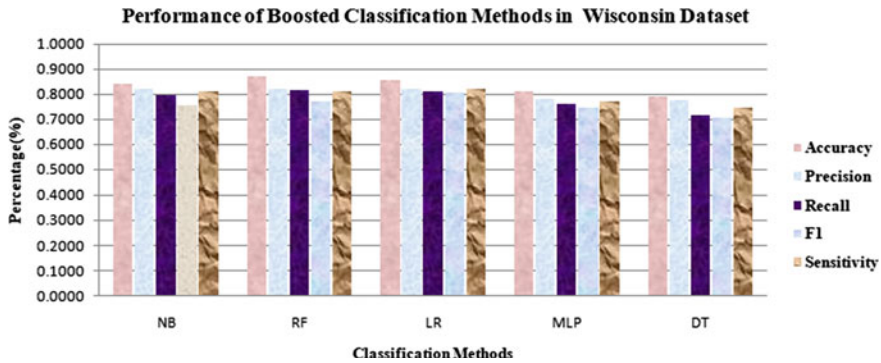


Fig. 8 Performance analysis of SMOTE–APSO in Wisconsin dataset

5 Conclusion

The traditional AdaBoost and PSO algorithm emphasizes on the samples that have been misclassified in the local space rather than the samples of the minority class and trying to maintain population diversity. In this research, a multi-level strategy was presented to handle the problem of imbalanced data. The strategy incorporated SMOTE and AdaBoost to analyze the uneven data. Both AdaBoost and SMOTE optimize synthetic samples by making implicit modifications to update weights and making adjustments for skewed distributions. In order to simplify the procedure, we applied Adaptive PSO, also known as APSO, to the SMOTE_AdaBoost findings and re-initialized our strategy in order to optimize the AdaBoost weak classifier coefficients. The suggested multi-level method is validated by four genuine unbalanced datasets using six classifiers: Naive Bayes (NB), Random Forest (RF), Multi-layer Perception (MLP), Decision Tree (DT), and K-Nearest Neighbor (KNN). A comparison is made between the strategies that was proposed (APSO_SMOTE_AdaBoost) and SMOTE-PSO. This strategy is applied to six different classifiers. The proposed multi-level technique is superior to the usual approach in terms of performance. It has also been noticed that the SMOTE algorithm successfully utilized the benefits of Boosting and PSO optimization to improve the predictive analysis of the class imbalance problem, particularly on datasets containing data from minority classes. The application of the suggested method to the field of gene analysis is the focus of our effort for the foreseeable future.

References

1. He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284
2. Chawla NV, Lazarevic A, Hall LO (2003) SMOTEBoost: improving prediction of the minority class in boosting. In: Proceedings of the 7th European conference on principles and practice of knowledge discovery in databases, Cavtat-Dubrovnik, Croatia, 22–26 Sept 2003, pp 107–109
3. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16(1):321–357
4. Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing, pp 878–887
5. Viola P, Jones M (2002) Fast and robust classification using asymmetric AdaBoost and a detector cascade. *Adv Neural Inf Process Syst* 14:1311–1318
6. Li Y, Guo H, Li Y (2016) A boosting based ensemble learning algorithm in imbalanced data classification. *Syst Eng Theor Pract* 36:189–199
7. Prachuabsubpakij W (2015) CLUS: a new hybrid sampling classification for imbalanced data. In: Proceedings of the 12th international joint conference on computer science and software engineering (JCSSE), Hat Yai, Thailand, 22–24 July 2015, pp 281–286
8. Yang X, Ma Z, Yuan S (2016) Multi-class Adaboost algorithm based on the adjusted weak classifier. *J Electron Inf Technol* 38:373–380
9. Li K, Xie P, Liu W (2017) An ensemble evolve algorithm for imbalanced data. *J Comput Theor Nanosci* 14:4624–4629. <https://doi.org/10.1166/jctn.2017.6867>
10. Guo Q-J, Li L, Li N (2008) Novel modified AdaBoost algorithm for imbalanced data classification. *Comput Eng Appl* 44:217–221
11. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60. <https://doi.org/10.1016/j.ins.2014.08.039>
12. Ren K-Q, Gao X-L, Xie B (2016) AdaBoost face detection algorithm based on fusion optimization of AFSA and PSO. *J Chin Comput Syst* 37:861–865
13. Gosain A, Sardana S (2019) Farthest SMOTE: a modified SMOTE approach. https://doi.org/10.1007/978-981-10-8055-5_28.
14. Kampa L, Yamini K, Basavaraju A, Anoop K A stack based ensemble learning method for diagnosing autism. *Math Stat Eng Appl* 71(3):237–251. ISSN 2326-9865 (SCOPUS) 7
15. Anu Priya K, Sravya E, C, Reddy GL, Sathvika J, Lavanya K (2022) Audio based sentiment prediction model. *Math Stat Eng Appl* 71(3):209–227. ISSN 2326-9865 (SCOPUS)
16. Wang KJ, Makond B, Chen KH et al (2014) A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. *Appl Soft Comput* 20:15–24
17. Hu K, Zhou Z, Weng L et al (2016) An optimization strategy for weighted extreme learning machine based on PSO. *Int J Pattern Recogn Artif Intell* 31(1):1751001
18. Ding Z (2011) Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics. Dissertation, Georgia State University
19. Lavanya K, Suresh GV (2021) An additive sparse logistic regularization method for cancer classification in microarray data. *Int Arab J Inform Technol* 18(2). <https://doi.org/10.34028/iajit/18/10>, ISSN 1683-3198, E-ISSN 2309-4524, Impact Factor is 0.654
20. Lavanya K, Syamala D, Vani KV, Gipsy C (2020) A novel SVM-KNN classifier for cervical cancer diagnosis using feature reduction and imbalanced learning techniques. *Int J Psychos Rehabil* 24(6):5151–5161. ISSN 1475-7192
21. Del Valle Y, Venayagamoorthy GK, Mohagheghi S et al (2008) Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Trans Evol Comput* 12(2):171C195
22. Lavanya K, Rambabu P, Suresh V, Bhandari R (2023) Gene expression data classification with robust sparse logistic regression using fused regularization. *Int J Ad Hoc Ubiquitous Comput (IJAHUC)* 42(4). Inderscience Publishers, 20 Apr 2023

23. Basavaraj GN, Lavanya K, Sowmya Reddy Y, Srinivasa Rao B (2022) Reliability-driven time series data analysis in multiple-level deep learning methods utilizing soft computing methods. *Meas Sens* 24:100501. ISSN 2665-9174

A Protocol for Mutual Authentication in Remote Keyless Entry Systems that Employs Random Variables



A. Nguyen Thi Thuy

Abstract The development of smart devices brings convenience to human life but also poses challenges in security replay attacks and tracking attacks by fake identities accessing assets. Therefore, the purpose of this paper is to improve previous mutual authentication protocols by proposing a mutual authentication protocol that uses random variables and encrypts frame data to address the issue of fake identity attacks. After implementing the protocol for the Remote Keyless Entry System (RKE) system, experimental results showed acceptable unlocking speed and security.

Keywords Mutual authentication protocol · Remote Keyless Entry system · Hash-based RFID mutual authentication protocol

1 Introduction

The Remote Keyless Entry (RKE) system [1, 2] is a technology that allows users to lock and unlock their vehicle using a small device such as a remote control. This technology has been widely adopted by many car manufacturers and has become a standard feature on many modern vehicles. The RKE system typically operates by using radio frequency (RF) waves to transmit a code between the smart key and the vehicle's locking system.

RKE increases convenience for users by allowing them to unlock their vehicle from a distance, saving time and making access to the vehicle easier. However, RKE transmits the access code via RF, which poses a risk of a third party intercepting the transmission and spoofing the access code to gain entry to the asset, resulting in a high risk of asset loss.

Since 2010, researchers around the world have proposed many mutual authentication protocols [3–5] for use in remotes and anti-replay attack locks on vulnerable devices. Mutual authentication is a security technique in which each participating

A. Nguyen Thi Thuy (✉)

Ho Chi Minh City University of Foreign Languages - Information Technology, Ho Chi Minh City 70000, Vietnam

e-mail: antt@hufit.edu.vn

component, typically a server and a client, verifies each other before communication can proceed, with no involvement from a third party. The principle of mutual authentication protocol does not rely on the current time of the computer or synchronized devices, but on a one-time randomly generated variable (nonce) at the time of communication, and is encoded by secure encryption algorithms that are practically feasible, such as symmetric encryption with a secret key K shared and stored securely between the two parties.

The “Scalable pseudo-random RFID private mutual authentication” protocol was introduced by a team of Chinese authors, led by Jianqing Fu. This protocol allows Reader and Tag to authenticate each other using symmetric encryption algorithms with a secret key and random variability, with state identification (IDT, K) being stored in both places. Furthermore, each tag has its unique key, $IDTA$ [6, 7]. Subsequently, in Mohammad et al.’s paper titled “Cryptanalysis of two mutual authentication protocols for low-cost RFID” [8], the authors analyzed the FWCFP protocol and found many security weaknesses introducing four types of attacks on the protocol based on asynchrony and non-repudiation, one of which is the ability for a hacker to replay a command frame in the second round.

To improve security weaknesses, the proposed protocol has many similarities in operation and use of variables with FWCFP but it differs in that it does not use synchronized variables but uses “session signatures” (see Fig. 1). Each time data is transmitted, and each party generates a random variable and signs it on top of the previously received message of the other party. Then, the algorithm for the protocol is optimized to apply it to both remote devices (see Fig. 2) and lock devices (see Fig. 3) of the “Remote Keyless Entry system.” As a result, RKE has an acceptable real-time running time of 199 ms (see Fig. 4).

The contributions of the paper are as follows:

- Proposing a mutual authentication protocol that can prevent attacks such as frame replay attacks and traceability attacks on RF communication.
- Optimizing the algorithm to apply the protocol on resource-constrained devices.

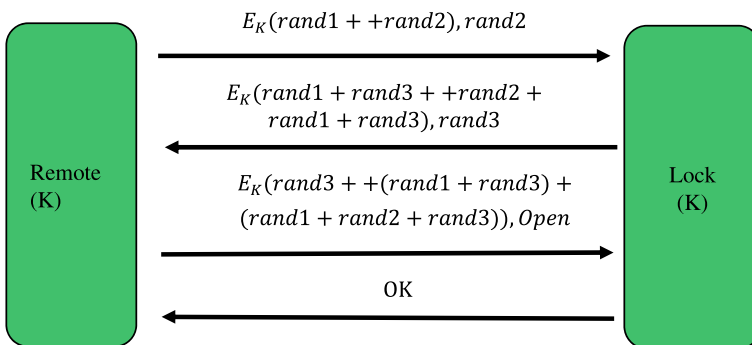


Fig. 1 Mutual authentication protocol using random variables

Fig. 2 Remote the device**Fig. 3** Lock the device

The remaining sections of the paper are structured as follows: Sect. 2 provides an overview of the related work literature. Section 3 outlines the proposed method. Section 4 illustrates the outcomes of the conducted experiments. Lastly, Sect. 5 presents the conclusion of the paper.

2 Related Work

As of our current knowledge, which is up to April 2023, some recent related work on RKE systems includes the following: In [9], the paper proposes a new ultrasonic distance-bounding protocol to secure RKE systems against relay attacks. The proposed protocol uses sound waves to measure the distance between the key and the car and ensures that the key is in close proximity to the car before allowing access. The authors in [10] propose a new scheme to enhance the security and privacy of Passive Keyless Entry and Start (PKES) systems for modern vehicles. The proposed scheme uses a combination of encryption, authentication, and location-based services to secure PKES systems against relay attacks and other types of attacks. The work [11] presents a security analysis of Bluetooth Low Energy (BLE) keyless entry systems and identifies several vulnerabilities that could allow an attacker to bypass the authentication process and gain unauthorized access to the car. The paper also proposes some countermeasures to mitigate these vulnerabilities. Li et al. [12] provide an overview of the security challenges and vulnerabilities associated with PKES systems. It also proposes a set of countermeasures to enhance the security of PKES systems, including

the use of encrypted key exchange and the deployment of physical security measures. In [13], Kumar et al. introduce a comprehensive survey of relay attacks in automotive keyless entry systems. It reviews various types of relay attacks and proposes a set of countermeasures to mitigate these attacks, including the use of distance-bounding protocols and signal strength analysis. And the paper of Lee et al. [14] proposes a new keyless entry system that uses machine learning-based behavioral authentication to verify the user's identity. The proposed system analyzes the user's behavioral patterns, such as their typing speed and swipe direction, to authenticate the user and grant access to the car.

3 Proposed Method

3.1 Mutual Authentication Protocol Using Random Variables

To overcome the weakness of the previously proposed "Scalable pseudo-random RFID private mutual authentication" protocol that used synchronized variables, which created a vulnerability for attackers to disrupt synchronization and prevent the Reader and Tag components from continuing the transaction process, this paper proposes not to use synchronized variables. Instead, only random variables are used in a transaction session, and each transaction session is independent of the previous session.

Figure 1 presents mutual authentication protocol using random variables. In which: E is the AES symmetric encryption function, and K is the shared secret key between Lock and Remote with a length of 128 bit, 256 bit, and 512 bit. Rand1 , rand2 , and rand3 are random numbers generated for each transaction, "Open" is the unlock command, and "OK" is the successful response command.

The operation of the protocol is as follows

- Step (1) Remote:
 - It generates two random variables rand1 , rand2 and combines them, then encrypts them using the function $\text{Encrypt}(\text{rand1}, \text{rand2}, C12, K)$, input: rand1 , rand2 , output: $C12$.
 - It sends the data consisting of $(C12, \text{rand2})$ to the Lock.
- Step (2) Lock:
 - It receives a data frame consisting of $(C12, \text{rand2})$ from the Remote, then decrypts $C12$ and extracts 2 numbers $\text{rand1}'$ and $\text{rand2}'$ by function $\text{Decrypt}(\text{rand1}', \text{rand2}', C12, K)$.
 - It checks whether $\text{rand2}' = \text{rand2}$ or not. If it is correct, then the Lock authenticates the Remote, then Lock generates 1 random variable rand3 , then it combines and encrypts by the function $\text{Encrypt}(\text{rand1} + \text{rand3}, \text{rand1} + \text{rand2} + \text{rand3}, C3, K)$ with output is $C3$.

- Lock sends the data consisting of (C3, rand3) to the Remote.
- Step (3) Remote:
 - It receives a data frame consisting of (C3, rand3) from the Lock, then decrypts C12 and extracts 2 numbers r13, r123 by function DeCrypt(r13, r123, C3, K).
 - It checks whether $r13 = rand1 + rand3$ and $r123 = rand1 + rand2 + rand3$ or not.
 - If it is correct, then the Remote combines rand3 with $r13 + r123$ and encrypts by the function EnCrypt(rand3, $r13 + r123$, C5, K) with output C5.
 - Remote sends the data consisting of (C3, rand3) to the Lock.
- Step (4) Lock:
 - It receives a data frame consisting of (C5, rand2) from the Remote, then decrypts C5 and extracts 2 numbers rand3' and r' by the function DeCrypt(rand3', r', C5, K).
 - It checks whether $rand3' = rand3$ and $r' = r13 + r123$ or not. If it is correct, then the Lock will execute the “Open” command to unlock, then it sends the command code “OK” to the Remote.

Security factors of the protocol

The algorithm of the proposed protocol is based on the principle of stacking the signatures of participating parties onto the transmitted message, making it highly secure. Even if an attacker intercepts the transmitted frame, it would be difficult to decipher. Moreover, the “session signatures” rand2 and rand3 only hold value for a single session and cannot be predicted or reused in subsequent sessions, not even known to the manufacturer or follow any rule, which means an attacker cannot use them for future attacks or predict them. An attacker can only succeed if they can obtain a set of three numbers (rand1, rand2, rand3) that they have collected before.

But the probability of the triplet (rand1, rand2, rand3) repeating is very low, specifically.

Assuming each number has n bits, the triplet has a total of $3 * n$ bits. Therefore, there are a total of possible unique triplets (rand1, rand2, rand3) that can be generated.

The process of generating a random triplet (rand1, rand2, rand3) involves selecting a number from the set of possible numbers. So the probability of selecting 2 identical sets is $*$.

The number of possible sets is 2^{3*n} , so the probability of having two identical sets is $2^{3*n} / (2^{3*n} * 2^{3*n}) = 1/2^{3*n}$. For example, if the length of each number rand1, rand2, rand3 is $n = 128$, then the probability of having a repeated (rand1, rand2, rand3) set is $1/2^{3*128} = 2^{-384}$, which is very small. Therefore, the protocol addresses the weaknesses of synchronous attacks and trace attacks.

Tracking attack script

The attacker has collected data frames from previous communications between the Remote and Lock. Assuming that the attacker is impersonating the Remote:

- Step (1) The attacker sends a data frame that was previously sent by the real Remote to the Lock.
- Step (2) The Lock receives the data frame (C12, rand2), decrypts it, and verifies that $\text{EnCrypt}(C12) = \text{rand1} + \text{rand2}$. Then, it sends (C3t, rand3t) to the attacker.
- Step (3) At this moment, the attacker receives (C3t, rand3t) from Lock, but it does not have the key K to correctly encrypt $\text{rand3t} + r13t + r123t \rightarrow C5t$. Therefore, it will send an estimated value of C5' and the "Open" unlock command to Lock to perform the unlock command.
- Step (4) At this moment, Lock receives (C5', Open). Lock decrypts C5' and compares it with the sum of $\text{rand3t} + r13t + r123t$ previously stored and finds it incorrect, so the protocol ends here. The Lock does not perform the unlock command of the attacker.

3.2 Application of Mutual Authentication Protocol Using Random Variables to Design Remote Keyless

Entry System

RKE consists of weak processors and uses RF waves to transmit data, which limits the amount of data that can be transmitted in a single frame. Each data frame transmission only contains 32 bytes, so each random number rand1, rand2, and rand3 can only be up to 8 bytes. Two 8-byte numbers are combined into one 16-byte number after being encrypted using the AES algorithm, resulting in a 16-byte code table being sent.

Next is the task of selecting the key K for the AES algorithm. To install and run on weak devices such as controllers, the selected key K has a limited length of 128 bits. Instead of using the key schedule algorithm for each AES encryption round, we implemented a key generation algorithm executed on the computer. From the main key of 16 bytes, it generates an expanded key table of 176 bytes and directly loads the expanded key table (round key) into the ROM memory of both Lock and Remote devices to increase speed.

Modules in the system:

Module KeySchedule: The input K0 is a 16-byte array and the output RoundKey is a 176-byte array. This module calculates the expanded key array from the initial key K0 on a computer, and then the computer connects to the Lock and Remote devices to write it into their memory for use in AES encryption and decryption. This process is represented by the following Algorithm 1.

Algorithm 1

```

KeySchedule(K0)
Begin
    Parse(K0,W0,W1,W2,W3);
    // Each word W is 4 bytes of the
    RoundKey array.
    For( tur Wi = 4 to 43)
        Begin
            If (i mod 4 = 0)
                KeySchedule(K0)
            Else
                Wi= Wi-4 Xor Wi Xor Rcon();
        End
    End
End
    
```

Module EnCrypt(rand1, rand2, State): The input of this function is 2 numbers of 8 bytes rand1, rand2, and the pre-stored RoundKey. The output is a 16-byte state which is the cipher of rand1 concatenated with rand2.

Module DeCrypt(rand1, rand2, State): The input of this function is the 16-byte state which is the cipher of rand1 concatenated with rand2 and the pre-stored RoundKey. The output is 2 numbers of 8 bytes, which are the plaintext of rand1 and rand2.

This process is represented by the following Algorithm 2.

Algorithm 2

```

EnCrypt(rand1,rand2,State)
Begin
    State= join(rand1,rand2);
    AddRoundKey(State,RoundKey,0);
    For( round=1 to 9)
        Begin
            SubBytes(State);
            ShiftRows(State);
            MixColumns(State);
            AddRoundKey(State,
RoundKey, round);
        End
        SubBytes(State);
        ShiftRows(State);
        AddRoundKey(State,Round
Key,10);
    End
End
    
```

```

DeCrypt(rand1,rand2,State)
Begin
    AddRoundKey(State,Round Key,10);
    For( round=9 downto 1)
        Begin
            InvShiftRows(State);
            InvSubBytes(State);
            AddRoundKey(State,
Round Key, round);
            InvMixColumns(State);
        End
        InvShiftRows(State);
        InvSubBytes(State);
        AddRoundKey(State,Round-
Key,10);
    End
    divide(rand1,rand2, State);
End
    
```

Module Main(): This function is the main function in protocol of 2 devices Remote and Lock and is presented by the following Algorithm 3.

Algorithm 3

```

RemoteMain()
Begin
  CreateRandom(rand1);
  CreateRandom (rand2);
  Encrypt(K);
  Send(frame);
  WaitingReceive();
  ReceiveData();
  CheckData();
  if(check true)
    Send(frame,Open);
  Else Break;
End.

LockMain()
Begin
  WaitingReceive();
  ReceiveData();
  CheckData();
  If(check true)
  Begin
    CreateRandom(rand3);
    Encrypt(K);
    Send(frame);
  End
  Else Break;
  WaitingReceive();
  ReceiveData();
  CheckData();
  If(check true)
  Begin
    Open Lock;
    Send(OK);
  End
  Else Break;
End.

```

4 Experimental Result

4.1 Experimental Process

The experimental process goes through the following steps.

- (1) We install software IAR 1.4.2.2 for the code program.
- (2) Code functions by C language for both the Remote and Lock devices.
- (3) Compiled into machine code files, "remote.hex" and "lock.hex," and use a programming kit or circuit to load the program code into the device.
- (4) Supply power to both Remote and Lock devices.

- (5) Press the open button on the Remote. When the Lock has been authenticated and opened, the word “open” will appear on the Lock device.
- (6) Connect the Remote device to the computer and collect data exchanged in the protocol.
- (7) Perform the process 2200 times to measure the average unlocking time is 197 ms.

Two devices Remote and Lock of RKE present by Figs. 2 and 3.
Experimental environment and parameters:

- The program for the devices is coded on the IAR 1.4.2.2 software environment, using the C language.
- The FTM8S003K3 ARM microcontroller is used as the central processing unit (CPU) for both the Remote and Lock boards. The FTM8S003K3 CPU has 8 KB of flash memory, 1 KB of RAM, and 128 bytes of EEPROM.
- A 2.4 GHz RF transmitter and receiver board is integrated into both the Remote and Lock boards to transmit and receive data.

4.2 Results of Data Collected During the Experiment

After performing 2200 consecutive lock-opening attempts on the Remote device, we collected the communication data on my computer, as shown in the experimental data appendix below. From this data, the average time taken to execute a lock-opening command was calculated to be 197 ms.

Description of the data obtained in the unlocking protocol: Two unlocking sessions are quoted as follows:

```
9/4/2017 12:39:05 PM:630:  $E_K(\text{rand1} \oplus \text{rand2}) \text{rand2}$   
05,17,c5,50,00,87,02,66,24,77,1d,a7,9a,5a,de,30,8d,f4,88,d9,e9,  
3a,fb,33,58,08,ca,41,f5,60,e1,aa, (Remote)rand3  
05,17,c5,50,00,87,02,92,7a,46,35,e3,eb,ac,28,1d,0d,2e,bc,2f,6d,  
4d,d4,a0,a2,8a,e8,16,b8,ba,5c,aa, (Lock)  
05,17,c5,50,00,87,a0,07,0c,c7,53,de,2c,8f,8c,be,b7,42,c4,05,cc,  
f0,e6,a0,a2,8a,e8,16,b8,ba,5c,aa,00, (Remote)  
05,17,c5,50,00,87,a0,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,  
aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,  
(Lock)  
9/4/2017 12:39:05 pm:815:  
The total execution time of the first session is 185 ms.  
9/4/2017 12:39:06 pm:951:
```

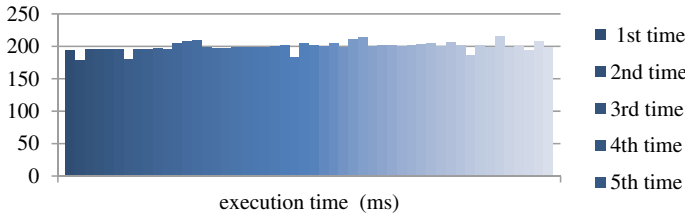


Fig. 4 Control execution time chart of door opening command

05,17,c5,50,00,87,02,56,31,56,64,9c,56,eb,ef,df,70,57,d1,5c,1b,
 0e,d1,4f,5e,1a,ce,6c,0a,6c,1e,aa,
 05,17,c5,50,00,87,02,75,55,8c,6c,f9,59,98,ef,b4,9a,8a,22,51,16,
 2f,46,70,33,8f,5c,e1,14,d7,0a,aa,
 05,17,c5,50,00,87,a0,78,da,48,2f,c6,c3,0d,0f,09,13,fb,59,a3,24,
 2a, 2a,70,33,8f,5c,e1,14,d7,0a,aa,00,
 05,17,c5,50,00,87,a0,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,
 9/4/2017 12:39:07 PM:141: aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa,aa.

The total execution time of the second session is 190 ms.

Figure 4 is a representation of the unlock execution time for 50 sampling instances obtained from the experiment.

The result shows that the average time for one unlocking operation is 199 ms.

4.3 Key Update

In practical use, issues may arise such as the Remote being damaged, lost, or stolen for duplication. In this case, the system will have the function of providing a new Remote and updating the shared Key for the entire set of Lock and Remote by setting up a new randomly generated Key and recalculating the expanded key table from the computer application and storing it in the memory of both Lock and Remote.

5 Conclusion

The paper has investigated current communication and system access protocols, comparing their operation, efficiency, and security. The paper has also researched and implemented various encryption methods for weak devices, comparing and selecting the appropriate encryption method for the practical conditions of the weak

device, such as the Remote. The paper has proposed a mutual authentication protocol that ensures both convenience and security during system access. The paper has successfully applied the mutual authentication protocol using random variables to produce a wireless unlocking system with a Remote that guarantees both security and acceptable response times in practical use.

References

1. Seto I et al (2022) Sub-GHz two-way ranging based on phase detection for remote keyless entry systems. *IEEE Trans Veh Technol* 71(9):9705–9720. <https://doi.org/10.1109/TVT.2022.3181623>
2. Parameswarath RP, Sikdar B (2022) An authentication mechanism for remote keyless entry systems in cars to prevent replay and roll jam attacks. In: 2022 IEEE intelligent vehicles symposium (IV), Aachen, Germany, pp 1725–1730. <https://doi.org/10.1109/IV51971.2022.9827256>
3. Ryu J et al (2022) Secure ECC-based three-factor mutual authentication protocol for telecare medical information system. *IEEE Access* 10:11511–11526. <https://doi.org/10.1109/ACCESS.2022.3145959>
4. Yadav AK, Misra M, Pandey PK, Liyanage M (2023) An EAP-based mutual authentication protocol for WLAN-connected IoT devices. *IEEE Trans Industr Inf* 19(2):1343–1355. <https://doi.org/10.1109/TII.2022.3194956>
5. Manikandan N, Muthaiah R, Yuvaraja T, Ramya K, Amruth RT (2022) Renovated XTEA encoder architecture-based lightweight mutual authentication protocol for RFID and green wireless sensor network applications. *Wirel Commun Mob Comput* 2022:12. <https://doi.org/10.1155/2022/8876096>
6. Fu J, Wu C, Chen X, Fan R, Ping L (2010) Scalable pseudo random RFID private mutual authentication. In: 2nd IEEE international conference on computer engineering and technology (IC CET), vol 7. China, pp 497–500
7. Daemon J, Rijmen V (1999) AES proposal: the Rijndael block cipher. *Proton World Int*
8. Habibi MH, Gardesh M, Alaghand M (2011) Cryptanalysis of two mutual authentication protocols for low-cost RFID
9. Malik H, Malik AQ, Ahmad F, Al-Fuqaha A (2022) An ultrasonic distance bounding protocol for secure keyless entry systems. *IEEE Internet Things J* 9(4):3209–3220. <https://doi.org/10.1109/JIOT.2021.3115515>
10. Zhang W, Liu X, Liu J, Chen L (2021) Enhancing security and privacy of passive keyless entry and start systems for modern vehicles. *IEEE Trans Veh Technol* 70(1):551–565. <https://doi.org/10.1109/TVT.2020.3034073>
11. Chen T, Qiu Y, Huang X, Zhou J (2022) Security analysis of Bluetooth low energy keyless entry systems. *IEEE Access* 10:36832–36844. <https://doi.org/10.1109/ACCESS.2022.3084344>
12. Li S, Zheng Z, Wang Q, Cheng X, Liu JK (2022) On the security of passive keyless entry and start systems: vulnerabilities, attacks, and countermeasures. *ACM Trans Cyber-Phys Syst* 6(1):1–25. <https://doi.org/10.1145/3516121>
13. Kumar R, Singh P, Gupta BB (2021) A comprehensive survey of relay attacks in automotive keyless entry systems. *IEEE Access* 9:51038–51058. <https://doi.org/10.1109/ACCESS.2021.3067998>
14. Lee Y, Kim J, Jeong D, Park Y (2022) Secure and efficient keyless entry system using machine learning-based behavioral authentication. *IEEE Trans Inf Forensics Secur* 17:2134–2148. <https://doi.org/10.1109/TIFS.2022.3088448>

Approximated Sparsity Regularization Factor for Monaural Speech Separation



Garima Chandel , P. P. Muhammed Shanir , Yash Vardhan Varshney ,
and Setu Garg 

Abstract Monaural speech separation problem can be solved by determining the basis vectors of the target speakers' speech. Sparse non-negative matrix factorization (SNMF) is one of the trending techniques for determining basis vectors and corresponding weights of any speech signals with constraint of controlled sparseness. Control of sparseness in matrix factorization is applied using sparseness regularization factor that cannot be the same as all speech signals. The precise sparseness regularization factor during SNMF is obtained by optimization, which is performed using different techniques in this work. The proposed work is tested for the separation of speech-speech and music-speech mixed signals. The present method has been compared with the other existing methods based on fixed sparseness regularization factor and has showed better performance with up to 13.96% improvement in terms of perceptual evaluation of speech quality and up to 0.94 dB improvement of signal to distortion ratio for case of opposite gender's mixed speech separation.

Keywords Monaural speech separation · Non-negative matrix factorization · Sparseness optimization

G. Chandel (✉)

Department of Electronics and Communication Engineering, Chandigarh University, Chandigarh, India

e-mail: chandelgarima5@gmail.com

P. P. Muhammed Shanir

Department of Electrical and Electronics Engineering, TKM College of Engineering, Kollam, India

Y. V. Varshney

Psychophysiology Lab, IIT Mumbai, Mumbai, India

S. Garg

Department of Electronics and Communication Engineering, I.T.S. Engineering College, Greater Noida, India

1 Introduction

Motivation: Throughout the last decades, human–computer interaction has increased in routine life. The most common and easy way for a human being to instruct the machine is to use the speech signal. Due to this, enormous interest in studying the single-channel problem mixed speech separation is observed. The problem is chosen due to the lower intelligence of the machine in the understanding of target speech signals in the presence of an unwanted signal. The unwanted signal may be music, noise, or another speech signal.

There are different approaches adopted by researchers to solve the problem of source separation. Some of the famous methods include microphone array processing, principal component analysis (PCA) [1], independent component analysis (ICA) [2], computational auditory scene analysis (CASA) [3], classical denoising and enhancement [4], and non-negative matrix factorization (NMF) [5].

Among all the approaches, NMF is a more common and fast-growing technique in source separation. For example, the microphone array processing technique can be applied only for the speech collected by more than one microphone. While ICA seeks directions that are largely independent of one another, PCA seeks paths that best describe data in a maximum variance sense. Like ICA, the NMF also assumes the data samples and dimensions to be independently distributed. But NMF holds non-negativity of gene expression data, and NMF can derive features more than the number of samples. That results in better signal representation.

CASA typically uses a small number of grouping principles, operating independently, and has predominantly focused on factors related to periodicities in voiced portions of the speech signal. So, it can be used in very specific conditions only. At the same time, classical denoising and enhancement techniques like spectral subtraction have the presence of processing distortions caused by the random variations of the noise. These do not utilize the statistics and the distributions of the signal, which results in inferior performance.

In recent times, the deep neural network-based approaches have also given a powerful solution for supervised audio source separation problems where a speech-and-noise mixture is present [6]. However, these methods work well when many training samples are available. The NMF is useful for audio source separation tasks, mainly where there is a limited supply of training data.

Even though NMF favors a sparse and component-based representation of non-negative data, this behavior is not guaranteed. Several writers put forth sparse NMF (SNMF) techniques that guarantee sparsity by limiting the factor matrix's l^1 -norm [7, 8]. However, the decision on the exact value of regularization factor (λ) is not directly possible for all kinds of speech signals due to the variable sparseness in speech representation.

Main contributions: This work focuses on the change of λ for each speech signal. This helps in controlling the sparse behavior of the factors done by SNMF. In the previous work [9], λ was optimized in each iteration by particle swarm optimization (PSO). As some more efficient optimization techniques (OT) have developed in recent

times, some of the highly rated optimization techniques are used for experimental purposes. The quality of separation is analyzed in terms of signal to distortion ratio (SDR), short-term objective intelligibility measure (STOI), perceptual evaluation of speech quality (PESQ), signal to interference ratio (SIR) [10–13].

Manuscript organization: Section 2 describes the basic concept of single-channel mixed speech separation using SNMF. Section 3 explains about proposed method for optimization of the sparseness regularization factor in SNMF depending on the source to be separated. Different sparseness optimizations used in the work are explained in Sect. 4. Section 5 is describing the SNMF fitness with optimized λ . Section 6 discusses the database used in the experiment. The experimental results are shown in Sect. 7 which is followed by the conclusion.

2 Single-Channel Source Separation Using SNMF

Taking a single mixed signal is necessary for single-channel source separation and separating it into its respective sources. Assume a mixture signal $X \in R^{N_t}$ is a finite, discrete-time signal that is created via,

$$X = \sum_{i=1}^{N_s} X_i, \tag{1}$$

where it is further defined as X_1, X_2, \dots, X_{N_s} be individual sound sources that have been previously mixed, N_t is the total number of samples taken for each signal, and T denotes the sampling time. When there is just one accessible channel for recording and several speakers are available, the problem is to recognize or isolate the speech of an individual speaker in a mixture of speech.

In [14–17], NMF is applied for mixed speech separation purpose. For such task, spectrogram magnitude of a mixed signal (V) is factorized in to its basis vector matrix and their corresponding weights, such that $V = [W][H]$, where $V = [W][H]$, where $W = [w_1, w_2, w_3 \dots w_N] \in R^{f \times N}$ and $H = [H_1^T, H_2^T, H_3^T \dots H_N^T]^T \in R^{N \times t}$. In SNMF, the sparseness regularization factor λ controls the sparsity of weight matrix H that associated with the minimization of the cost function and is formulated as shown in Eq. 2.

$$\cos t = \min_{W, H} \left[D(V||WH) + \lambda \sum_{i,j} H(i, j) \right]; \quad W, H > 0. \tag{2}$$

Methods used to optimize this cost function are taken into account when calculating a maximum posterior (MAP) estimate given a Gaussian likelihood function

and exponential prior distribution over H . W and H can be updated by the update rule obtained from the gradient descent technique is stated in Eqs. (3) and (4).

$$H_{i,j} \leftarrow H_{i,j} \frac{V_i^T \bar{W}_j}{[WH]_j^T \bar{W}_j + \lambda}, \tag{3}$$

$$W_j \leftarrow W_j \frac{\sum_i H_{i,j} [V_j + ([WH]_i^T \bar{W}_j) \bar{W}_j]}{\sum_i H_{i,j} [[WH]_i + (V_i^T \bar{W}_j) \bar{W}_j]}, \tag{4}$$

where W_j is the j th vector of basis matrix W , V_i is the i th row of the spectrogram magnitude matrix V , and \bar{W}_j is the normalized column basis vectors.

Here, it is clearly shown that the basis vectors and their weights are affected by the λ . λ controls the degree of sparsity. This work is focused on the changing the value of λ for individual speech signals.

3 Proposed Work

Usually, a fixed value of λ is used to control the sparseness of SNMF factors that may vary for speech-to-speech case. For example, a music signal has basis vectors with a low degree of sparseness, while a factory noise has a high degree of sparseness. To develop a universal system for signal factorization, the variable sparseness regularization factor-based SNMF is proposed. Figure 1 shows that optimized sparseness regularization factor will be calculated by minimizing the cost function of SNMF using the training signals. It will give the basis vectors for the individual signals as well, which will cascade to produce basis vectors of mixed speech. The basis vectors calculated by training signals will be used to calculate the weight matrices of the mixed speech signals and provide individual separated signals. This work used multiple optimization techniques (OT) to find optimized the sparseness regularization factor. The applied OTs are discussed in the following section.

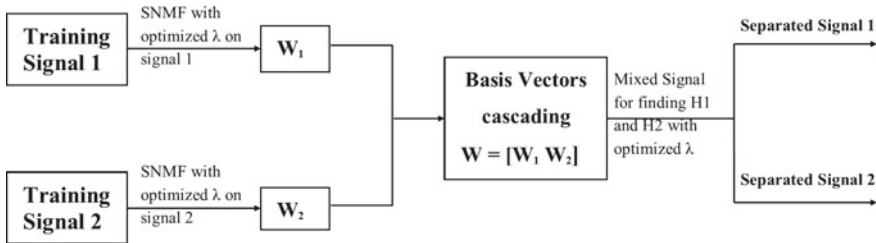


Fig. 1 Proposed framework for SNMF with optimized λ for individual speech signals

4 Sparseness Optimization

Mostly, metaheuristic algorithms are nature inspired. They have superior search efficiency, and they continuously introduce and improve the robustness to solve various complex, nonlinear optimization problems. The nonlinear optimization problems are usually solved by gradient-based optimization methods. The introduction of the genetic algorithm marked the beginning of the development of nature-based algorithms. Its foundation was the idea of biological systems given by Darwin [18, 19]. Since then, some popular techniques are evolved to find global extremes. For example, Ant Colony Optimization (ACO) [20], the most well-known global optimization algorithms that draw inspiration from nature are Simulated Annealing (SA) [21] and Particle Swarm Optimization (PSO) [22].

The search approach of deterministic and stochastic methods is entirely different. Whereas deterministic approaches solve an optimization problem in a predetermined way, stochastic methods are based on randomly sampled search points. The nature-inspired techniques are not an exception; they also start with random search points. This results in a different optimized point in different runs. However, a significantly large number of objective functions are required by traditional nature-inspired optimization methods like GA and SA. The evaluation of such problems comes with a very long computation time. The problem makes the use of such OTs impractical and infeasible.

Numerous recently released naturalistic optimization techniques are tested by Saad et al. in [23]. They compared the performance of six GO methods that are inspired by nature: Artificial Bee Colony (ABC), Firefly Algorithm (FFA), Cuckoo Search (CS), Bat Algorithm (BA), Flower Pollination Algorithm (FPA), and Grey Wolf Optimizer (GWO). These algorithms launched with enhanced capability to handle challenging, high-dimensional global optimization issues. However, the results show better performance of GWO and ABC among the others. In [24, 25], five evolutionary-based optimization algorithms are compared, including a Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), genetic algorithm (GA), memetic algorithm (MA), and shuffled frog leap (SFL) algorithm. They have shown the better performance of PSO and ACO as compared to others. So, for experimentation and comparison purposes, λ in SNMF is optimized using PSO, ACO, ABC, and GWO and compared with the fixed value of λ for all speech signals used in the experiment.

4.1 Particle Swarm Optimization

PSO examines how a swarm of insects, or a flock of birds behaves. Here, each insect or bird can be considered as an individual particle. All particles are responsible for choosing the best path for the swarm [22]. All particles are initially taken as random position (x_i) and best position from all particles obtained that is termed

as particle best (P_{best}). All particles change their positions with some velocity v_i decided based on their previous position and the particle's best position. A best new position $P_{best}(new)$ is obtained in this iteration and compared with the previous $P_{best}(previous)$. The best among all P_{best} is termed as global best G_{best} . The position of all particles keeps changing until the cost function is minimized/maximized, or there are no more iterations possible.

4.2 Ant Colony Optimization

In an ACO initially, all ants move in a random direction with equal pheromone level [20]. These ants release the pheromone during the search for food which has the property of evaporation with time. So, if an ant gets food with a shorter path, pheromone density will be high compared to the longer path. So, ants can discover the shortest path via high-density pheromone trails. If N number of ants are considered, the probability of selecting k th ant's path to reach from its position (x) to target (y) is P_{xy}^k .

P_{xy}^k depends on two factors, one is the attraction coefficient η_{xy} and the other factor is the pheromone coefficient (τ_{xy}) as shown in Eq. 5.

$$P_{xy} = \frac{\tau_{xy}^\alpha \eta_{xy}^\beta}{\sum_{\text{all allowed } x} \tau_{xy}^\alpha \eta_{xy}^\beta}, \tag{5}$$

where

$$\tau_{xy} = (1 - \rho)\tau_{xy} + \Delta \tau_{xy}^k. \tag{6}$$

ρ is the vaporization coefficient and $\Delta \tau_{xy}^k = \frac{Q}{L_k}$; here, L_k is the distance traveled by k th ant, and Q is the constant, while η_{xy} depends on the distance between source to target by a particular path.

4.3 Artificial Bee Colony

ABC algorithm depends on the foraging behavior of a swarm of bees [26, 27]. There are three types of bees in the swarm: employed, onlookers, and scouts. Employed bees search for the food source and pass the information to onlooker bees by their unique waggle dance. Onlooker bees decide the quality of food source based on their dance and evaluate the nectar amount in the source. Abandoned food sources are determined by onlookers and are replaced with the new food sources discovered by scouts. Onlookers store the best food found till then.

The probability of selecting food sources by i th bee among N bees is given by Eq. 7

$$p_i = \frac{\text{fit}_i}{\sum_{j=1}^N \text{fit}_j}. \tag{7}$$

For the numerical optimization minimization issues, fit_i is calculated by Eq. 8.

$$\text{fit}_i = \begin{cases} \frac{1}{1+f_i} & \text{if } f_i \geq 0 \\ 1 + |f_i| & \text{otherwise} \end{cases}, \tag{8}$$

where f_i is the objective function value for i th source.

4.4 Grey Wolf Optimizer

GWO mimics the hierarchy of wolves' swarm when they go for a hunt [28]. The whole population of wolves is divided into four levels based on the fitness of a wolf named alpha, beta, delta, and omega. The leader with the best fitness falls into the alpha category. In the beta category, wolves (agents) are advisers of leader alpha. Delta wolves dominate omega wolves but report to alpha and beta. Delta can be categorized into the following: scouts, sentinels, elders, hunters, and caretakers. These are the ones who take the responsibility of watching the boundaries, protecting the pack, who were sometimes alpha or beta, helping alpha and beta in hunting, and taking care of ill and weak wolves, respectively. Omega category consists of the lowest fitness wolves. The hunt is done by exploration (searching and encircling) and exploitation process.

Mathematically, exploration can be represented using Eqs. 9 and 10.

$$\vec{D} = \left| C \vec{X}_p - A \vec{X}(t) \right|, \tag{9}$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \vec{D}, \tag{10}$$

where $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$, and $\vec{C} = 2\vec{r}_2$ are the coefficient vectors, t is the current iteration, X is the position vector of the grey wolf, \vec{X}_p is the position vector of prey, \vec{D} is the distance coefficient vector, \vec{r}_1 and \vec{r}_2 are random vectors $\in [0, 1]$ which allow wolves to reach any position between their original one and its neighbor's position and the components of \vec{a} linearly vary from 2 to 0.

During the hunting, it is considered that alpha, beta, and delta have better knowledge about the potential location of the source (prey).

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t) \right|,$$

$$\begin{aligned}\vec{D}_\beta &= \left| \vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t) \right|, \\ \vec{D}_\delta &= \left| \vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t) \right|,\end{aligned}\tag{11}$$

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha(t) - \vec{A}_1 \cdot (\vec{D}_\alpha), \\ \vec{X}_2 &= \vec{X}_\beta(t) - \vec{A}_2 \cdot (\vec{D}_\beta), \\ \vec{X}_3 &= \vec{X}_\delta(t) - \vec{A}_3 \cdot (\vec{D}_\delta),\end{aligned}\tag{12}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}.\tag{13}$$

$\vec{X}_\alpha(t)$, $\vec{X}_\beta(t)$, and $\vec{X}_\delta(t)$ are the position of the grey wolves or the leading elements with best fitness.

5 SNMF Fitness with Variable λ

The basic purpose of optimizing λ is to separate the individual basis components of different sources, and the sparseness present in the basis vectors helps to separate the individual components. But it is not necessary to have the higher sparsity with all perfect zeros in basis vectors. The low magnitude of elements in basis vectors may be present in the ideal basis vectors of any speech. So, the algorithm is to design with modification in basis vectors with variable λ in each iteration that forces the basis vectors to be closer to the ideal one. The change in λ after each iteration brings to the optimized value. This approach involves utilizing various optimization techniques to optimize the parameter λ , and subsequently determining the basis vectors and their corresponding weights.

5.1 Database Used and Evaluation Parameters

The objective is to design a system that can work for all kinds of monaural signals; a wide range of speech signals are chosen from the TIMIT dataset. This dataset is chosen for two speeches mixed-signal separation. TIMIT dataset consists of the English language speech signals spoken by 630 speakers from 10 dialect regions of the USA. For the experiment purpose, 20 male and 20 female speakers' speech signals (40 * 10 utterances from each speaker = 400 utterances) are randomly chosen from different dialect regions. The first two utterances are the same for all speakers and served for training purposes. The eight utterances are considered to make a mixture with the other speaker's utterances. The experiment has been done over the

separation of 2560 mixed speech signals in the opposite gender and same gender mixed-signal case.

An experiment is also performed over the MIR-1K dataset which has 1000 song clips recorded at a 16-kHz sampling rate with 16-bit resolution. These clips were taken from 110 songs that have both a music accompaniment track and a mixed track. These songs were sung by eight female and eleven untrained male singers. The first three singing voices and music clips of each artist were used for practice, and the next three clips were utilized for evaluation purposes. A total of 550 mixed signals were used for the experiment purpose.

6 Simulation and Results

To assess the effectiveness of the suggested strategy, 100 basis vectors for training signals for each speaker were extracted using the first two utterances of each speaker as these are common for all. The experiment is done for both opposite genders and same gender's mixed speech signals. Utterances of 40 random speakers (20 males and 20 females) are chosen to make the mixed speech. Utilizing 512-point Fast Fourier analysis, the spectrogram magnitude matrix for training signals and mixed signals was determined. Transform with 50% overlap and a 10 ms window size.

The parameters used to approximate λ by applying different optimization techniques are as follows:

For PSO, the number of particles chosen is 15, and maximum iterations are set to 10 [9]. For ACO, 25 ants are chosen with $\alpha = 1$, $\beta = 0$; evaporation rate, $\rho = 0.5$ and constant quality function $Q = 100$ [29]. In ABC, the number of food sources and the number of bees finding target are equal. The algorithm starts with 20 bees and limit the number to 100 with maximum number of cycles restricted to 10 [24, 26]. In GWO, two variables a and C are kept in the range of $[0, 2]$ and $[0, 3]$, respectively [23, 28, 30–32].

The results are obtained in two steps. At first, the experiment is performed to check the present work performance with chosen optimization techniques to optimize λ . Results are obtained for 320 mixed signals in each case in terms of SDR, SIR, PESQ, and STOI. In the second step, the best results obtained using the chosen algorithm for the proposed work are compared with the fixed λ in three conditions, i.e., $\lambda = 0.01, 0.1$, and 0.5 .

The bar chart shown in Fig. 2 presents the performance of the proposed model with approximating λ using different techniques for the same gender's mixed speech. The bar chart shows the superior performance of GWO-based implementation of SNMF for separation purposes over the other techniques.

The comparison of constants sparseness regularization factor based SNMF for all speech signals with the proposed algorithm is shown in Fig. 3. The proposed work has shown improvement in speech separation by 0.46 dB, 0.65 dB, 3.68%, and 11.22% in terms of SDR, SIR, STOI, and PESQ score, respectively.

Fig. 2 Source separation with controlled λ using different optimization techniques for same gender's mixed speech. SDR and SIR are measured in dB and STOI and PESQ are the scores

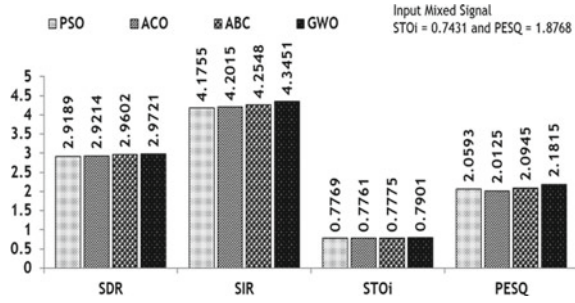


Fig. 3 Source separation with fixed and controlled λ for same gender

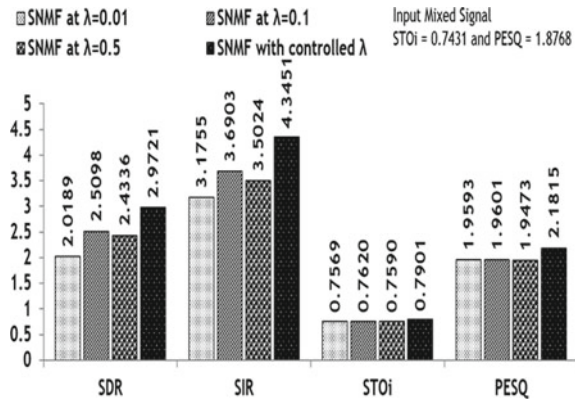


Fig. 4 Source separation with controlled λ using different optimization techniques for opposite gender's mixed speech

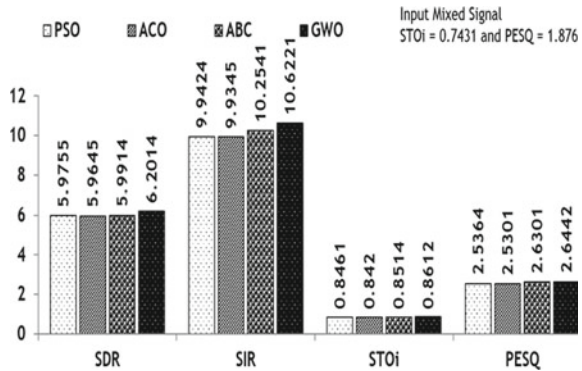


Figure 4 gives analysis of present model with optimizing λ using different OTs for opposite genders' mixed speech. The bar chart shows that the GWO-based optimization of λ approximates more accurate signal components as compared to other OTs.

The comparison of constant sparseness regularization factor-based SNMF for all speech signals with the present model is shown in Fig. 5. The proposed work has

Fig. 5 Source separation with fixed and variable λ for opposite gender

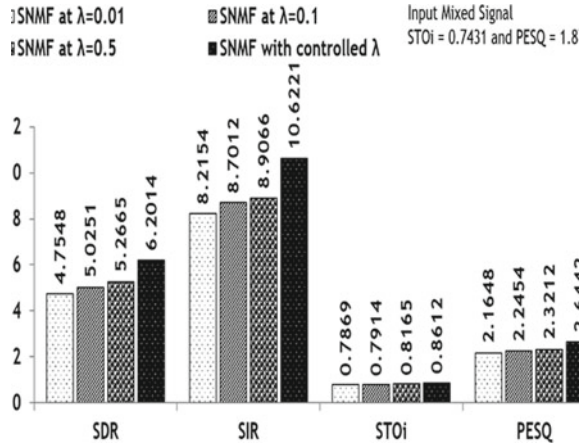
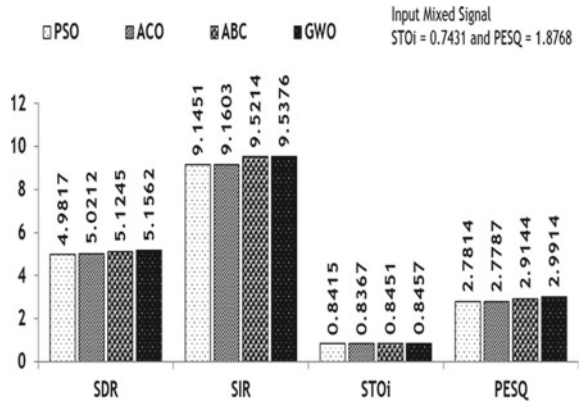


Fig. 6a Source separation with controlled λ using different optimization techniques for speech-music separation

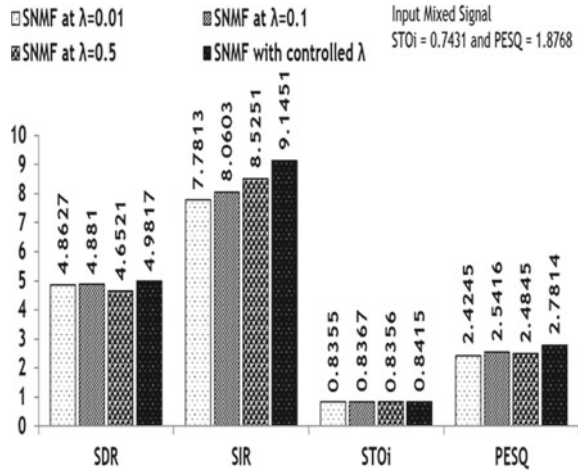


improved speech separation by 0.94 dB, 1.72 dB, 5.5%, and 13.96% in SDR, SIR, STOI, and PESQ, respectively.

Figure 6a describes the present method performance with approximating λ using different techniques for music-speech separation from a song. Here, PESQ and STOI are evaluated for speech signal only as these are the measure to find quality of required signals only.

The comparison of constant sparseness regularization factor-based SNMF for all songs with the present method is shown in Fig. 6b. This work has shown improvement in speech separation by 0.1 dB, 0.62 dB, 0.5%, and 9.4% in terms of SDR, SIR, STOI, and PESQ, respectively.

Fig. 6b Source separation with controlled λ using different optimization techniques for speech-music separation



7 Discussion and Conclusion

A common sparseness regularization factor in factorizing the spectrogram magnitude of the speech signal cannot provide proper basis vectors and their weights. So, variable λ -based SNMF is proposed that improves the approximation of basis vectors. To find the proper value of λ , four promising optimization techniques (PSO, ACO, ABC, and GWO) were applied, and GWO with the high capacity of avoidance of local extreme [28] has performed better as compared to others. Separation using fixed λ -based SNMF and proposed algorithm were tested. The experiments have shown that the performance for fixed λ -based SNMF for $\lambda = 0.01, 0.1, \text{ and } 0.5$ provide approximately similar results as $\lambda = 0.1$ may provide a better approximation for one signal but not for the other. In other cases, $\lambda = 0.5$ may provide a better approximation than other values of λ . So, a common algorithm with optimized λ for each case provides better results as found in the experiments. The results have shown improved performance in all the instances using GWO-based optimization of λ . However, the presented study is limited to the 0 dB mixture of two clean individual signals. It is required to extend the study for consideration of external noise factor or different level of mixing of target and unwanted speech signals. The presented work has also limited for the conditions when we have training signal before mixed speech separation. In further study, basis signals can be generated by the reinforcement learning of dominating speech signal in the mixed one to reduce the dependency of a specific training speech signal for mixed speech separation.

References

1. Bavkar S (2013) PCA based single channel speech enhancement method for highly noisy environment. In: *Advances in computing, communications and informatics (ICACCI)*, pp 1103–1107
2. Park H-M, Jung H-Y, Lee T-W, Lee S-Y (1999) Subband-based blind signal separation for noisy speech recognition. *Electron Lett* 35(23):982–984
3. Runqiang H, Pei Z, Qin G, Zhiping Z, Hao W, Xihong W (2006) CASA based speech separation for robust speech recognition. In: *Proceedings of the ninth international conference on spoken language processing (ICSLP)*, pp 2–5
4. Bach F, Jordan MI (2005) Blind one-microphone speech separation: a spectral learning approach. *17*, pp 65–72
5. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788–791
6. Grais EM, Sen MU, Erdogan H (2014) Deep neural networks for single channel source separation. In: *ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings*, pp 3734–3738
7. Hoyer PO (2004) Non-negative matrix factorization with sparseness constraints. *J Mach Learn Res* 5:1457–1469
8. Donoho DL, Elad M (2003) Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proc Natl Acad Sci* 100(5):2197–2202
9. Varshney YV, Abbasi ZA, Abidi MR, Farooq O (2017) Variable sparsity regularization factor based SNMF for monaural speech separation. In: *2017 40th international conference on telecommunications and signal processing, TSP 2017*, vol 2017
10. Févotte C, Gribonval R, Vincent E (2005) BSS EVAL toolbox user guide. Technical report 1706, IRISA
11. Vincent E, Gribonval R, Févotte C (2006) Performance measurement in blind audio source separation. *IEEE Trans Audio Speech Lang Process Inst Electr Electron Eng* 14(4):1462–1469
12. ITU (2001) Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs. In: *ITU-T recommendation*, vol 2, pp 1–32
13. Taal CH, Hendriks RC, Heusdens R, Jensen J (2011) An algorithm for intelligibility prediction of time—frequency weighted noisy speech. *IEEE Trans Audio Speech Lang Process* 19(7):2125–2136
14. Virtanen T (2007) Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans Audio Speech Lang Process* 15(3):1066–1074
15. Kang TG, Member S, Kwon K, Member S, Shin JW (2015) NMF-based target source separation. *IEEE Signal Process Lett* 22(2):229–233
16. Cooke M, Hershey JR, Rennie SJ (2010) Monaural speech separation and recognition challenge. *Comput Speech Lang* 24(1):1–15
17. Varshney YV, Abbasi ZA, Abidi MR, Farooq O (2017) Frequency selection based separation of speech signals with reduced computational time using sparse NMF. *Arch Acoust* 42(2)
18. Holland JH (2005) Genetic algorithms. In: *Holland understand genetic algorithms*, pp 12–15
19. Holland JH (1975) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. MIT Press, p 183
20. Stützle T, López-Ibáñez M, Pellegrini P, Maur M, Montes de Oca M, Birattari M, Dorigo M (2012) Parameter adaptation in ant colony optimization
21. Johnson DS, Aragon CR, Mcgeoch LA, Schevon C, Aragon R (1989) Optimization annealing: an experimental evaluation. *Oper Res* 37(6):865–892
22. Laskari EC, Parsopoulos KE, Vrahatis MN (2002) Particle swarm optimization for minimax problems. In: *Proceedings of the 2002 congress on evolutionary computation, CEC 2002*, vol 2, pp 1576–1581

23. Saad A, Dong Z, Karimi M (2017) A comparative study on recently-introduced nature-based global optimization methods in complex mechanical system design. *Algorithms* 10(4):120
24. Elbeltagi E, Hegazy T, Grierson D (2005) Comparison among five evolutionary-based optimization algorithms. *Adv Eng Inform* 19:43–53
25. Garg A, Juneja D (2012) A comparison and analysis of various extended techniques of query optimization, vol 3, no 3, pp 184–194
26. Xu Y, Fan P, Yuan L (2013) A simple and efficient artificial bee colony algorithm. *Math Probl Eng* 2013:1–9
27. Servet M (2015) A directed artificial bee colony algorithm, vol 26, pp 454–462
28. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
29. Wong KY (2008) Parameter tuning for ant colony optimization: a review. In: 2008 international conference on computer and communication engineering, pp 542–545
30. Li X, Yang G (2016) Artificial bee colony algorithm with memory. *Appl Soft Comput J* 41:362–372
31. Zhang X, Xiu X, Zhang C (2023) Structured joint sparse orthogonal nonnegative matrix factorization for fault detection. *IEEE Trans Instrum Meas* 3(72):1–5
32. Xie Z, Yang H, Ye Z (2022) Speech enhancement using group complementary joint sparse representations in modulation domain. *Appl Acoust* 1(201):109081

Depression Level Analysis Using Face Emotion Recognition Method



Sudarshan Khandelwal, Shridhar Sharma, Suyash Agrawal,
Gayatri Kalshetti, Bindu Garg, and Rachna Jain

Abstract Globally, most of the population faces depression or stress for a variety of reasons and at different stages of their lives. Stress in the modern world assists to depression over time, because of the hectic pace of our lives. Artificial intelligence systems can mimic the empirical system of a human person. Comprehensively capable machines and robots are capable of automatically identifying the mental state of a person from their facial expressions and body language. In order to determine and categorize the amount of depression, artificial intelligence (AI) and deep learning algorithms are employed to recognize the facial expressions of people in real-time. By using this software analysis to obtain revised depression levels, this model will enable psychiatrists and other medical personnel involved in human psychology to gain a new viewpoint. This model is trained on FER Plus dataset obtained from Kaggle, and later, CNN model is used to channelize the output with the accuracy of 62.44%. The primary driving force behind this effort is to increase the precision of the following model statement, which has the potential to have an effect on subsequent work and long-term implementation.

Keywords Convolutional Neural Network · Haar cascade · Deep learning · Emotion recognition · Face detection

1 Introduction

A psychiatric condition known as depression affects more than 300 million people globally. A depressed person has anxiety on a daily basis, which negatively impacts their relationships with their family and friends, worsens their health, and in the

S. Khandelwal (✉) · S. Sharma · S. Agrawal · G. Kalshetti · B. Garg
Department of CSBS, Bharati Vidyapeeth (Deemed to be University) College of Engineering,
Pune, India
e-mail: sudarshankhandelwal99@gmail.com

R. Jain
Department of Information Technology, Bhagwan Parshuram Institute of Technology,
New Delhi 110089, India

worst-case scenario, results in suicide. People are under greater strain than ever because of how quickly work and life are moving forward, which raises their risk of developing depression. Due to the significant disparity in the doctor-to-patient ratio globally, numerous patients may experience delays in receiving a prompt diagnosis. The primary indicators of depression include a lack of appetite and sleep, thinning hair, a loss of interest in once enjoyed hobbies, and a general lack of social life. All of these symptoms may occasionally exacerbate depression. According to [1], roughly 15% of individuals in India require active assistance for one or multiple mental health concerns, with a prevalence of depression observed in one out of every 20 individuals. As stated by World Health Organization (WHO), depressive disorders affect 350+ million people globally of all ages. Depression is one of the most severe but prevalent mental illnesses in the world (also known as depressive disorder or clinical depression).

The inability to perform necessary everyday activities for a minimum period of two weeks could be caused by depression's severe impairments. Apart from either a low mood or loss of interest, at least four other symptoms such as issues with concentration, self-image, food, energy, or persistent thoughts related to death or suicide must exist consistently for two weeks. Depression is receiving more and more attention from a variety of connected areas due to its dangers and the recent increase in occurrence. Depression may be treated with medicine, psychotherapy, and other therapeutic techniques even if it is a serious condition. The more quickly that therapy can start, the better it is. The detection of depression in its initial phase is imperative in managing the condition and curtailing the societal and financial burdens associated with this disorder.

The primary sources of data utilized in conventional methods of diagnosing depression include patient self-reports during clinic interviews, behaviors noticed by friends or family, and questionnaire like the Patient Health Questionnaire (PHQ-9). However, because they all rely on subjective evaluations, the outcomes might vary depending on the situation or the setting. To arrive at a somewhat objective diagnosis, numerous clinical specialists must be involved. Early-stage diagnoses and reassessments for monitoring therapy results are frequently restricted and time-consuming as the number of depressed individuals rises. In light of this, it is envisaged that recognition based on machine learning will enable objective and speedy diagnosis, assuring excellent clinical treatment quality and considerably reducing the probability of harm in actual life.

Depression-related behavior disorder-based indications for depression detection, such as voices, facial expressions, gestures, gaits, and eye movements, are becoming more prevalent under its impact. This study focuses on analyzing facial expressions to detect people who may be at risk for depression. Videos or photos are mostly used in studies over depression based on facial expressions. With video-based technology relying heavily on picture processing through converting motion footage into static shots, these limitations hinder its accuracy and reliability. The recognition performance will be impacted if these aspects are not properly addressed.

Therefore, to identify stress or depression levels in a certain facial image in real-time, this model will be employing Convolutional Neural Network (CNN) and face

recognition techniques that might be done via the Haar cascade approach. Depression signal's representations are extracted using deep learning (DL) from video and translate them into frames of visuals for a depression diagnosis in order to improve existing medical therapy. Here are the objectives mentioned that are being pursued:

- To detect refined depression level statistics from facial expressions.
- To close the gap created by the unbalanced doctor–patient ratio.
- To improve the performance and accuracy through large-scale datasets.

1.1 Motivation and Main Contribution

According to a meta-analysis of 41 studies, general practitioners and sometimes expert psychiatrists can only identify depression in 47% of the cases. More precise tools are required, with the goal of assisting psychiatrists in their decision-making rather than replacing humans in the diagnosis process. We will provide psychiatrists or psychologists a piece of software to use in order to assess the precise level of depression that a patient is likely to be experiencing. This software will provide a number along some suggestion as per scale from 1 to 10 as an output which depicts a simple mental pain health scale, by dividing the scale from:

- Mild Depression—from 1 to 3.
- Moderate Depression—from 4 to 6.
- Severe Depression—from 7 to 10.

Advantages of proposed solution:

- Depression detection using computer vision has higher accuracy than average general human practitioners, requiring only 30–40 s of image data to get processed and detect the depression level.
- Using this software along with their own conventional approaches will give psychiatrists a new perspective on decision-making and a quick, accurate understanding.
- A one-time software investment can produce superior outcomes for the long-term diagnosis of a patient at various stages of depression.

The subsequent parts of the article are organized as follows: Sect. 2 offers a brief overview of related studies in this domain; Sect. 3 discusses the research methodology and provides a description of the dataset utilized. Section 4 provides valuable insights into the results and decisions, while Sect. 5 concludes the paper by outlining its future possibilities.

2 Related Works

Certain scholars have shown interest in the identification of facial expressions and contributed their studies and work accordingly. Fan and Tjahjadi provided a framework for recognizing facial expressions that combines CNN and custom characteristics. They discovered that the neural network could extract texture information from face patches to provide outstanding recognition results and that the incorporation of CNN had a positive impact on the detection of facial expressions [5].

Reddy et al. suggested combining deep learning features with a manual technique for detecting facial expressions in an integrated manner. In trials, they confirmed the method's applicability in natural settings, which showed the method's efficacy when deep learning and manual production were combined [6]. Liang et al. proposed an old hand-crafted facial representation, which could only show superficial characteristics. They presented the Patch Attention Layer of embedding handmade features, a novel approach to facial emotion detection that is based on reinforcement (patch) of interest, in order to overcome this constraint and learn the characteristics of each patch on face pictures [7].

Jain et al. [8] present the theory in which Recurrent Neural Network (RNN) and Convolutional Layer were combined to extract information from face photos, hybrid convolutional-recursive neural network for facial emotion detection [8]. Avots et al. discovered human emotions by analyzing audio-visual data. Also, they used the Viola-Jones facial recognition algorithm and multiple datasets as test sets to classify the emotions on face photos [9].

Li et al. established on the face recognition L1 norm, constructed a deep learning-based network for two-dimensional principal component analysis, assessed its performance using a collection of facial images, and reached the conclusion that the network exhibited dependability [10]. Bernhard et al. realize since emotions significantly affect human decision-making, so they used deep learning to enhance the outcomes of emotion identification. RNNs and transfer learning achieved better results compared to traditional machine learning methods, which were primarily influenced by their use in emotion computing applications [11]. Kumar et al. covered creating representations of abnormal facial expressions using computer vision techniques and emotional anomalies. They discovered that deep CNN might be crucial in training and classification of face expressions, which gave visual surveillance systems a new visual modeling technique [12].

Mishra et al. used Convolutional Neural Network to identify various intensity levels and emotions over faces of human, laying the groundwork and providing aid for further research on computer emotion identification [13]. Björn Schulle et al. offered the first amalgamate open Audio/Visual Emotion and Depression identification Challenge, AVEC 2013. It tackles two child challenges: the estimate of a self-reported state of depression and the detection of the valence and arousal of the emotional dimension in continuous time and value [14]. Young-Shin Lee et al. put forth a model utilizing AI which aids in the identification and assessment of depressive disorders. A model utilizing fast region-based Convolutional Neural Networks

(R-CNNs), an advanced deep learning approach that comprehends vector-based data, can assist in diagnosing depressive disorder by examining alterations in the eye and lip positions and deducing emotions from a set of photographs [15].

Sharifa Alghowinem et al. in order to conduct a binary classification job examined the effectiveness of eye movement characteristics collected from face films using Active Appearance Models. The model found that by employing statistical measurements with SVM classifiers, the low-level features of eye movement yielded a 75% accuracy rate across the entire interview [16]. P. Ramesh Naidu et al. in order to take advantage of the fact that mouth, head, and eye movements differ from normal situations when a person is under stress proposed an algorithm to recognize stress from photographs captured with a camera and a deep neural network that receives facial landmarks as input [17]. Lang He et al. proposed a promising psychological investigation by discovering some variations in facial expression and speech b/w healthy and depressed people. They provide the objective markers for automated depression estimate in the databases [18].

Karen Schepman et al. following a standardized evaluation of diagnostic and mood symptoms conducted an experiment in which participants supplied accurate data after being given a computerized face emotion detection test. The AVEC2014 dataset was selected so that studies utilizing unprocessed visual and audio data could be conducted [19]. Ninad Mehendale put forth emotion recognition and proposed that it is crucial to creating successful human-computer interactions. The input is a picture. Prior to identifying the face, the face is first detected in the picture, from which key traits are then extracted. Extraction of the expression features from the picture is the next stage. The classifier is then given the retrieved characteristics in order to classify the output as expressions [20]. Bhavna Singh Parihar et al. say that depression is the most severe mental disease that can be found with a variety of symptoms in different people, making it challenging to diagnose. A CNN model is developed that analyzes a person's facial traits to determine whether or not they are depressed [21].

Qian Chen et al. proposed sequential fusion approach for depression identification using faces in order to concurrently learn face movements and appearance in a single framework, and a chained-fusion technique is presented for mining the associated and complimentary patterns of depressions in multimodal learning [22]. Asim Jan et al. asserted that DL approaches can be useful in the field of issues related to mental health since they recognize the value of gathering thorough information to characterize the various psychiatric diseases. Visual and verbal data are essential for building an effective artificial system for recognizing sadness since a system utilizing a camera and microphone can quickly collect them [23].

S. Modi et al. put forth that the primary driving force behind the effort is to increase the correctness of the defined model statement, that might have an effect on further research. Additionally, a comparison of the transfer learning model and the used model will give out the study effort for necessary innovation [24]. Weitong Guo et al. come up with a unique method for identifying probable depression risk and was based on two separate deep belief network (DBN) models. The other model utilizes a Kinect to collect 3D face points and extract dynamic characteristics, whereas the first

model relies on an optical camera to capture facial photos and extract 2D appearance features. The two models are combined to provide the final decision outcome. Finally, we assess each deep model on our constructed dataset [25].

G. Giannakakis et al. says, video-recorded facial signals can be used to identify and analyze emotional states associated with stress and anxiety. Mainly concentrated on semi-voluntary and non-voluntary facial signals to more accurately evaluate the emotion representation [26]. Nandita Sharma et al. developed a theory where stress is being modeled using Bayesian networks, artificial neural networks, and support vector machines [27]. Amir Hasanbasic et al. used wearable sensors to monitor ten students in order to gauge their degrees of exam-related stress. Different categorization techniques were utilized as input with characteristics of the ECG and electrodermal activity signals [28]. Andre Teixeira Lopes et al. for the purpose of recognizing facial expressions suggested a straightforward approach that combines well-known techniques like Convolutional Network and certain picture preprocessing processes [29].

Octavio Arriaga et al. proposed a real-time vision system that can recognize faces, identify genders, and identify emotions [30]. Ali Mollahosseini et al. suggested a deep neural network design to handle the FER problem across several well-known facial datasets. After two convolutional layers, our network incorporates four inception layers, with each one being succeeded by max pooling. The network, which consists of a single component, classifies recorded face pictures as input into either the six fundamental expressions or the neutral expressions. Over seven open to publicly accessible facial expression databases—MultiPIE, MMI, CK+, DISFA, FERA, SFEW, and FER2013—we conducted extensive studies [31].

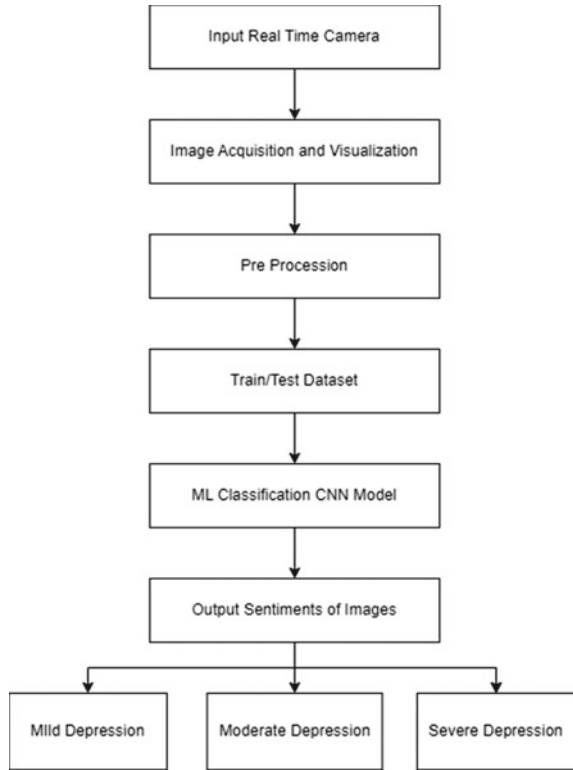
In summary, deep learning may considerably increase the ability of recognition of facial expression, particularly by implementing the hybrid model, whereas the classic manual approaches are no longer appropriate for the present study on facial expression recognition. Even while facial expression recognition has made considerable progress in the past, there are not many research that pair it with psychological analysis.

3 Research Methodology

Let us understand this segmentation with the help of a flowchart which depicts the flow of tasks processing in the system from input to output. As seen in Fig. 1, a proper procedure has been showed as how the program will work and what could be the possible outcomes, which are Mild Depression, Moderate Depression, and Severe Depression. A CNN model will be learning and getting trained based on the input images in the ML classification stage (Fig. 2).

The process of analyzing emotions from a video stream involves several stages. The algorithm first broadcasts a live video feed, after which it gathers a certain number of frames for use. Then, a CNN model that was trained and validated on a

Fig. 1 Flowchart for depression detection

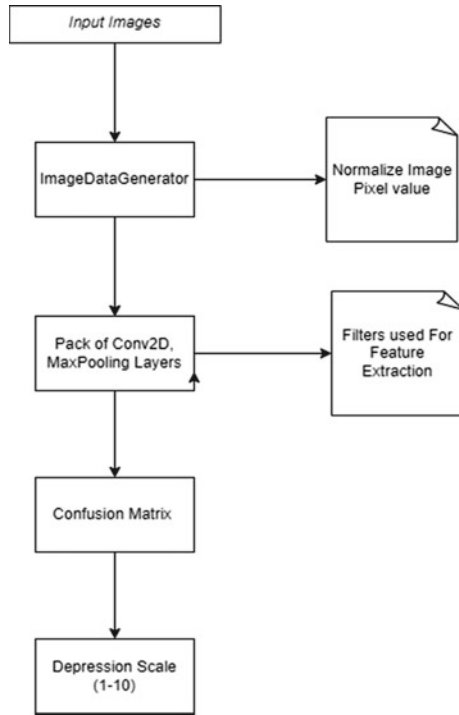


big dataset of images representing six to seven different emotions is compared to these frames.

The ImageDataGenerator script is then used to turn all grayscale-textured pictures into datasets, and sets of tensor image data are generated with real-time data augmentation. The training dataset is fitted across 60 epochs with various layers of Conv2D and MaxPooling, with a rate of learning of 0.0001 and category cross-entropy loss. There are separate training and validation components for the dataset. With a 25% dropout rate, the set of available classes are used to map labels to pictures.

A confusion matrix is created to provide an accuracy score for each image's predicted emotion, and model's accuracy and loss are shown using graphs during training and testing. After utilizing the CNN model, the model analyzes frames and identifies emotions in each of them and thereafter retrieves the emotions for each image in the stack. It has the ability to detect positive and negative emotions and ascertain the degree of these emotions. Subsequently, it can also estimate whether there has been a shift in emotions over time. The percentage level of emotion falls under a certain threshold value, and then, the model generates results displaying depression levels as mild, moderate, or severe and providing an appropriate suggestion.

Fig. 2 System architecture for depression detection



3.1 Dataset Description

The FER-2013 database, often known as FERPlus, has about 35k face expressions based on seven fundamental expressions, published in 2013 as part of a Kaggle challenge. This dataset is taken from Kaggle and the major contributor is Mr. Manas Sambare. The web-based photos are gathered, transformed to grayscale, and scaled to (48×48) . Since this database reports a $68\% \pm 5\%$ human accuracy, theoretically it might be mislabeled. However, this model used it as pre-training data because it is a sizable spontaneous collection of required expressions over the faces. The faces in the images have been automatically aligned so that they occupy a similar amount of space and are positioned approximately at the center.

Established on the facial emotions shown in Fig. 3, the facial expressions are categorized into one of seven types (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Neutral, 5 = Sad, 6 = Surprise). The training set consists of 28,709 examples.

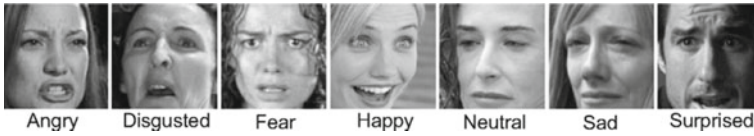


Fig. 3 Facial expression in FER-2013

4 Result and Discussion

Here, two activation functions are tested on the same dataset to find out the best results. First function is **ReLU**, which is a piecewise linear function that will result the output only if the required input is positive, else, it will output zero and other is **SIGMOID** which is a nonlinear function that assigns any number ranging between 0 and 1, inclusive, to itself. The figure given below is the over-fitted model and accuracy is 72.23% and split is of (80–20), but still it is giving false-positive results. So, this model was discarded and the test–train split is changed (Fig. 4).

In the graph (Fig. 5), model loss is decreased after many epochs retry which makes it less optimized. Feature extraction is stabilized between 10 and 20× scale values.

Fig. 4 Model accuracy (80–20) using ReLU

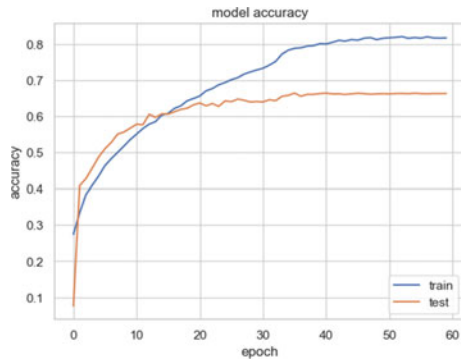


Fig. 5 Model loss (80–20) using ReLU

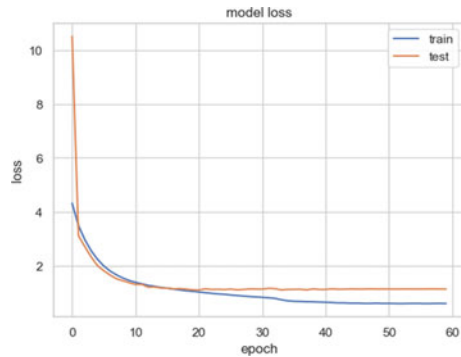
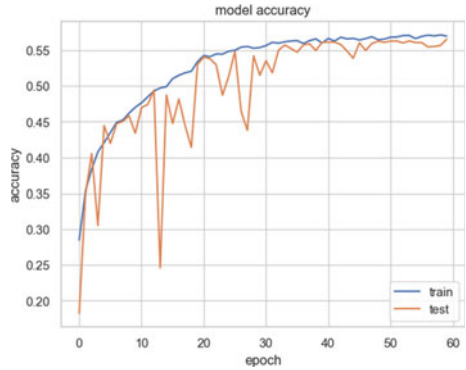


Fig. 6 Model accuracy (70–30) using SIGMOID



Here, SIGMOID nonlinear function is used to attain a higher accuracy and the splitting ratio of train and test is set to (70–30) which still resulted in unstable test case accuracy as shown in the graph (Fig. 6), and this makes it non-reliable.

But here as shown (Fig. 7) that model loss is not getting stabilized so the feature extraction is very unoptimized. So, this model is also not very much reliable either.

Now, the same splitting ratio for training and testing of (70–30) is reused, but this time again using the ReLU function on the same dataset. And as the result, it does provide us with a much higher accuracy of 83.88% and very less model loss as seen in Figs. 8, 9 and Table 1.

So, at last, it is decided to go with ReLU function in our model as it is giving the best accuracy for the (70–30) split and with vey less model loss and high feature extractions.

Fig. 7 Model loss (70–30) using SIGMOID

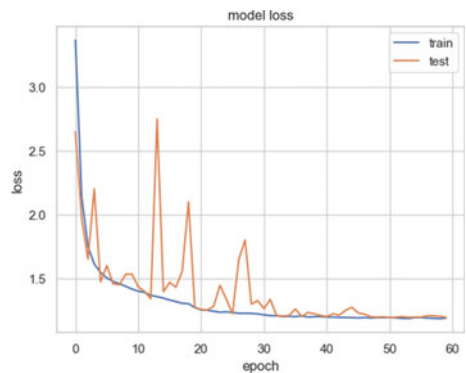


Fig. 8 Model accuracy (70–30) using ReLU

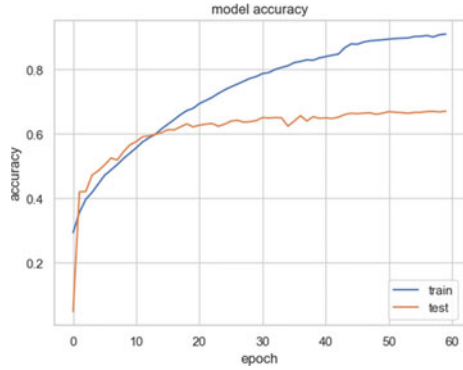


Fig. 9 Model loss (70–30) using ReLU

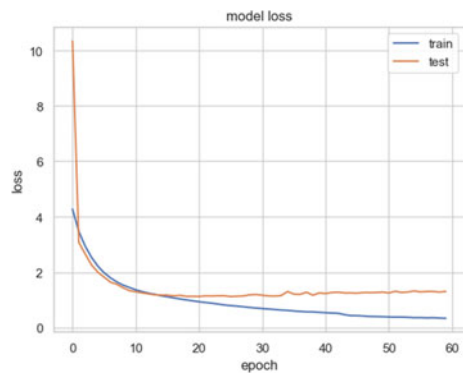


Table. 1 Accuracy values from all functions used

Function used	Split ratio	Final train accuracy (%)	Validation accuracy (%)
ReLU	80–20	72.23	63.66
SIGMOID	70–30	59.85	56.51
ReLU	70–30	83.88	64.03

5 Comparative Analysis

As seen in Table 2, another published paper under the ICAMIMIA conference named as “Deep Learning Based Facial Emotion Recognition using Multiple Layers Model, Sandra et al. [32]” achieved an accuracy of 60% using ResNet architecture and our model has achieved an accuracy of 64.03% using AlexNet.

Table 2 Comparative analysis with ICAMIMIA (2021) published paper

Architecture	Training accuracy (%)	Testing accuracy (%)	Layers
AlexNet (CNN)	83.88	64.03	13
ResNet50 (ANN)	65	60	50

6 Limitations

- **Limited emotions:** The dataset only includes seven basic emotions, which means that the model may struggle to recognize more nuanced emotions or subtle changes in facial expressions.
- **Biased dataset:** It may be challenging for the model to accurately differentiate other emotions due to the skewness of the dataset toward neutral and happy expressions.
- **Variations in facial types:** The model may struggle to generalize to unfamiliar faces, not present in the training data, as face expressions can significantly vary between ages, individuals, and cultures.
- **Facial occlusions:** Emotion interpretation can be hindered by accessories like facial hair, glasses, or other items that obscure facial expressions.
- **Low sample size:** Overfitting and limited generalization to new data can occur due to the relatively small number of samples in the dataset.
- **Lack of contextual information:** Information regarding the context in which the facial expressions were captured, such as the situation, surroundings, or cultural background, is not included in the dataset, making it unavailable for the model.

This can make it difficult for the model to accurately interpret the emotions.

7 Conclusion and Future Scope

The research study found that the model was successful in predicting the nature of images, but improvements were needed for future purposes. The suggested AI treatment had potential applications in detecting drowsiness in drivers and monitoring elderly patients' tension and anxiety using emotion detection. The model could also provide audio-based refinement for depression detection and suggest mental activities, a healthy diet, and music to promote stability and prevent depressive thoughts. Further enhancements were necessary to integrate the model into an Android or iOS application for identifying potential depression in friends and family.

References

1. Health & consumer protection directorate general. Mental health in the EU (2008)
2. Marcus M, Yasamy MT, van Ommeren M, Chisholm D (2012) Depression, a global public health concern, pp 1–8
3. World Health Organization (2017) Depression and other common mental disorders: global health estimates. Tech. Rep.
4. Luxton DD (2015) Artificial intelligence in behavioral and mental health care. Academic Press
5. Fan X, Tjahjadi T (2019) Fusing dynamic deep learned features and handcrafted features for facial expression recognition. *J Vis Commun Image Represent* 65:102659
6. Viswanatha Reddy G, Dharma Savarni C, Mukherjee S (2020) Facial expression recognition in the wild, by fusion of deep learnt and hand-crafted features. *Cogn Syst Res* 62:23–34
7. Liang X, Xu L, Liu J et al (2021) Patch attention layer of embedding handcrafted features in CNN for facial expression recognition. *Sensors* 21(3):833
8. Jain N, Kumar S, Kumar A, Shamsolmoali P, Zareapoor M (2018) Hybrid deep neural networks for face emotion recognition. *Pattern Recogn Lett* 115:101–106
9. Avots E, Sapiński T, Bachmann M, Kamińska D (2019) Audiovisual emotion recognition in wild. *Mach Vis Appl* 30(5):975–985
10. Li YK, Wu XJ, Kittler J (2019) L1-2D2PCANet: a deep learning network for face recognition. *J Electron Imag* 28(02), 023016, 1
11. Kratzwald B, Ilić S, Kraus M, Feuerriegel S, Prendinger H (2018) Deep learning for affective computing: text-based emotion recognition in decision support. *Decis Support Syst* 115:24–35
12. Kumar RK, Garain J, Kisku DR, Sanyal G (2018) Estimating attention of faces due to its growing level of emotions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), Salt Lake City, UT, USA
13. Mishra S, Prasada GRB, Kumar RK, Sanyal G (2017) Emotion recognition through facial gestures—a deep learning approach. In: Proceedings of the international conference on mining intelligence and knowledge exploration. Springer
14. Valstar M, Schuller B, Smith K, Eyben F, Jiang B, Bilakhia S, Schnieder S, Cowie R, Pantic M (2013) AVEC 2013: the continuous audio/visual emotion and depression recognition challenge. In: Proceedings of the 3rd ACM international workshop on audio/visual emotion challenge, pp 3–10
15. Lee YS, Park WH (2022) Diagnosis of depressive disorder model on facial expression based on fast R-CNN. *Diagnostics* 12(2):317
16. Alghowinem S, Goecke R, Wagner M, Parker G, Breakspear M (2013) Eye movement analysis for depression detection. In: 2013 IEEE international conference on image processing. IEEE, pp 4220–4224
17. Naidu PR, Sagar SP, Praveen K, Kiran K, Khalandar K (2021) Stress recognition using facial landmarks and CNN (Alexnet). *J Phys Conf Ser* 2089(1):012039
18. He L, Niu M, Tiwari P, Marttinen P, Su R, Jiang J, Guo C, Wang H, Ding S, Wang Z, Pan X, Dang W (2022) Deep learning for depression recognition with audiovisual cues: a review. *Inf Fusion* 80:56–86
19. Schepman K, Taylor E, Collishaw S, Fombonne E (2012) Face emotion processing in depressed children and adolescents with and without comorbid conduct disorder. *J Abnorm Child Psychol* 40:583–593
20. Mehendale N (2020) Facial emotion recognition using Convolutional Neural Networks (FERC). *SN Appl Sci* 2(3):446
21. Parihar BS, Satam SS, Satam SS, Dange K (2020) CNN model for depression detection using JAFFE dataset. *PiCES* 4(6):135–139
22. Chen Q, Chaturvedi I, Ji S, Cambria E (2021) Sequential fusion of facial appearance and dynamics for depression recognition. *Pattern Recogn Lett* 150:115–121. ISSN 0167-8655
23. Jan A, Meng H, Gaus YFBA, Zhang F (2018) Artificial intelligent system for automatic depression level analysis through visual and vocal expressions. *IEEE Trans Cogn Dev Syst* 10(3):668–680. <https://doi.org/10.1109/TCDS.2017.2721552>

24. Modi S, Bohara MH (2021) Facial emotion recognition using convolution neural network. In: 2021 5th international conference on intelligent computing and control systems
25. Guo W, Yang H, Liu Z, Xu Y, Hu B (2021) Deep neural networks for depression recognition based on 2D and 3D facial expressions under emotional stimulus tasks. *Front Neurosci* 15:609760. <https://doi.org/10.3389/fnins.2021.609760>
26. Giannakakis G, Pediaditis M, Manousos D, Kazantzaki E, Chiarugi F, Simos PG, Marias K, Tsiknakis M (2017) Stress and anxiety detection using facial cues from videos. *Biomed Signal Process Control* 31:89–101
27. Sharma N, Gedeon T (2012) Objective measures, sensors and computational techniques for stress recognition and classification: a survey. *Comput Methods Programs Biomed* 108(3):1287–1301
28. Hasanbasic A, Spahic M, Bosnjic D, Mesic V, Jahic O (2019) Recognition of stress levels among students with wearable sensors. In: 2019 18th international symposium INFOTEH-JAHORINA (INFOTEH). IEEE, pp 1–4
29. Lopes AT, De Aguiar E, Oliveira-Santos T (2015) A facial expression recognition system using convolutional networks. In: 2015 28th SIBGRAPI conference on graphics, patterns and images. IEEE, pp 273–280
30. Arriaga O, Valdenegro-Toro M, Plöger P (2017) Real-time Convolutional Neural Networks for emotion and gender classification. arXiv preprint [arXiv:1710.07557](https://arxiv.org/abs/1710.07557)
31. Mollahosseini A, Chan D, Mahoor MH (2016) Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE winter conference on applications of computer vision (WACV), pp 1–10
32. Sandra L, Heryadi Y, Suparta W, Wibowo A (2021). Deep learning based facial emotion recognition using multiple layers model. In: 2021 international conference on advanced mechatronics, intelligent manufacture and industrial automation (ICAMIMIA). IEEE, pp 137–142

Early Prediction and Detection of Anxiety Level Using Support Vector Machine



Tisha Sadariya and Shanti Verma

Abstract According to the National Library of Medicine, anxiety is a physiological and behavioral status induced in humans by an intimidation to interests or survival. It is typify by increased provocation anticipation, autonomic and immunologic establishment, and specific behavior patterns. The role of these changes is to cope with an unsympathetic or unpredicted condition. Social media is now a companion of every human. It is also considered as the basic necessity of humans. Social media usage has dark and light sides, but nowadays with excessive usage of social media, there are many dark sides seen in humans. Anxiety is one of the dark sides of excessive usage of social media. In this paper, authors try to build a classification model based on demographics and psychological factors using a Support Vector Machine (SVM). Authors used a secondary dataset available on kaggle.com. The result of the study claims the model accuracy of 95.31% to classify anxiety level. Authors also compare the results of SVM with other classification algorithm, Decision Tree, Naïve Bayes, and K-Nearest Neighbor (KNN), and found that SVM model accuracy is higher than other classification algorithms.

Keywords Classification · Prediction · Anxiety · Media usage · Support Vector Machine

1 Introduction

Social media is a medium where we can connect with peoples around the world using websites and applications. We all recognize that knowledge is power, but only few understand the role social media has played to the society. In current world, social media acts as an important role in influencing our culture and economy. The online social media has caused strong revolution in the way people be in touch and interact with each other. Social media has both brighter and darker sides on human health. In

T. Sadariya · S. Verma (✉)

Department of Computer Applications, L. J. University, Ahmedabad, India

e-mail: verma.shanti@gmail.com



Fig. 1 Benefits of social media usage

the past decade, we saw great change in people's life because of social media usage [1].

There are various positive impacts of social media usage on young generations which are engagement, critical thinking, collaboration, creativity, and empowering the introverts which depicts in Fig. 1 [2]. In India, many guidelines are for usage to social media contents by different age groups [3]. By Congress in the Children's Online Privacy Protection Act (COPPA), the people having age 13 or more can use social media content without their parent's permission (Fig. 2).

Figure 1 depicts the various dark sides of usage of social media. Major drawbacks for extensive use of social media are: peer-to-peer; improper content; lack of privacy of contents; and influence of marketing agencies [2, 4]. In India, many schools use social media platforms to share homework and worksheets to students. This makes children allow to get usage of parents smart phones. Many schools also use YouTube platform to teach pre-primary school children's for better understanding and good execution of contents [5]. Young generations are mostly use social media to connect with friends and family [6].

Many authors used word social media depression which is related to the health issues faced by people for excessive social media usage [7]. The challenges faced by many young generation people for excessive social media usage are depicted in Fig. 3 which are isolation, cyber bullying, anxiety, stress, lack of sleep, decreased productivity, false connections, lack of privacy, and unrealistic expectations [8, 9]. In this paper, authors try to find the impact of social media usage on anxiety level.

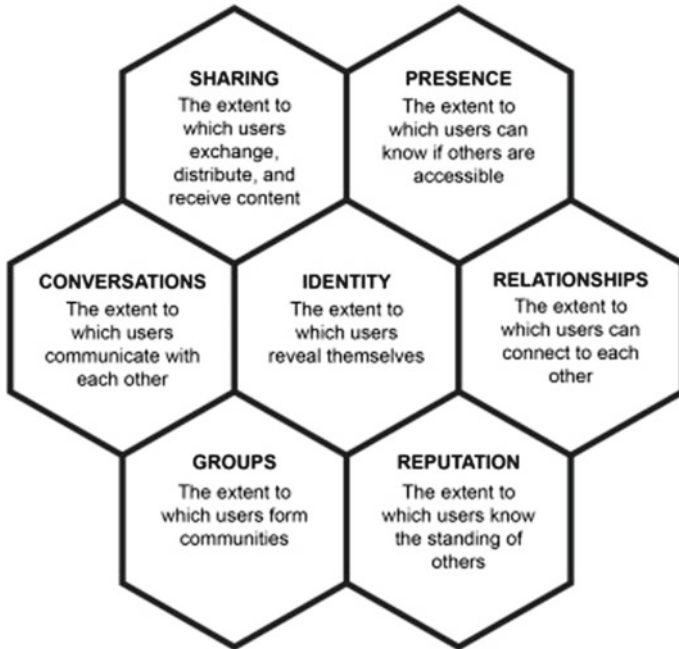


Fig. 2 Dark sides of social media usage



Fig. 3 Effects of social media on mental health

2 Objective of Study

The objectives of research study are given below:

- (1) To study association between age and media usage.
- (2) To study gender-wise anxiety level analysis.
- (3) Create a SVM model to classify anxiety level on the basis of media usage and demographic factors.

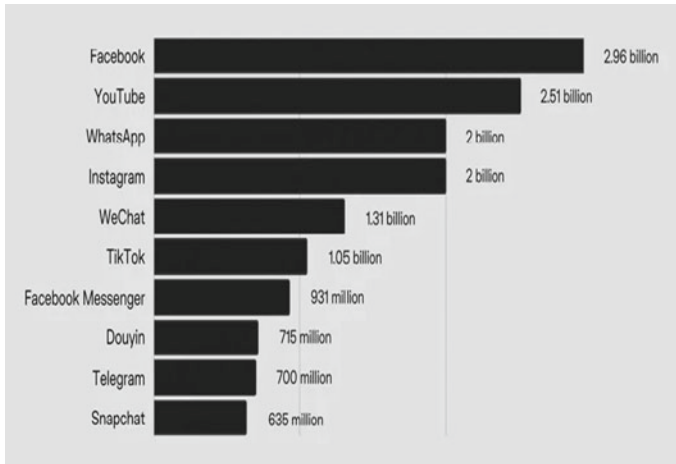


Fig. 4 Customer base-wise social media platform

3 Literature Review

3.1 Related Work: Social Media Usage

Survey identifies that mostly young people spend more time on social media websites [10]. Any websites or applications which provide real-time communication are considered as social media [2, 3, 11, 12]. On the basis of the social media usage, we can find that impact of it on humans [13]. There are various theories like conceptualized cognitive, emotional, attitudinal, and behavioral self-effects that may occur before, during, and after message creation/sending on social media. On the basis of the receiver reactions, sender makes unrealistic theories in their mind. Finally, reception effects are enhanced by self-effects [9]. The usage of various social media website by humans shown in Fig. 4. In this figure, we can see that major stakeholder is Facebook.

3.2 Related Work: Social Media Effects on Mental Health

Social media usage is an essential part of our daily life nowadays. Due to excessive usage of online social media, many people face mental health problems. Many authors found influence of social media use on depression, anxiety, and psychological distress in adolescents [1, 4]. During COVID-19, social media has provided a platform for people to access updated information [14, 15]. After COVID-19, our life changed very much in terms of communication, information access, and education. Through social media applications, public communication and interaction go beyond personal

message delivery to seeking correct information and the full scope of the COVID-19 pandemic to develop a real sense of virus prevention. The emergence of the COVID-19 outbreak has changed life patterns in response to preventive measures. [16]

3.3 Related Work: Support Vector Machine

Support Vector Machines (SVMs) are still one of the most popular and precise classifiers. The accuracy of using optimal parameter values in kernel functions is as a determinant to obtain maximum accuracy results on image retrieval with Support Vector Machine (SVM) classification [17]. Experiments conducted in this study aimed to obtain optimal Gaussian/Radial Basis Function (RBF) kernel function parameter values on nonlinear multi-class Support Vector Machine (SVM) method [18, 19]. SVMs are used in applications like handwriting recognition, intrusion detection, face detection, email classification, gene classification, and in web pages. In this paper, authors used a RBF-based SVM to classify anxiety level on the basis of media usage and demographics [7, 20].

4 Methodology

4.1 Data Collection

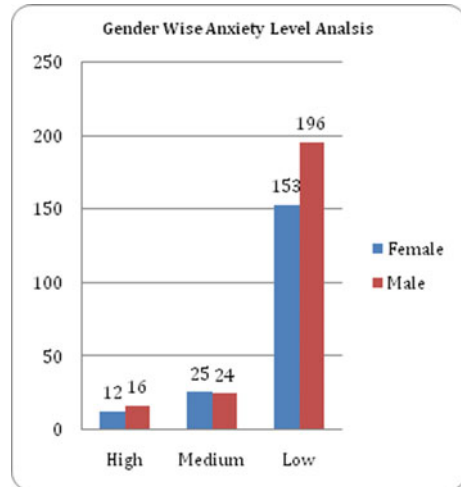
To achieve the defined objectives in paper, authors used a secondary dataset. The dataset used in study is available in (<https://www.kaggle.com/datasets/spscientist/mental-illness>).

There are nine columns in the dataset which are defined as age, age category, sex, qualifications, habitat, media use (h/day), media category, well-being score, well-being category, anxiety score, and anxiety category. First eight columns of the dataset are considered as independent and the anxiety category is considered as a dependent variable. Anxiety category has three different categories: Low, Medium, and High. Authors transform these categories as 0, 1, and 2 for classification.

4.2 Descriptive Data Analysis

The dataset used in study has some demographics of users like age and sex. Figure 5 depicts the gender-wise analysis of users for anxiety level. In the dataset, there are 426 records out of which 196 have gender female and 236 have gender male. The results of Fig. 1 show that 56% male and 24% have low anxiety level.

Fig. 5 Gender-wise anxiety level analysis



The dataset used in the study does not show that normal distribution. So, the first step authors used in analysis is to normalize the dataset using Z score. After normalization, authors study the effect of age on media usage (h/day). The age group used in study has minimum age 18 years and maximum age 79 years. Media usage (h/day) used in study has a minimum value of 4 h/day and maximum value 16 h/day. In this analysis as shown in Fig. 6, authors found that there is a highly negative correlation (-0.23398) between age and media usage (h/day). Study also shows that the 24–27 age group people use social media websites most per day. It shows that the social media is most used by adults between the age 24 and 27 years according to dataset used in study.

Figure 7 depicts the heat map for the dataset used in study. Heat map is a data visualization technique used for better understanding of the dataset. In this map, color coding is used to depict the value. This map also shows the correlation values of every column with other columns. This map is useful to check which columns have more impact on results of study [21, 22]. We can use the results of the heat maps to decide the threshold values for reducing the dimensions of the dataset [23].

4.3 Tools Used for Analysis

Authors used a Python open-source tool for data visualization, data normalization, data reduction, and data classification using SVM. The main feature of Python language is having an extensive amount of packages. In this paper, the authors used

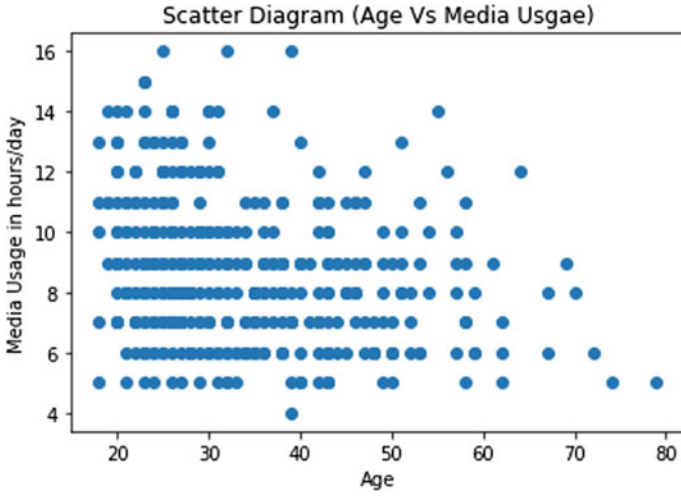


Fig. 6 Scatter diagram age versus media usage

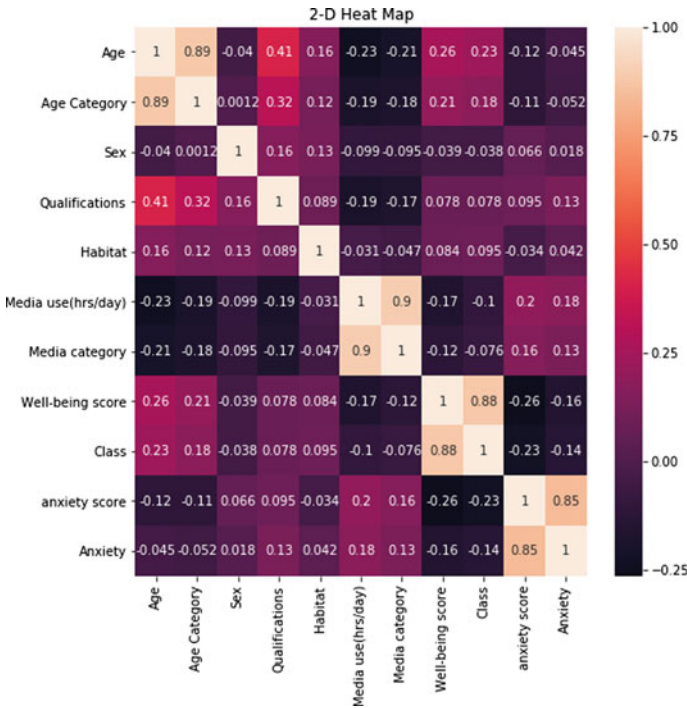


Fig. 7 Heat map

the sklearn package of Python which is used for data classification. For data visualization, matplotlib package is used, and for data normalization, StandardScaler package is used. For dimension reduction, principal component analysis (PCA) is used in study.

4.4 Flowchart of Proposed Work

Figure 8 depicts the flow of work carried by authors to build classification model using secondary dataset on the given threshold value. First step is to collect a secondary dataset available in kaggle.com (<https://www.kaggle.com/datasets/spscientist/mental-illness>).

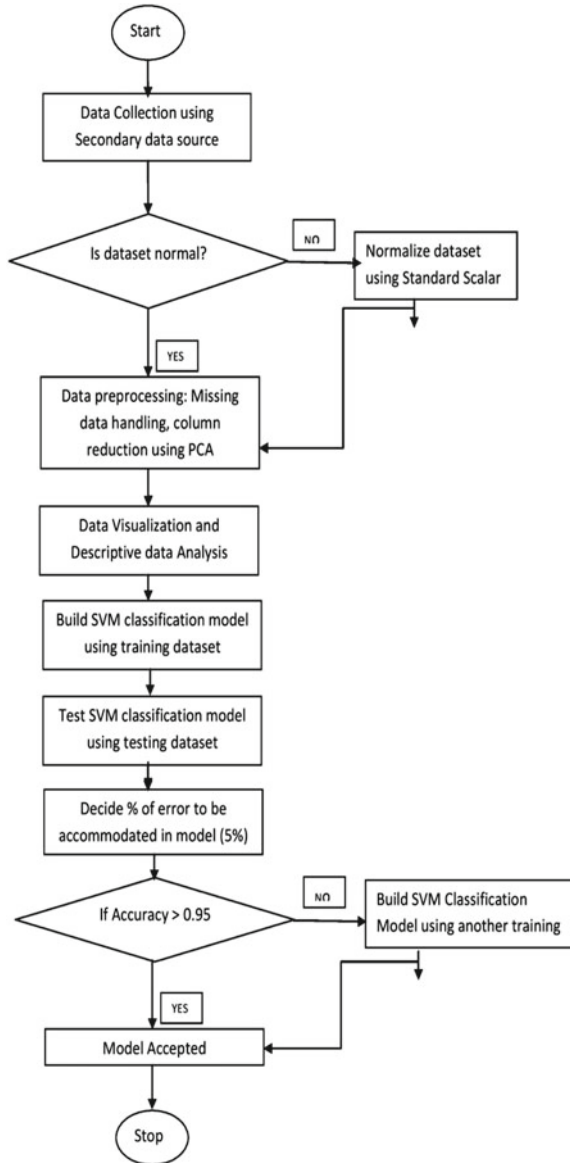
In the next step, authors check the normality of the dataset and found that the dataset is not following normal distribution. Now, authors normalize the dataset using the standard scalar library of Python which is based on the Z score formula. In the next step, authors apply data preprocessing techniques; missing data handling; and column reduction using principal component analysis method on normalized dataset. In the result of this step, authors get clean data with three columns out of nine columns from the original dataset. Now, authors start the process of building a classification model on a new dataset. This requires dividing the dataset in training and testing. Authors take 70% of total dataset as training and 30% of total dataset as testing with randomness. Now, authors start building SVM classification methods using training datasets. The build model is tested with the help of testing dataset, which gives model accuracy, precision, and recall as result. If the results of the model are greater than given threshold 0.95, model accepted else another training data is generated to build another model to be acceptable.

5 Results and Discussion

Machine learning is a subset of artificial intelligence. In machine learning, algorithms learn from experiences based on the dataset provided in the algorithm. There are three types of algorithms used in model construction: supervised, unsupervised, and reinforcement. In this paper, authors used supervised learning algorithms for model construction and model usage. Model construction involves describing the set of predefined classes. Training dataset is used for classification model construction. Model usage involves classifying future data or identifying unknown objects of the dataset. In model usage, authors find the accuracy of the model using a testing dataset. If the model accuracy is acceptable, authors use this model to classify future values.

There are various classification algorithms available like Decision Tree, Bayesian classifier, random forest, K-Nearest Neighbor (KNN), support vector machine (SVM), etc. In this paper, the authors used the SVM classification model because of higher accuracy than other algorithms. SVM is a supervised learning algorithm

Fig. 8 Proposed work flowchart



used for classification, regression, and outlier detection. SVM is useful where data have high dimensions and sample size is less than dimension size. In most of the applications, SVM provides better accuracy than other classification algorithms [19].

In this paper, authors try to build classification model using four classification algorithms which are Decision Tree, Naïve Bayes, KNN, and SVM. In find the results of all the mentioned algorithms, authors follows the steps defined in Fig. 8.

Four parameters are chosen to evaluate the model that is accuracy, precision, recall, and $F1$ -score. These parameters values are calculated with the help of confusion matrix. Confusion matrix is a matrix which stores actual label available in testing data and predicted label defined by model. The sample confusion matrix for three labels is shown in Table 1. There are four quadrants in confusion matrix, which are

TP—True Positive—Actual value of anxiety level is Low and model also predicted Low.

TN—True Negative—Actual value of anxiety level is medium and model also Medium.

FP—False Positive—Actual value of anxiety level is Low and model predicted Medium.

FN—False Negative—Actual value of anxiety level is Medium and model predicted High.

To calculate the confusion matrix for mentioned classification algorithms, authors develop a Python code available in github repository <https://github.com/research2006/Mental-Illness.git>. The results of the mentioned four algorithms are shown in Table 2.

Table 2 shows the results of defined classification algorithms in terms of accuracy, precision, recall, and $f1$ -score. You can see in the table that all four parameters of model evaluation are approximately same for Decision Tree and Naïve Bayes algorithm. The result of KNN algorithm is better than above two algorithms but not fulfills the threshold criteria. The evaluation parameters of SVM algorithm are best compared to DT, NB, and KNN and also fulfill the threshold criteria. The result of the SVM shows that model predicts 95% results accurate that there is only 5% error in the model. These errors can also be reduced with help of ensemble learning techniques like bagging and boosting.

Table 1 Sample confusion matrix for three labels

	Predicted value			
	Anxiety level	Low	Medium	High
Actual value	Low	TP	FN	FN
	Medium	FP	TN	FN
	High	FP	FN	TN

Table 2 Model evaluation and comparison

Algorithm	Accuracy	Precision	Recall	$F1$ -score
Decision Tree (DT)	0.90058	0.89372	0.90058	0.894751
Naive Bayes (NB)	0.90058	0.88996	0.90058	0.891586
KNN	0.923976	0.91788	0.92397	0.918682
SVM	0.953125	0.95271	0.95312	0.950538

Fig. 9 Confusion matrix for SVM classification

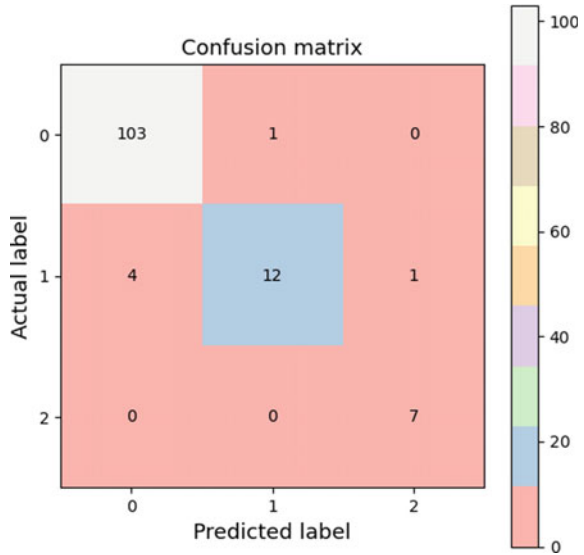


Figure 9 depicts the confusion matrix of SVM model. In this matrix, you can see that TP cases are 103 and TN cases are 19; also, total cases are 128. So according to formula,

$$\begin{aligned}
 \text{Accuracy} &= (\text{TP} + \text{TN})/\text{Total cases i.e. } 122/128 = 0.953125, \\
 \text{Precision} &= \text{TP}/(\text{TP} + \text{FP}) \text{ i.e. } 103/(103 + 4) = 0.96261, \\
 \text{Recall} &= \text{TP}/(\text{TP} + \text{FN}) \text{ i.e. } 103/(103 + 2) = 0.98095, \\
 F1\text{-Score} &= (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \\
 &= 1.888544/1.94356 = 0.97169.
 \end{aligned}$$

These calculations are done using weighted average method available in precision, recall, and *f1*-score function in sklearn package of Python to get the results shown in Table 2.

6 Conclusion

Mental health is very important for any person. But in society, we mostly focus on physical health not mental health. With this reason, we see in society that many people face mental issues like anxiety, fear, lowliness, etc. In this paper, authors try to build a classification model which is able to classify the level of anxiety in humans based on their demographics, media usage, and physiological factors. As a result of this study, authors can conclude that there is a positive correlation between anxiety

level and media usage. Authors also perform gender-wise analysis and found that high anxiety level is more in female as compared to male. The model build by authors performs well with accuracy of approximately 95% which justifies that there is only 5% error in the classification done by model. Authors also compare the evaluation results of various classification algorithms and found that SVM gives better results for given dataset. The results of the study is useful for early detection and prediction the anxiety level of human on the basis of their demographics, social media usage and physiological factors.

References

1. Tripathi M et al (2018) Effect of social media on human health. *Virol Immunol J* 2(2):1–4
2. Schurgin G, Clarke-Pearson K (2011) Clinical report—the impact of social media on children, adolescents, and families. *Pediatrics* 127(4):800–804
3. Auxier B, Anderson M (2021) Social media use in 2021. *Pew Res Center* 1:1–4
4. Keles B, McCrae N, Grealish A (2020) A systematic review: the influence of social media on depression, anxiety and psychological distress in adolescents. *Int J Adolesc Youth* 25(1):79–93
5. Zhuravskaya E, Petrova M, Enikolopov R (2020) Political effects of the internet and social media. *Ann Rev Econ* 12:415–438
6. Allcott H et al (2020) The welfare effects of social media. *Am Econ Rev* 110(3):629–676
7. Wang K, Cheng L, Yong B (2020) Spectral-similarity-based kernel of SVM for hyperspectral image classification. *Remote Sens* 12(13):2154
8. Bashir H, Bhat SA (2017) Effects of social media on mental health: a review. *Int J Indian Psychol* 4(3):125–131
9. Valkenburg PM (2017) Understanding self-effects in social media. *Hum Commun Res* 43(4):477–490
10. Olanrewaju A-ST et al (2020) Social media and entrepreneurship research: a literature review. *Int J Inf Manage* 50:90–110
11. Anderson M, Jiang J (2018) Teens, social media & technology 2018. *Pew Research Center* 31(2018):1673–1689
12. Abbas J et al (2021) The role of social media in the advent of COVID-19 pandemic: crisis management, mental health challenges and implications. *Risk Manage Healthcare Policy*: 1917–1932
13. Meier A, Reinecke L (2021) Computer-mediated communication, social media, and mental health: a conceptual and empirical meta-review. *Commun Res* 48(8):1182–1209
14. Akram W, Kumar R (2017) A study on positive and negative effects of social media on society. *Int J Comput Sci Eng* 5(10):351–354
15. Coyne SM et al (2020) Does time spent using social media impact mental health? An eight year longitudinal study. *Comput Human Behav* 104:106160
16. Braghieri L, Levy R, Makarin A (2022) Social media and mental health. *Am Econ Rev* 112(11):3660–3693
17. Thurnhofer-Hemsi K et al (2020) Radial basis function kernel optimization for support vector machine classifiers. *arXiv preprint arXiv:2007.08233*
18. Upadhyay PK, Nagpal C (2020) Wavelet based performance analysis of SVM and RBF kernel for classifying stress conditions of sleep EEG. *Sci Technol* 23(3):292–310
19. Cortes C, Vapnik V (1995) Support-vector networks (PDF). *Mach Learn* 20(3):273–297. *CiteSeerX* 10.1.1.15.9362. <https://doi.org/10.1007/BF00994018.S2CID206787478>
20. Wang Y, Yu W, Fang Z (2020) Multiple kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information. *Remote Sens* 12(1):120

21. Verma S, Patel K (2017) Association between shopping habit and demographics of m-commerce user's in India using two way ANOVA. 2017 2nd international conference for convergence in technology (I2CT). IEEE
22. Verma S, Patel K (2018) Impact of consumer gender on expenditure done in mobile shopping using test of independence. In: Information and communication technology for sustainable development: proceedings of ICT4SD 2016, vol 1. Springer, Singapore
23. Verma S (2016) Deciding admission criteria for master of computer applications program in India using chi-square test. In: Proceedings of the second international conference on information and communication technology for competitive strategies
24. Valkenburg PM, Peter J, Walther JB (2016) Media effects: theory and research. *Annu Rev Psychol* 67:315–338

Empirical Analysis of Depression Detection Using Deep Learning on Twitter



Arunima Jaiswal, Payal Porwal, Anushka Singh, Pooja Kumari, Priyadeep Bhalla, and Nitin Sachdeva

Abstract People spend a major portion of their time from their daily routine over social media platforms. Many researchers have previously attempted to examine the mental status of the users through their thoughts and opinions in the form of text all over social media. People who are active on social media tend to consume both positive and negative content which may hamper their mental health subconsciously. This gives rise to increased mental health problems as people are consistently consuming a lot of negative content available publicly. The motivation behind this research is early detection of depression, which is the most common mental health condition and second top reason for death worldwide, so that people can be aware and take timely action for their mental health before it deteriorates inexorably. Text classification is an application of natural language processing. This study uses a dataset consisting of thoughts in textual form, collected from Twitter posts or ‘Tweets’, that were posted by the people over the platform, and classify them as depressed or non-depressed. The dataset used is publicly available on Kaggle named ‘Depression Detection’. The study uses four combined models that are BERT + BiGRU, BERT + BiLSTM, BiGRU, and BiLSTM and compares them. The accuracies of the following models are as follows—99.71%, 99.52%, 99.58%, and 98.83%, respectively. It is observed that

A. Jaiswal · P. Porwal (✉) · A. Singh · P. Kumari · P. Bhalla
Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University
for Women, New Delhi, India
e-mail: payal036btcse19@igdtuw.ac.in

A. Jaiswal
e-mail: arunimajaiswal@igdtuw.ac.in

A. Singh
e-mail: anushka067btcse19@igdtuw.ac.in

P. Kumari
e-mail: pooja063btcse19@igdtuw.ac.in

P. Bhalla
e-mail: priyadeep070btcse19@igdtuw.ac.in

N. Sachdeva
Department, Galgotias College of Engineering and Technology, Greater Noida, India
e-mail: nitin.sachdeva@galgotiacollege.edu

by using the novel BERT + BiGRU model the accuracy achieved is 99.71% which is highest, though marginally, among other proposed models in detecting whether a person is depressed or not by reading social media texts as data input. As depression is the second most common reason for death all across the world, this study attempts to benefit the society by helping in timely detection of depression among users by analysing their thoughts that they continuously express over social media. Sometimes, users themselves are not aware of their mental health and continue to suffer. Also, many people hesitate consulting a psychiatrist. This study can be proved to be a first and the foremost step in analysing the mental health condition of people which may save someone's life by timely detection.

Keywords Depression detection · BERT · Deep learning · BiLSTM · BiGRU · Neural network · Machine learning · Depressed text

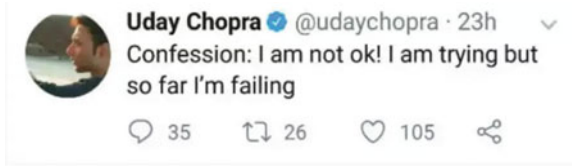
1 Introduction

The Internet came into use from 1 January 1983. Since then, it's been 40 years, and major advancements have been done over the Internet which includes the creation of an infinite number of websites and web applications all over the world. These websites and web applications enable the user to interact with any person residing in any corner of the world all around the globe. Social media is a major domain which has overtaken the market nowadays. This includes many websites such as Facebook, Instagram, Twitter, and Reddit through which people express their opinions and thoughts about a person, situation, politics, etc., including both negative and positive thoughts which are flooding all over these websites. These thoughts and opinions are many times a reflection of people's intrusive thoughts that they are able to express through these social media platforms knowingly or unknowingly. Statistics indicate that in 2021, over 4.26 billion people were using social media in any form, and this number is expected to increase to 6 billion by 2027. The thoughts and opinions that are shared by people over these platforms when accessed carefully through studies can be helpful in determining whether people are suffering from any kind of mental illness or not. It becomes easy to categorise people as depressed or not depressed. Depression is a worldwide concern and the second most reason for death [1].

The daily usage of social media accounts to around 2.5 h per day per person which is roughly up by 2 per cent in comparison with the daily average reported at the start of year 2022 [2]. Sometimes people are themselves not aware about their mental health. They may be suffering from some serious chronic mental illness. Many people may hesitate in consulting a psychiatrist for diagnosing themselves. This brings an opportunity for analysing such texts/graphics which may help people in detecting if they are suffering from depression or not through their thinking patterns that they express over social media platforms.

By analysing the posts over various social media platforms, experts and researchers have found some common patterns of words that associate with the

Fig. 1 What depressive text looks like



person describing them as depressed or not. The words associated with depression are often related to emotions, whereas those associated with loneliness are linked to cognition [3]. There are some particular words that people use to represent their feelings and the contents. For example, a sentence consisting of negative words like ‘sad’, ‘meaningless’, ‘worthless’ indicate that the user may be depressed.

This is what depressive text may look like (Fig. 1).

This study consists of a review of previous studies conducted on depression detection from social media text (and other modalities as well like audio, video, etc.) and presents a comparative analysis between different embedding and natural language processing techniques to be applied with a deep learning model, like BiGRU and BiLSTM. The dataset used is a benchmark dataset which is publicly available on Kaggle. Integrating such efficient techniques to build a model which provides better accuracy or proposing a novel model is the primary objective of this study (Fig. 2).

The methodology used in the research paper is a stepwise procedure. The first step includes the collection of dataset from a known source for our model. In this study, we have used publicly available Twitter dataset. After the collection of the dataset, the next step is the cleaning of the dataset before preprocessing. After that, it was stemmed and lemmatized, and then various NLP techniques were applied to it. Finally, the classification of data is done as depressed or non-depressed.

In this study, depression detection has been carried out on textual social media data, using various deep learning models, and those models have been compared on the basis of their efficacy. The various methodologies used are as follows:

- (i) BERT + BiLSTM
- (ii) BERT + BiGRU

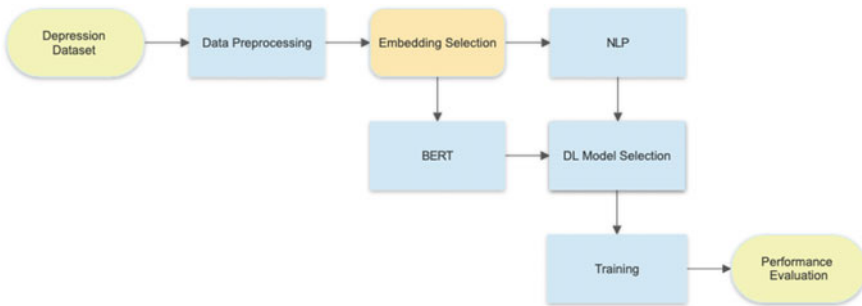


Fig. 2 Experimentation procedure diagram

- (iii) BiLSTM
- (iv) BiGRU.

The study's main contribution is detection of depression in a user who is publicly expressing his/her thoughts and opinions over social media. The study includes classification of a user as depressed or non-depressed using various NLP techniques, comparing them and observing the results achieved in each case. Here, it can be observed that the highest accuracy achieved is 99.71% by combination of the BERT + BiGRU model. The novel BERT + BiGRU approach suggested in this study outperforms the existing models in the field of depression detection on textual data.

2 Related Work

There have been a large number of studies performed so far on identifying depression through social media networks in the past, but the majority of them have been based on classical feature engineering techniques [3]. During these review of literature, it has been discovered that lexicons used for linguistic inquiry word counts, or what is commonly referred to as the linguistic inquiry word count (LIWC) lexicon, have been widely used as a feature engineering tool and technique for detecting and identifying lexical features. The psychological traits described in this lexicon have been classified into at least 32 types. In a recent study, researchers tried to improve the performance of the models by using vector space representations, as well as a convolutional neural network model with a bidirectional mechanism for identifying and describing the health hazards in the posts [5].

The related work section below comprises the studies by previous researchers which focused on mental health detection in humans through social media and related websites. Reference has been taken from the below mentioned related work.

Table 1 Literature Review

Study	Dataset	Model proposed	Performance metric	Result (%)
Yu [4]	THUC news dataset	BERT-BiGRU	Accuracy, precision, <i>F1</i> score, recall	94.6, 95, 95, 95
Ahmed [5]	LIWC	BiLSTM	Accuracy, precision	93, 89
Ghosh [6]	Bangla social media text dataset	BiLSTM	Accuracy, sensitivity, specificity	94.3, 92.63, 95.12
Shaw [7]	Twitter dataset	Multichannel CNN, CNN, GRU, Capsule network, and BERT	Accuracy, precision, recall, <i>F1</i> score	97.5, 96.8, 97.5, 97.2

(continued)

(continued)

Study	Dataset	Model proposed	Performance metric	Result (%)
Park [8]	Twitter dataset (Reddit)	BiLSTM, BERT, CNN	Precision, recall, <i>F1</i> score	0.9879, 0.9945, 0.9892
Zeberga [9]	Reddit, Twitter	BiLSTM, BERT	Accuracy	98
Yadav [10]	Distress analysis interview Corpus-Wizard-of-Oz interviews dataset	BGRU	<i>F1</i> score	0.92
Ameer [11]	Reddit	BERT, LSTM, BiLSTM, GRU, BiGRU	Accuracy, <i>F1</i> score	0.83, 0.83
Nadeem [12]	Twitter tweets dataset	LSTM, GRU	Accuracy1, Accuracy2, <i>F1</i> score	97.4, 82.45, 94.4
Islam [13]	Social media text	LSTM, GRU, CNN-BiLSTM	Accuracy	88.59
Singh [14]	SD-Sdford-09, DD-Red-14, DD-Kgg-22, SD-Twi-23	BiLSTM, GRU, BERT, ALBERT	Accuracy, precision, recall, AUC, mean increase	91.92, 92.04, 91.35, 0.9024, 4.49
Triantafyllopoulos [15]	Reddit posts on the Pirina dataset, Reddit RSDD dataset	BERT, GRU	Accuracy, <i>F1</i> score, precision, recall	93.87, 95.65, 95.16, 96.14
Arif [11]	Reddit posts	RoBERTa	Accuracy, <i>F1</i> score	83, 83
Dheeraj [16]	WebMD about 2086 and HealthTap about 5328 questions (an online medical healthcare platform)	MHA-BCNN	Accuracy	89
Orabi [17]	CLPsych2015 and bell let's talk	CNN and RNN	CNN accuracy, RNN accuracy	83.2, 93.4
Deshpande [18]	Collection of tweets using twitter API	Naive Bayes Classifier and support vector machine	Naive Bayes accuracy, <i>F1</i> score	83, 83.2
Amanat [19]	Tweets scraped dataset	Deep RNN LSTM	Accuracy	99
Cong [20]	Reddit self-reported depression diagnosis (RSDD)	X-A-BiLSTM	Precision, <i>F1</i> score, recall	69, 60, 53

(continued)

(continued)

Study	Dataset	Model proposed	Performance metric	Result (%)
Burdisso [21]	CLEF 2017 and eRisk pilot task	SS3	<i>F1</i> score, π , precision	0.54, 0.44, 0.69
Vasha [22]	Self-made dataset, collected from Facebook texts, comments or posts of a single line	SVM	Precision, <i>F1</i> score, accuracy	77, 78, 80
Ashraf et al. [23]	Reddit dataset	BERT, BiGRU, IMFine	Precision, recall, <i>F1</i> score	0.95, 0.93, 0.94
Kabir et al. [24]	Dataset collected from social media	LSTM, GRU, kernel support vector machine (SVM), random forest, logistic regression K-nearest neighbour (KNN), and complement naive Bayes (NB)	Accuracy1, Accuracy2	81, 78
Singh et al. [25]	Twitter dataset	Naive Bayes, logistic regression, SVM and ensemble machine learning models, random forest and XGBoost, LSTM and GRU	<i>F1</i> score	0.92
Chowdhury et al. [26]	Social media text	LSTM, GRU, CNN	Accuracy	90
Hasan et al. [27]	Twitter dataset	BERT	Accuracy	97
Hadjiharalambous et al. [28]	News reports dataset	BERT, BiGRU, BiLSTM	Accuracy	80–90

The study proposed in this research paper is inspired by the previous related work of the authors mentioned above. The related work has proven to be an excellent source of knowledge and information throughout the journey of this study and research. After going through all the related research work, the model achieving highest accuracy was finalised and implemented. This source of motivation and information proved to be of great help which further steered this study's efforts in the right direction and inspired better logic on how to improve the accuracy of existing research work.

3 Proposed Methodology

The study proposes a comparative analysis between various deep learning methodologies, with respect to their efficacy in depression detection of textual social media data.

The proposed work in this research paper includes the steps as mentioned further—the first step is to collect the necessary dataset for the depression related texts over social media. A publicly available dataset from Kaggle, named ‘Depression Detection’ was used. This dataset is a collection of tweets from Twitter social media platform. The dataset was cleaned before it was preprocessed, followed by stemming, lemmatization, and other NLP techniques were applied to it. The dataset was then classified into depressed and not depressed, and labelled as 0 and 1 to identify the class. The dataset was then ready to be used for the study. In order to perform the tasks in the procedure, the set-up of software environment and machine specification used for this research work is as follows. Silicone M1 Pro 8-core Neural Engine CPU, 16 GB RAM, 14-core M1 Pro GPU, Google Colaboratory IDE, 3.7.15 version of Python, 2.9.2 version of Tensorflow, 2.9.0 version of Tensorflow-text, 2.9.0 version of Keras are used.

3.1 Dataset Preprocessing

In the data preprocessing phase, the data is being processed to remove unnecessary text, which can hinder the classification process, and make it free from outliers which can skew the model or overtrain it for unnecessary features. To clean the tweet dataset, twitter ids, https link tags, and website related tags were removed. BeautifulSoup was used which removes the tags added at the time of scraping into actual punctuations and spaces if that’s what they mean. Also, all the negation words which are being shortened are converted into full words like ‘aren’t’ into ‘are not’. These changes make the text more clean and understandable by the model, which helps in attaining high accuracy.

Table 2 Dataset Description Label

Label	Count
Non-depressed	20,952
Depressed	18,190

For the models, where BERT encoding was used, each sentence was tokenized and passed through the BERT preprocessor, whereas for the models where BERT encoding was not used, one hot encoding was used after application of which, the input was passed to an embedding layer.

3.2 *Feature Extraction Techniques*

Feature extraction is used to build models which take as input one or more documents and classifies them by their content. BERT is a pre-trained model which is used to do the same. In feature extraction, BERT's output is taken along with implicit representation of all or some of BERT's layers, and then a different model is trained on those features to forge ahead of building the model.

Bidirectional Encoder Representation from transformer is a framework that is used in natural language processing (NLP). It is a deep learning model, in which every input element is connected to every output element of the dataset in such a way that based upon their connection the weightages are calculated dynamically. This process in NLP is known as Attention. From the BERT model, the text is read from both the direction, i.e. left-to-right and right-to-left direction at the same time. In the BERT model, two sizes of model have been proposed: BERT base and BERT Large, where BERT base consists of 12 layers of encoder block, 12 head attention, 110 million parameters; it is usually comparable in size to the OpenAI transformer in order to compare performance, whereas BERT Large consists of 24 layers of encoder block, 16 head attention, 340 million parameters.

BERT is a pre-trained transformer-based language model that generates contextualised embeddings for the input text. In this study, 'bert_multi_cased_L-12_H-768_A-12', a pre-trained saved model for text embedding was loaded.

3.3 *Why BERT Over Traditional NLP?*

Traditional NLP techniques aim to comprehend human language as it is. In BERT's case, this means predicting a word. To be able to do this, models are trained on large sized labelled training data. This onerous task of data labelling falls on linguists.

BERT uses a process called Transfer Learning to improve on this. It reduces this human effort as it was pre-trained on unlabelled plain text corpus. It continues to do so unsupervised and improves as it is applied in practical domains. Subsequently, BERT can adapt to the perpetually increasing searchable queries and be customised for a user's specifications.

3.4 *GRU and BiGRU*

GRU or Gated Recurrent Unit is a type of RNN, which is an alternative to LSTM networks. GRU, like LSTM, can process sequential data like time series, speech, and text data.

GRU has been introduced to solve the vanishing gradient problem. To do so, it uses 'update gate' and 'reset gate', which basically decide what information is passed

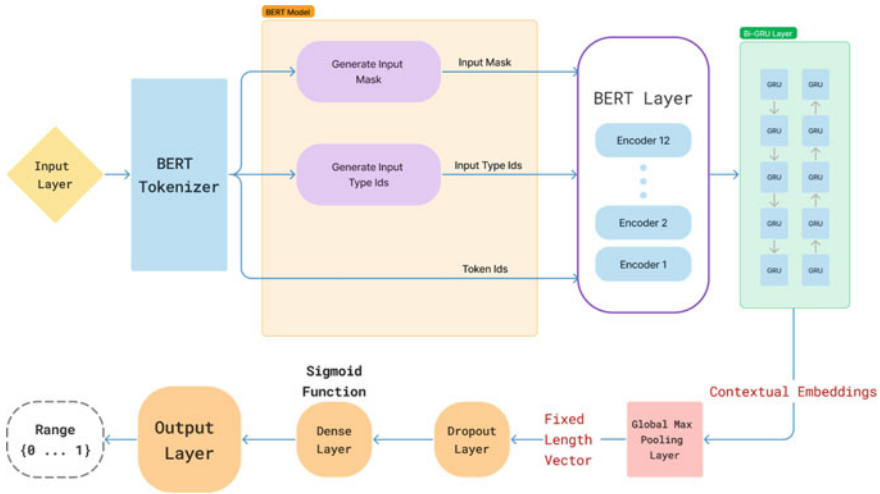


Fig. 3 Model architecture

forward. GRU selectively updates the hidden state of the network at each time step. The reset gate is responsible for determining how much of the previous hidden state will be forgotten, and the update gate determines how much of the new input will be used to update the hidden state. The GRU output is based on the updated hidden state. Bidirectional GRU or BiGRU is a model that consists of 2 GRUs. One of the GRUs takes the input in forward direction, while the other takes it in backward direction. It is a bidirectional RNN with only input and forget gates (Fig. 3).

In this study, depression detection has been carried out on textual social media data, using various deep learning models, and those models have been compared on the basis of their efficacy. The various methodologies used are as follows:

- (i) BERT + BiLSTM
- (ii) BERT + BiGRU
- (iii) BiLSTM
- (iv) BiGRU.

4 Results and Comparative Analysis

4.1 Dataset Description

Depression Detection is a field which has recently been given much attention. The dataset that is being used named ‘Depression Detection’ is publicly available on Kaggle website. The dataset used should be properly preprocessed in order to achieve better results and only then can the quality and efficacy of our model be of the highest order. The dataset consists of 39,142 tweets, classified as depressed or non-depressed

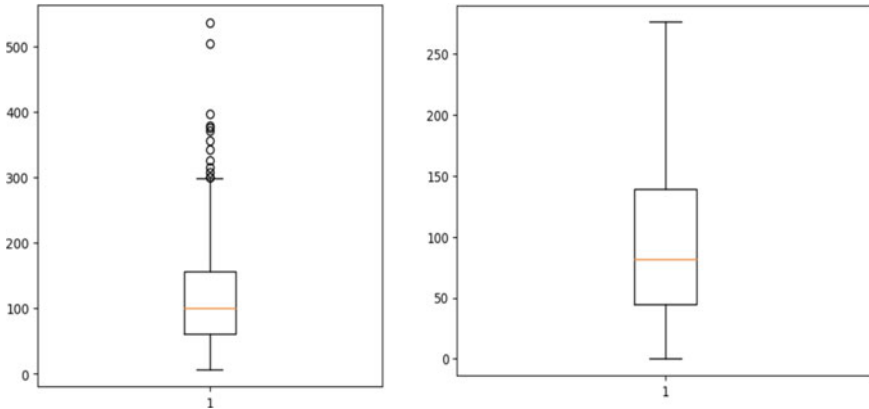


Fig. 4 Data cleaning

tweets. In Fig. 4, the length of sentences in the dataset has been plotted before and after preprocessing, in the form of boxplot. It can be observed that the length of sentences decreases after preprocessing.

4.2 Evaluation Metrics

The proposed models were evaluated on the basis of accuracy, loss, precision, recall, and F1 score. Various evaluation metrics are used to measure the efficacy of machine learning and deep learning models that have been developed, to compare them to other models, and so that they may be improved.

Accuracy is an evaluation metric denoted as the number of correctly predicted labels divided by the total number of possible outcomes predicted by the model. Mathematically, it is represented as

$$\text{Accuracy} = \frac{T_n + T_p}{T_n + T_p + F_n + F_p}.$$

Another evaluation metric is precision, which refers to the ratio of true positives given by the model to the sum of true positives and false positives.

$$\text{Precision} = \frac{T_p}{T_p + F_p}.$$

Recall is an evaluation metric which shows how accurately the model predicted true positives out of all positive cases. This is obtained by deriving the ratio of true positives to sum of true positives and false negatives.

$$\text{Recall} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}}$$

The value of *F1*-Measure evaluates the harmonisation of two of the factors based on recall and precision, because they measure distinct features.

$$F1 \text{ Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Here, the data is trained and cross-validation technique is used by taking validation_split = 0.25, Hence, the comparison of accuracy between training data and validation data was plotted against epochs during model training process for each model. Similar graph was plotted for loss as well, as shown in Figs. 5, 6, 7, and 8.

The ROC curve for the highest accuracy model which is BERT + BiLSTM is also shown in Fig. 9. The Receiver Operator Characteristic (ROC) curve is an evaluation

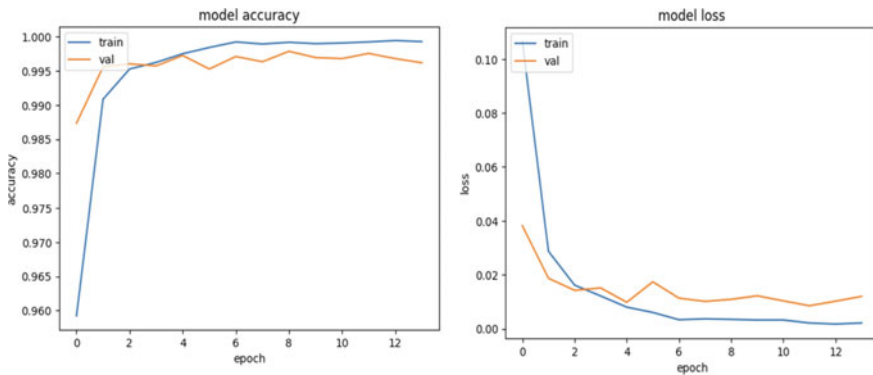


Fig. 5 BERT + BiGRU

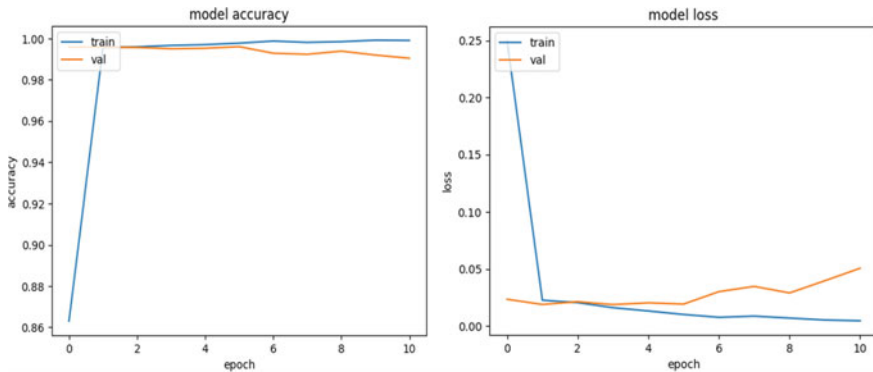


Fig. 6 BiGRU

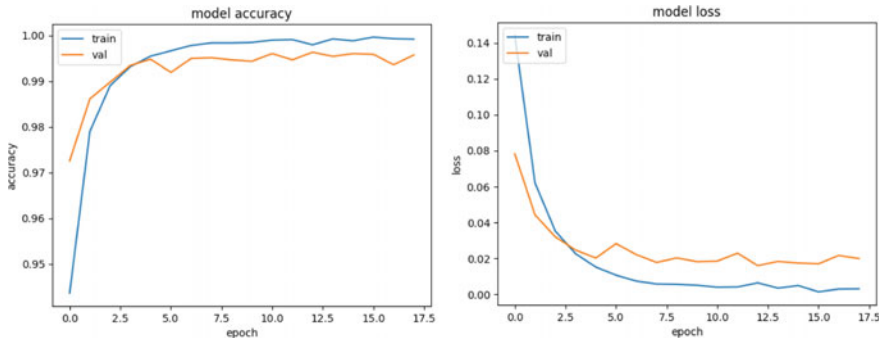


Fig. 7 BERT + BiLSTM

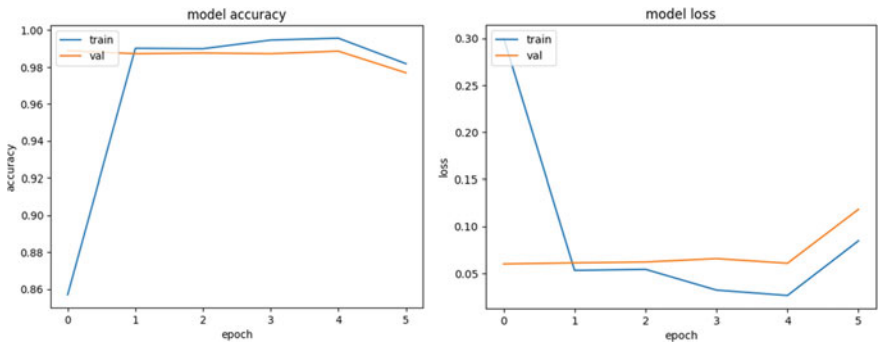


Fig. 8 BiLSTM

metric for binary classification problems predominantly; however, it can be extended to multiclass problems as well. ROC curve is a probability curve that plots the True Positive Rate (TPR) against False Positive Rate (FPR) at various threshold values. The ROC curve principally separates the ‘signal’, i.e. true data from the ‘noise’, i.e. outlier data. It indicates the performance of a classification model at all classification thresholds.

Figure 10 visualises the accuracy achieved by the models that were used in this study, namely BiGRU and BiLSTM. It can be observed that the BiGRU model acquired an accuracy of 99.71 and 99.58% for BERT and NLP embedding, while the BiLSTM achieved an accuracy of 99.52 and 98.83% for BERT and NLP embedding. It can be observed that the accuracy of the BiGRU model is slightly higher than that of BiLSTM.

4.2.1 Comparative Analysis of Classification Report for All the Models

See Fig. 11.

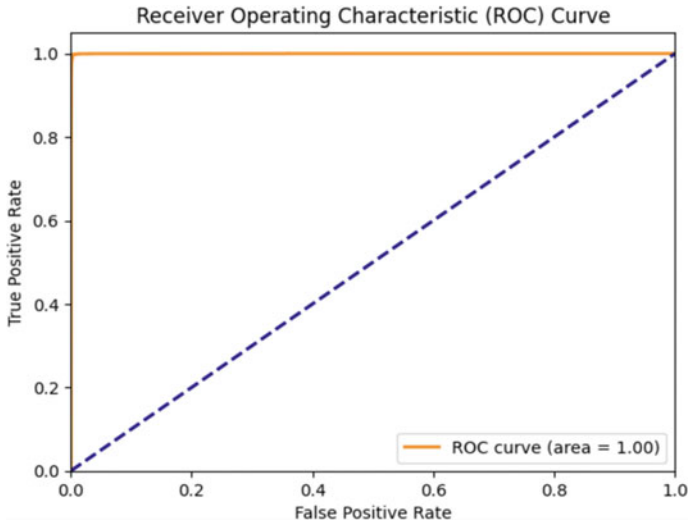


Fig. 9 ROC curve for BERT + BiGRU

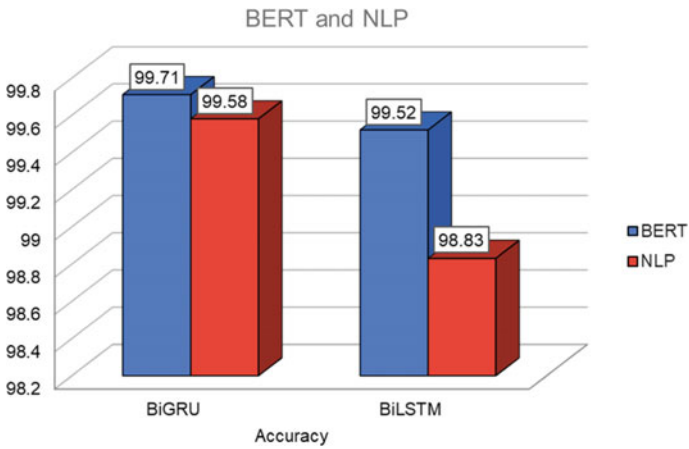


Fig. 10 Accuracy

5 Limitations of the Proposed Study

This study uses data present on twitter which was publicly available. Then, various natural language processing techniques were applied on the dataset such as—BERT + BiLSTM, BERT + BiGRU, BiLSTM, and BiGRU. The highest achieved accuracy is 99.71% using the BERT + BiGRU model. One of the major limitations of the study is that the depression detection can only be predicted on textual data as of now. Audio, visual, emojis, and audio + visual combined input cannot be investigated by

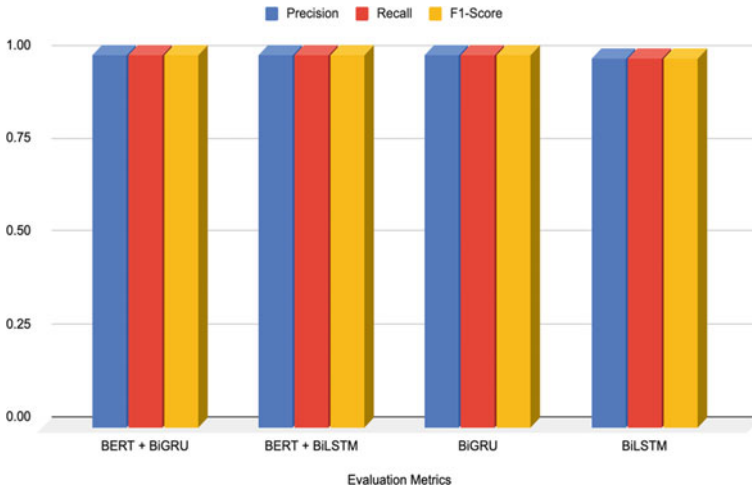


Fig. 11 Visual comparison of various models on different evaluation metrics

this study. However, this study can be further optimised for audio and visual inputs as well so that the combined model can be used for any kind of input (textual, audio, visual). Also, this study focuses on detection of depression; this can be considered as a limitation as there are a wide variety of emotions (both negative and positive) which a human goes through. This study does not help in detection of other negative emotions of a user; thus, the scope of research is limited. Also, an improvement on reliability of the result can be done as there might be a possibility that a user may or may not be depressed, and the model predicts a contrasting result. Hence, a more reliable result may be needed from a mental health professional. The study also relies on the assumption that the user is expressing himself or herself clearly and honestly, which is practically, not often the case. Therefore, accuracy of the result can be further verified, and the research study can be extended further.

6 Conclusion

In today's online world, social media provides users a convenient way for expressing their emotion. Through textual data from online social media data, it is possible to analyse users' feelings and sentiments through different deep learning techniques [3]. For this task of detecting depression through textual data, a publicly available Kaggle dataset, namely Depression Detection, has been used. This dataset contains two columns, Tweets text and labels consisting of 0 or 1, 0 denoting not depressed and 1 denotes depressed. This dataset was cleaned before its preprocessing, then it was stemmed and lemmatized, and further different NLP techniques were applied to it. In the preprocessing phase, the unnecessary texts such as twitter ids, hypertext

link tags, and website related tags were removed from the tweet dataset, also the negotiation words were converted to full words such as ‘aren’t’ into ‘are not’. This study proposes four combined models that are BERT + BiGRU, BERT + BiLSTM, BiGRU, and BiLSTM, and their accuracies of the models are as follows—99.71%, 99.52%, 99.58%, and 98.83%, respectively. It is seen that by using the novel BERT + BiGRU model the accuracy achieved is 99.71%, which is highest among other proposed models in detecting whether a person is depressed or not by reading social media texts as data input.

7 Future Scope

By using different techniques, a varied range of emotions may be detected as well. A dataset of larger size can also validate the efficacy of the techniques used. The study proposed in this research paper can be utilised as a base model for further extending the studies for audio and visual modalities input of the data.

An interactive UI may also be designed such that people can diagnose themselves for depression, although professional validation may still be required. The process of detecting depression can be automated within the social media platforms by automatically reading thoughts and opinions on Twitter, etc., so that users will automatically get a warning if their mental health is at risk. It can prove to be of great help as users will get a good idea about their current mental health.

References

1. Priya A, Garg S, Tigga NP (2020) Predicting anxiety, depression and stress in modern life using machine learning algorithms. *Procedia Computer Science* 167:1258–1267
2. Digital 2023 deep-dive: how much time do we spend on ... Data Reportal. <https://datareportal.com/reports/digital-2023-deep-dive-time-spent-on-social-media>
3. Racherla AS, Sahu R, Bhattacharjee V (2022) A graph convolutional network based framework for mental stress prediction. In: *Artificial intelligence, machine learning, and mental health in pandemics*. Academic Press, pp 73–92
4. Yu Q, Wang Z, Jiang K (2021) Research on text classification based on bert-bigru model. *J Phys Conf Ser* 1746(1):012019
5. Ahmad H, Asghar MZ, Alotaibi FM, Hameed IA (2020) Applying deep learning technique for depression classification in social media text. *J Med Imag Health Inform* 10(10):2446–2451
6. Ghosh T, Al Banna MH, Al Nahian MJ, Uddin MN, Kaiser MS, Mahmud M (2023) An attention-based hybrid architecture with explainability for depressive social media text detection in Bangla. *Expert Syst Appl* 213:119007
7. Shaw B, Saha S, Mishra SK, Ghosh A (2022) Investigations in psychological stress detection from social media text using deep architectures. In: *2022 26th international conference on pattern recognition (ICPR)*. IEEE, pp 1614–1620
8. Cha J, Kim S, Park E (2022) A lexicon-based approach to examine depression detection in social media: the case of Twitter and university community. *Human Soc Sci Commun* 9(1):1–10

9. Zeberga K, Attique M, Shah B, Ali F, Jembre YZ, Chung TS (2022) A novel text mining approach for mental health prediction using Bi-LSTM and BERT model. In: Computational intelligence and neuroscience
10. Yadav U, Sharma AK (2023) A novel automated depression detection technique using text transcript. *Int J Imaging Syst Technol* 33(1):108–122
11. Ameer I, Arif M, Sidorov G, Gómez-Adorno H, Gelbukh A (2022) Mental illness classification on social media texts using deep learning and transfer learning. arXiv preprint [arXiv:2207.01012](https://arxiv.org/abs/2207.01012)
12. Nadeem A, Naveed M, Islam Satti M, Afzal H, Ahmad T, Kim KI (2022) Depression detection based on hybrid deep learning SSCL framework using self-attention mechanism: an application to social networking data. *Sensors* 22(24):9775
13. Islam S, Islam MJ, Hasan MM, Ayon SSM, Hasan SS (2022) Bengali social media post sentiment analysis using deep learning and BERT model. In: 2022 IEEE symposium on industrial electronics and applications (ISIEA). IEEE, pp 1–6
14. Singh J, Singh N, Saba L, Suri JS (2023) Attention enabled ensemble deep learning models and its validation for depression detection: a domain adoption paradigm. Available at SSRN 4404870
15. Triantafyllopoulos I, Paraskevopoulos G, Potamianos A (2023) Depression detection in social media posts using affective and social norm features. arXiv preprint [arXiv:2303.14279](https://arxiv.org/abs/2303.14279)
16. Dheeraj K, Ramakrishnu T (2021) Negative emotions detection on online mental-health related patients texts using the deep learning with MHA-BCNN model. *Expert Syst Appl* 182:115265
17. Orabi AH, Buddhitha P, Orabi MH, Inkpen D (2018) Deep learning for depression detection of twitter users. In: Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic, pp 88–97
18. Deshpande M, Rao V (2017) Depression detection using emotion artificial intelligence. In: 2017 international conference on intelligent sustainable systems (ICISS). IEEE, pp 858–862
19. Amanat A, Rizwan M, Javed AR, Abdelhaq M, Alsaqour R, Pandya S, Uddin M (2022) Deep learning for depression detection from textual data. *Electronics* 11(5):676
20. Cong Q, Feng Z, Li F, Xiang Y, Rao G, Tao C (2018) XA-BiLSTM: a deep learning approach for depression detection in imbalanced data. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp 1624–1627
21. Burdisso SG, Errecalde M, Montes-y-Gómez M (2019) A text classification framework for simple and effective early depression detection over social media streams. *Expert Syst Appl* 133:182–197
22. Vasha ZN, Sharma B, Esha IJ, Al Nahian J, Polin JA (2023) Depression detection in social media comments data using machine learning algorithms. *Bull Electr Eng Inform* 12(2):987–996
23. Kamal A., Mohankumar, P., & Singh, V. K. (2022, December). IMFinE: An Integrated BERT-CNN-BiGRU Model for Mental Health Detection in Financial Context on Textual Data. In Proceedings of the 19th International Conference on Natural Language Processing (ICON) (pp. 139-148).
24. Islam MR, Kabir MA, Ahmed A, Kamal ARM, Wang H, Ulhaq A (2018) Depression detection from social network data using machine learning techniques. *Health InfSci Syst* 6:1–12
25. Singh NK, Singh P, Chand S (2022, November) Deep learning based methods for cyberbullying detection on social media. In: 2022 international conference on computing, communication, and intelligent systems (ICCCIS). IEEE, pp 521–525
26. Ghosh T, Chowdhury AAK, Banna MHA, Nahian MJA, Kaiser MS, Mahmud M (2022, October) A hybrid deep learning approach to detect bangla social media hate speech. In: Proceedings of international conference on fourth industrial revolution and beyond 2021. Springer Nature Singapore, Singapore, pp 711–722
27. Essam N, Moussa AM, Elsayed KM, Abdou S, Rashwan M, Khatoun S, Hasan MM, Asif A, Alshamari MA (2021) Location analysis for arabic covid-19 twitter data using enhanced dialect identification models. *Appl Sci* 11(23):11328

28. Hadjiharalambous G, Beisert K, Jose JM (2022) End-to-end hierarchical approach for emotion detection in short texts. In: Responsible data science: select proceedings of ICDSE 2021. Springer Nature Singapore, Singapore, pp 1–12
29. Figure 1 source. <https://images.app.goo.gl/czJx2JFKUMShaLo9A>
30. Manuel et al (2023) Depression Detection, Kaggle, Depression Dataset
31. <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/>, Statistics Data Used

Assessment of Driver Fatigue and Drowsiness Based on Eye Blink Rate



Samarpit Karar and Tirupathiraju Kanumuri

Abstract Drowsy and fatigued driving is a major factor in many traffic accidents. The slow onset of drowsiness or extreme fatigue in a driver can be detected, although it is more difficult to do so than it is to detect closed eyes. We propose a novel yet simple single camera-based real-time computer vision technique for detecting drowsiness and fatigue levels that solely relies on the eye blinking rate estimated from the eye aspect ratio and moving average calculation over a period of 30 s which is updated every 10 s. An alert is generated to stop the user from going into microsleep or caution the user in case of extreme fatigue if the rate of eye blinking falls below a level or is too high respectively that has been scientifically validated in the literature. The existing methods use facial expression-based detection, blink-based detection using Electrooculogram or simple eye aspect ratio-based methods or calculation of blink rate in pixels/seconds or combination of these which only results in detection of drowsiness, while the proposed method uses single camera-based detection calculating blink rate to estimate both—the drowsiness and fatigue levels in blinks/minute; thus, the proposed method is much less complicated in terms of hardware and computation, and results are repeatable over different ambient illuminance conditions.

Keywords Driver fatigue · Drowsiness · Eye blink rate · Eye aspect ratio · Ambient illuminance · Microsleep

1 Introduction

Previously regarded as a luxury, automobiles are now a necessity in every man's life. People often feel fatigued and drowsy after driving for a long period or at unusual hours, yet they continue driving anyhow in order to arrive to their destination as quickly as possible. The rising number of traffic accidents is one of the main problems with the widespread use of automobiles. The primary contributors to the accident scenario are the driver's inattention, alcoholism, and irresponsibility.

S. Karar (✉) · T. Kanumuri
Department of Electrical Engineering, National Institute of Technology, Delhi, India
e-mail: samarpit.karar@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
A. Swaroop et al. (eds.), *Proceedings of Data Analytics and Management*, Lecture Notes in Networks and Systems 787, https://doi.org/10.1007/978-981-99-6550-2_24

311

Each year, hundreds of traffic accidents are caused by driver drowsiness. Although it is very difficult to estimate the precise number of sleep-related incidents, research suggests that driver sleepiness may play a role in about 20% of traffic accidents and about 25% of fatal accidents. Due to the fact that drowsy drivers cannot brake or maneuver to prevent or lessen the impact, these collisions have a 50% greater chance of causing death or serious injury and frequently involve high speeds [1]. Increased reaction time, a crucial component of safe driving, is caused by drowsiness. Additionally, it lowers attentiveness, awareness, and concentration, impairing the capacity to carry out attention-based tasks, affects information processing and the decision-making standards is also impacted. The severe fatigue and/or drowsiness result in frequent blinking, heavy eyelids, or trouble focusing, daydreaming, straying or unconnected thoughts, difficulty recalling the most recent few miles traveled, or failing to notice exits and street signs, recurrent yawning, eye rubbing, difficulty maintaining head elevation, changing lanes often, tailgating, etc. [2, 3].

Drowsy and fatigued driving is known to contribute significantly to hundreds of commercial vehicle accidents each year. The transportation system continues to face problems despite the fact that almost all drivers are affected by this decreased awareness. Driver fatigue and drowsiness is a sociotechnical problem that is related to the decisions a driver takes in the hours before a trip as well as the circumstances and pressures they encounter while performing their daily. In light of this, it is important to take precautions to spot the signs of drowsiness and fatigue while operating commercial vehicles, comprehend the common ways that drivers combat drowsiness, and be aware of the structural obstacles that prevent them from making wise choices regarding maintaining their alertness [4].

The existing methods relying on change in pressure distribution on seat, detection of vehicle deviation and position [5], physiological parameters such as heart rate, electroencephalogram and respiration rate [6, 7], eye closure and yawning detection and their fusion, facial monitoring measures [8–11], measure of position of head [12], change in behavioral aspects of driver [5], etc., are quite complex methods in terms hardware as well as in terms of computation and execution in real time. Moreover, these methods are either measuring the drowsiness level or the fatigue level alone but as such no classification method has been proposed in a single technique. The techniques somewhat similar in broad principle to the proposed technique such as blink-based detection using Electrooculogram, or Simple eye aspect ratio-based methods or calculation of blink rate in pixels/seconds, or Combination of these, etc., [13, 14] have complex computation.

Lower alertness and sleepiness of driver are a dangerous condition which accounts for multiple traffic accidents. To alert the driver in case of drowsy and extreme fatigue condition, a novel technique has been developed, where a camera along with real-time execution of the algorithm employing blink rate detection based on moving average estimation has been used to alert the user of drowsiness and extreme fatigue when the threshold levels established in the literature are crossed, and relevant drowsiness and fatigue alerts are generated.

2 Literature Review

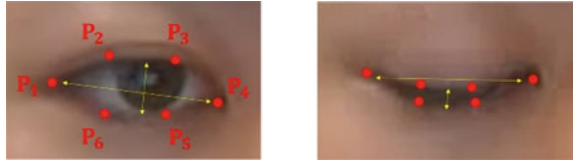
The eye blink refers to the phenomenon, where the eye is momentarily hidden, and the upper and lower lids are touching. Since a change in the blink artifact's shape can be utilized to detect hypovigilance, it is crucial to be able to differentiate between eye blinks and vertical eye movements. The blink behavior is described by the parameters of blink frequency, amplitude, and duration. Although only 2–4 blinks are required physiologically, a relaxed individual blinks roughly 15–20 times each minute. In contrast to an increase in blink frequency, which signals increased vigilance, the blink rate lowers to as little as 3 blinks per minute when completing cognitive tasks. The act of blinking helps to maintain the health of our eyes by keeping them clean and moisturized as well as by giving our eyes and minds a rest. When compared to babies, who only blink twice per minute, adults blink at a constant rate throughout their lives, increasing from roughly 12 to 15 times per minute in adolescence. People who lack sleep or have irregular sleep patterns tend to blink 20–30 times per minute; thus, we may blink twice as frequently when we're weary. When someone enters microsleep mode and briefly nods off, this could be followed by the eyes closing and a severe drop in the blink rate, which would eventually lead to closed eyes without blinks. An excellent predictor of the user's or driver's present degree of attention may therefore be the blink rate [15–18].

The driver tends to exert themselves excessively due to this habit. Long hours spent behind the wheel harm drivers' health. As a result, the drivers are frequently exhausted or nearly so. This causes individuals to experience brief episodes of microsleep, which can last anywhere from a fraction of a second to few tens of seconds and cause the victim to lose consciousness after failing to respond to an environmental stimulus. A study was carried out in an attempt to create a drowsy eyes detection system in which 68 key points as detection approach for facial regions use as key point to evaluate the driver's state [13, 19].

As computer vision technology has improved, smart/intelligent cameras have been created that can detect driver drowsiness and alert them, which helps to prevent accidents when drivers are fatigued. In studies on the detection of driver tiredness based on eye state, a framework was proposed that identifies whether the eye is in a sleepy or non-drowsy state and alert with an alarm when the eye is in a drowsy condition. The Viola-Jones detection algorithm was used to identify the face and eye regions. For the learning phase, features from stacked deep convolution neural networks were collected and employed. A softmax layer in a CNN classifier was then used to determine whether the driver was sleeping or not [20]. A detailed review on driver drowsiness measurement technologies was carried out; the technologies studied were vehicle-based driving behavior, video-based driving behavior, and the driver physiological signals measure-based technologies. It was observed that there are merits and demerits of each method [21, 22].

The two methods used for blink detection are the Electrooculogram (EOG) and the eye aspect ratio-based method. The EOG is derived from the electrical potential captured by electrodes positioned above and below the eye during an eye blink.

Fig. 1 Eye aspect ratio (EAR) [14]



The potential between the two electrodes above and below the eye is influenced by the natural motions of the eyelids during blinking. The wave-form of the measured potential difference and the standard EOG of the blink have a strong connection, which is the basis for blink detection. Eye aspect ratio (EAR) is obtained by estimating Euclidean distance of the corresponding eye coordinates. It is then substituted into the following formula where P1, P2, P3, P4, P5, and P6 are 2D facial landmark locations (refer Fig. 1) [14, 23–25]:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}. \tag{1}$$

3 Methodology

A real-time system is required to avoid countless mishaps due to drowsy or fatigued driver behavioral changes by focusing on driver eye blinks. This work reports study of drowsiness and fatigue levels of the participants of different age groups under different ambient illuminance based on the experimental study under simulated environment and the associated qualitative analysis. The experimental setup is shown in Fig. 2.

A code is written in Python which performs the image capture, calculation of blink rate based on moving average for consecutive three periods of 10 s each and generation of alert. The moving average is updated every 10 s, and thus, the blink rate (in blinks/minute) is also updated every 10 s which is obtained by multiplying the



Fig. 2 Experimentation setup

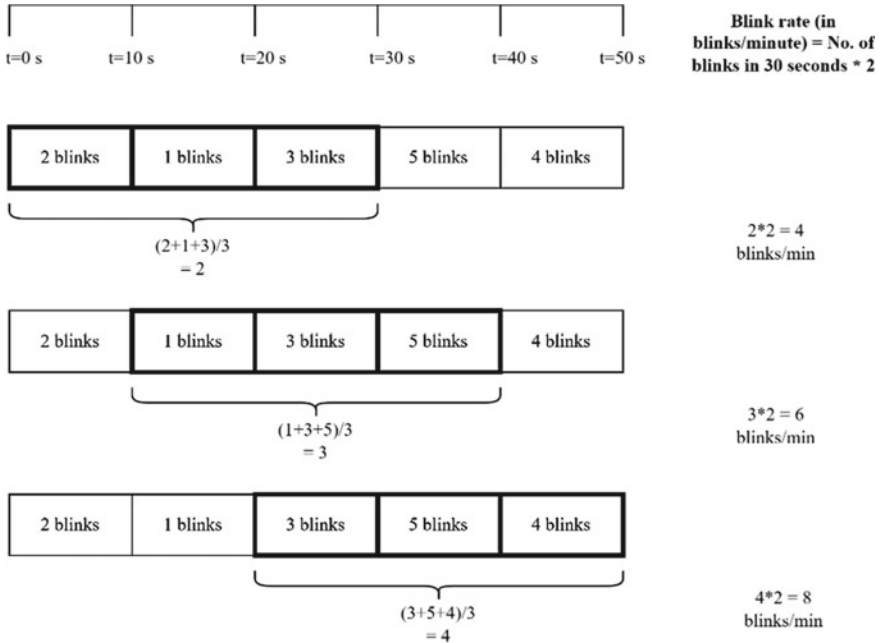


Fig. 3 Illustration of algorithm to calculate blink rate in blinks/minutes using moving averages

moving average by 2. The algorithm to calculate blink rate in blinks/minutes using moving averages is illustrated in Fig. 3.

The essence of the developed algorithm is discussed below:

- The blink rate is calculated using the moving average of blink counts over three consecutive durations of 10 s each.
- The blink rate of 12–15 is considered normal as also found through various researches. The blink rate at the start is calculated based on for first 30 s of the onset of execution of the algorithm and the hardware.
- The blink rate categorization is done in the following manner:
 - **Poor Fatigue Level:** When blink rate > 20 blinks/minute, blink rate is considered high, and a Yellow alert is generated.
 - **Poor Drowsiness Level:** When blink rate < 8 blinks/minute, blink rate is considered low, and a Red alert is generated.
 - Drivers are said to be sleepy if blink rate is below 10 per minute, while for fatigued ones, the number of blinking is between 20 and 40 [26].

In this eye blink-based method, computer vision algorithm and camera hardware for real-time detection of driver state of alertness are used. A small camera is installed in front of the participant/user/driver which monitors every blink of the eye to detect drowsiness and fatigue of the participant. The assumption is that the driver’s attention is too low when they are looking away for an extended period of time or when their

eyes are covered by an extended period of time—by blinking or rotation occlusion [26–28]. The proposed algorithm of the system is presented in the flowchart in Fig. 4.

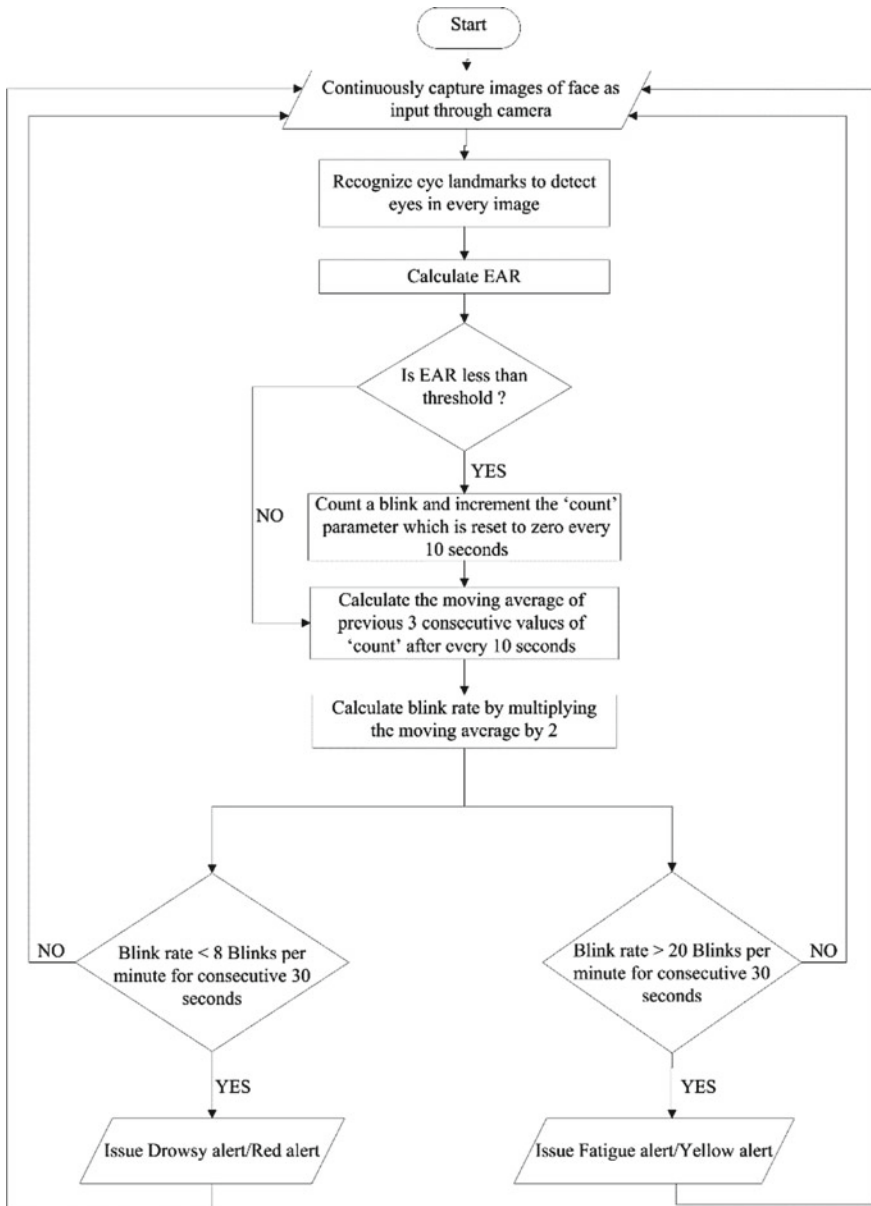


Fig. 4 Flowchart for system algorithm

In computer vision, the eye area is analyzed to identify whether or not the eye is open using a measure called eye aspect ratio (EAR). The EAR is derived using the ratio of the distance between the eye's horizontal landmarks and vertical landmarks. An eye is said to be open if its EAR value is high, whereas it is said to be closed if it has a low EAR value. In order to automatically analyze a subject's blinking patterns, EAR is frequently employed in the context of eye blink rate detection. The frequency and length of blinks can be identified by tracking variations in EAR over time. This information is useful for determining a subject's level of visual attention and cognitive function by revealing how often and how long they blink. In this case, the EAR is used to determine whether the "white" area of the eyes ever actually disappears. P1, P2, P3, P4, P5, and P6 are 2D facial landmark positions as illustrated in Fig. 1. Each eye is represented by six (x, y) -coordinates, starting from the left-corner of the eye (as if we are gazing at the person) and moving clockwise around the remaining region. The EAR is given by the expression (1). While the numerator is used to calculate the distance between vertical eye landmarks, the denominator is used to calculate the distance on horizontal eye landmarks while weighting the denominator as there are two sets of vertical points and only one set of horizontal points.

4 Results and Discussions

The major goal of this research study is to provide a system for the identification of driver impairment caused by drowsiness and fatigue. In order to design, validate, and improve such system, fresh investigations and experimentation have been conducted, which are discussed in this paper.

The test setup is established and experimentation is conducted in various conditions. The distance between the participant eyes and the camera is kept at 2 feet. The ambient illuminance is varied in three steps, viz. 5, 100 and 300 lx which is measured using Light Meter App, which is a simple light meter for measuring illuminances (lux or fc) by using the light sensor of the Android mobile device. The approximate angle of user head with respect to the camera's optical axis (in degree) is kept $0^\circ \pm 5^\circ$ in vertical and horizontal direction. Consent was taken from each participant before the experimentation, and they were explained about the experimentation goals. Details of the participants and test preparation is given below:

User-1: Age = 21 Y, B.Tech student; User-2: Age = 34 Y, PhD Student; User-3: Age = 45 Y, Professional at mid-level; User-4: Age = 50Y, Professional at senior-level; User-5: Age = 64 Y, Retired Govt. employee. No special preparation in terms of daily routine of the participant was made; however, the tests were performed for minimum 30 min in each session for each user.

The software interface is developed in Python for processing the real-time image. The developed method uses this to detect face in image and further process information to detect the alertness/drowsiness condition of the driver. The facial landmark detection is performed, and thus, blink rate is calculated from video streams. The

processing frames for computing blinks involve facial landmark recognition, eye localization, thresholding to find the eye whites, calculating the eye aspect ratio (EAR), and recording eye blinks.

The test results observed during the experimentation are listed in Table 1.

The graphical representations of blink rate variation with fatigue/drowsiness level on 5 different times of the day at 10:00, 12:00, 15:00, 19:00, and 23:30) for the five participants (User-1, User-2, User-3, User-4, and User-5) are shown in Fig. 6a–e.

The screenshots during experimentation process showing drowsiness and fatigue conditions are shown in Fig. 5a, b, respectively.

Inferences drawn from the test results obtained during the study are as under:

- The study involved limited number of participants. The fatigue or drowsiness level of the participant is simulated by carrying out measurements at different parts of the day, viz.
 - 10:00, when a person is fresh at the start of the day so blink rate is likely to be normal
 - 12:00, when a person feels slightly lesser fresh due to sleepy effect after consumption of breakfast and nearing of lunch time so blink rate may be slightly reduced
 - 15:00, when a person feels slightly sleepy effect after consumption of lunch and general tiredness as the day has progressed so blink rate may drop down further
 - 19:00, when a person is tired after elapse of full working day but still since meals time has passed some 6 h before so blink rate is likely to be normal or above normal
 - 23:30, when a person feels sleepier after consumption of dinner and tiredness at the end of the day so blink rate is likely to be lowest.

The experiments were performed for two days for each participant.

- Five participants are involved in different age group, viz. 21, 34, 45, 50, and 64. They represented different age group and different professional backgrounds in terms of being students, working professional at different levels and one being the retired govt. employee, thus covering a decent range of variation in the test subjects.
- The simulated test setup and conditions worked well, and we could get the results where some correlation could be made in terms of relation of eye blink rate to the fatigue/drowsiness level of the participant. A moving average of blink counts over three consecutive periods of 10 s each is used to calculate the blink rate. The blink rate is calculated based on moving average calculation which is updated every 10 s. The blink rate between 8 and 20 blinks/minute was considered as normal.
- The following results were observed during the course of test:
 - The condition of low blink rate is observed when the test was conducted around 23:30 in all the cases irrespective of age and professional status. The test was however needed to be conducted for more time (30 min or more) with results

Table 1 Results during the experimentation

User details	Amb_illum_1 (lx)	Eye blink rate	QA	User details	Amb_illum_1 (lx)	Eye blink rate	QA
Time: 10:00				User 3	100	15	High
User 1	5	12	High	High	300	15	High
User 1	100	12	High	User 4	5	6	Low
User 1	300	13	High	User 4	100	8	Medium
User 2	5	14	High	User 4	300	8	Medium
User 2	100	13	High	User 5	5	13	High
User 2	300	14	High	User 5	100	12	High
User 3	5	13	High	User 5	300	14	High
User 3	100	15	High	Time: 19:00			
User 3	300	15	High	User 1	5	17	High
User 4	5	13	High	User 1	100	18	Medium
User 4	100	14	High	User 1	300	19	Medium
User 4	300	14	High	User 2	5	16	High
User 5	5	14	High	User 2	100	18	Medium
User 5	100	13	High	User 2	300	18	Medium
User 5	300	14	High	User 3	5	19	Medium
Time: 12:00				User 3	100	17	High
User 1	5	10	High	User 3	300	17	High
User 1	100	11	High	User 4	5	16	High
User 1	300	12	High	User 4	100	18	Medium
User 2	5	12	High	User 4	300	18	Medium
User 2	100	13	High	User 5	5	17	High
User 2	300	13	High	User 5	100	19	Medium
User 3	5	14	High	User 5	300	17	Medium
User 3	100	15	High	Time: 23:30			
User 3	300	15	High	User 1	5	10	High
User 4	5	10	High	User 1	100	12	High
User 4	100	11	High	User 1	300	9	Medium
User 4	300	11	High	User 2	5	9	Medium
User 5	5	15	High	User 2	100	8	Medium
User 5	100	14	High	User 2	300	7	Low
User 5	300	14	High	User 3	5	11	High
Time: 15:00				User 3	100	26	Low
User 1	5	8	Medium	User 3	300	11	High
User 1	100	7	Medium	User 4	5	6	Low
User 1	300	7	Medium	User 4	100	7	Low

(continued)

Table 1 (continued)

User details	Amb_Illum_1 (lx)	Eye blink rate	QA	User details	Amb_Illum_1 (lx)	Eye blink rate	QA
User 2	5	10	High	User 4	300	7	Low
User 2	100	11	High	User 5	5	4	Low
User 2	300	10	High	User 5	100	6	Low
User 3	5	14	High	User 5	300	7	Low

Test Conditions: Dist = 2 feet, Amb_Illum = 5, 100, and 300 lx; Ang = 0°; General qualitative assessment of Alertness level—QA (drowsiness/fatigue) in 3 levels by visual inspection of the participants: high, medium, and low

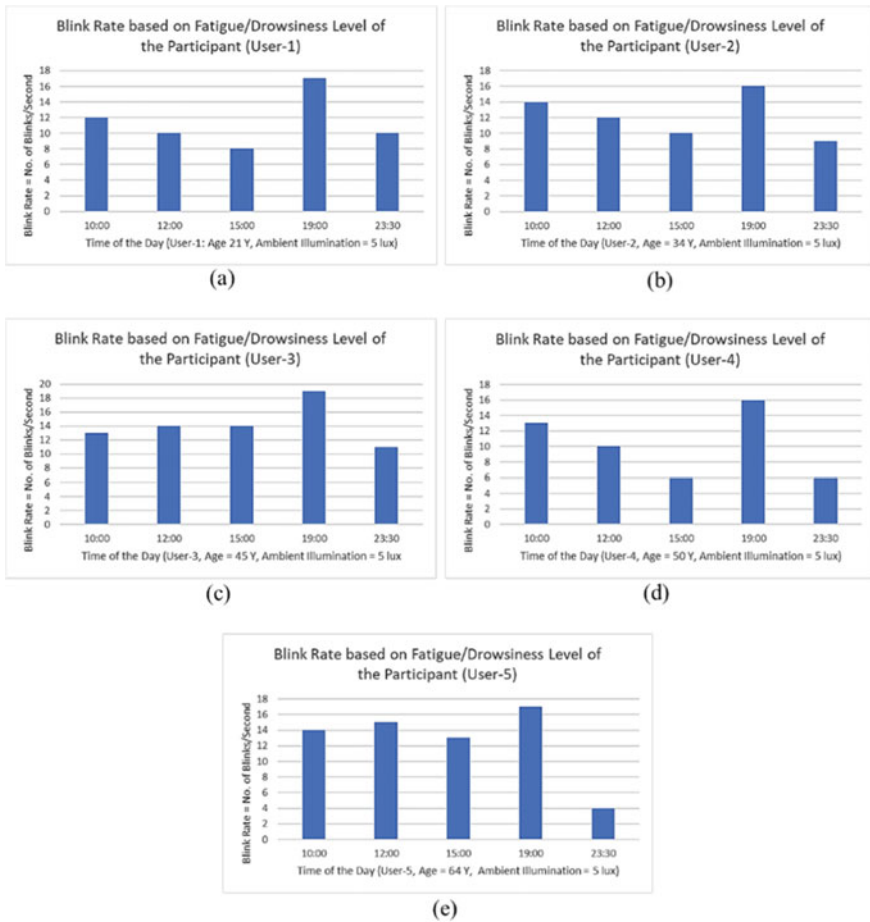


Fig. 5 Graphical representations of blink rate variation with the fatigue/drowsiness level for **a** User-1, **b** User-2, **c** User-3, **d** User-4, **e** User-5

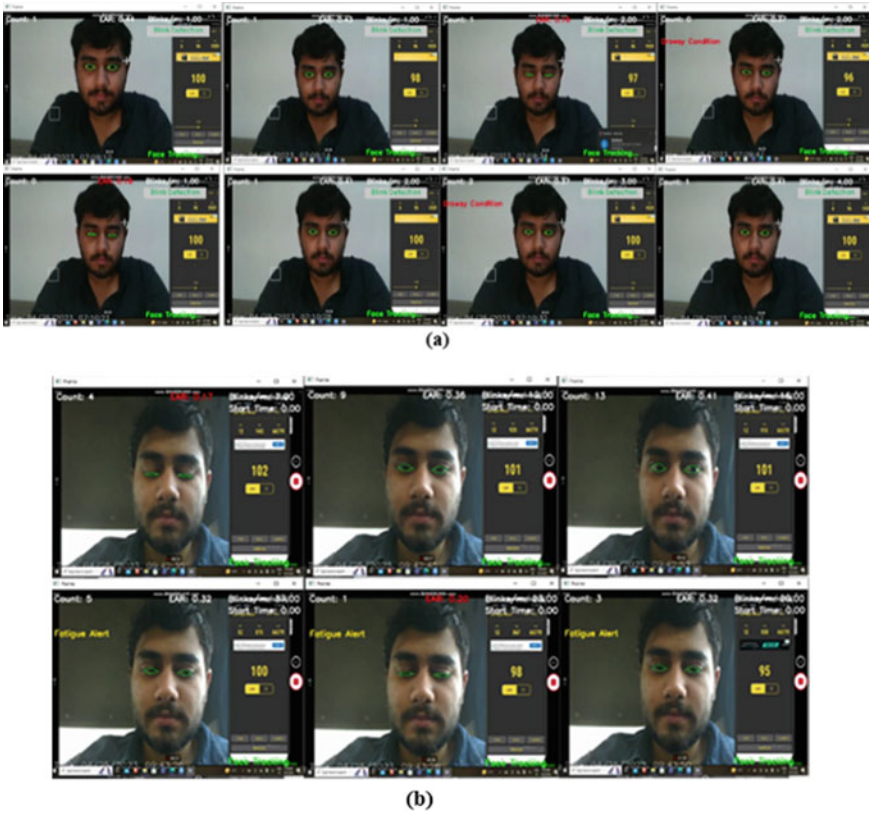


Fig. 6 a Screenshots during experimentation process showing drowsiness condition, b screenshots for experimentation showing fatigue condition

being more prominent when the participants were resting their back on the chair. The blink rate of less than 8 blinks/minute was observed in two cases (User-4 and User-5); while in other cases (User-1, User-2, and User-3), the blink rate was observed to be between 8 and 12 which indicated slight lower alertness level but not at the alarming level.

- For time of the day at 10:00 and 12:00, the blink rate was normal and between 10 and 15 blinks/minute in all the cases. All the participants were also found to be alert qualitatively during the course of tests conducted during this duration.
- For the test conducted at 15:00, the blink rate for User-1 is less than 8 blinks/minute for all ambient illuminance levels and between 6 and 8 blinks/minute for User-4 indicating lower alertness level referring to poor drowsiness level. For other users, the blink rate is found to be between 10 and 15 blinks/minute indicating normal alertness level meaning thereby that there is no drowsiness in these users.

- For the test conducted at 19:00, the blink rate for all users is found to be in slighter higher between 16 and 19 blinks/minute indicating increased fatigue level but not at the alarming level.
 - When the blink rate is less than 8 blinks/minutes, then a Red alert is generated indicating poor drowsiness level.
 - When blink rate is over 20 blinks/minute, a Yellow alert is generated indicating poor fatigue level.
- The ambient light/illuminance has impact on alertness level of the participant, but more rigorous experimentation is required. The effect of higher and lower ambient illuminance can be overcome by use of camera with auto IRIS control having good sensitivity at lower light levels.
 - Although quite accurate, physiological parameter-based drowsiness and fatigue level monitoring is obtrusive and therefore unsuitable for vehicle applications. The driver behavior-based solutions rely on sensor data, which may not always be a valid predictor of a driver's level of fatigue because sensors cannot distinguish between the effects of stress and medication. The methods for detecting changes in head position and in the speed, deviation, and position of a moving object depend too much on the accurate collection of data from numerous sensors, and they also require complex computations and hardware. The hardware and computational requirements for the facial expression-based approaches are high, whether used alone or in conjunction with eye blink detection. Only eye fatigue is detected by the EAR-based detection method using spontaneous blink rate, and it requires substantial computational training. The other EAR-based method estimates the rate of eye blinking in terms of pixels/seconds to identify tiredness. Individual user training is required for these EAR-based techniques. Additionally, neither of these two methods carries out the simultaneous detection of drowsiness and fatigue. The proposed method, in contrast, uses a single camera for detection and calculates blink rate to estimate both drowsiness and fatigue levels in blinks/minute based on a moving average calculation over a period of 30 s that is updated every 10 s and then classifies the drowsiness or fatigue alert based on related studies published in the literature. As a result, the proposed method is much less complex in terms of hardware and computation, and the results are repeatable.

5 Conclusion

In this study, we demonstrated that it is possible to discern between a driver's awake and drowsy blinks in real time by measuring the pace of the eyes blinking. We are especially adept at spotting a participant's (driver's) drowsiness or fatigue as it develops over time. When it comes to reducing automotive accidents brought on by microsleep or increased fatigue, such a real-time drowsiness monitoring system is crucial. The lower and higher blink rates could be effectively detected in real time, thus resulting in real-time alertness which can alert the driver in fatigue and drowsy conditions. The moving average calculation performed over a period of 30 s which is

updated every 10 s provides real-time update of the blink rate. This is a novel finding reported in this work which results in estimation of drowsiness and fatigue condition in real time with repeatable results. While the effect of distance, angle and ambient illuminance play a role in detection but those could be overcome by proper camera settings as well as through fine tuning of the algorithm. The threshold of blink rates below 8 for drowsy condition and blink rate above 20 for fatigue condition can be fine-tuned by increasing the scope of experimentation in the future.

The proposed simple and cost-effective solution in terms of real-time detection of driver drowsiness and fatigue can alert drivers when they become drowsy or fatigued, which in turn will help reduce the number of road accidents and thus reduce the economic costs associated with road accidents.

Experimentation could be carried out for more types of eye shapes such as narrow and round, in order to estimate a more generic EAR threshold for determination of an eye blink. Effect of varied eye positions and distance of head with respect to camera could be studied.

References

1. Road safety factsheet: driver fatigue and road accidents factsheet. The Royal Society for the Prevention of Accidents, July 2020. <https://www.rosopa.com/media/documents/road-safety/driver-fatigue-factsheet.pdf>
2. Driver fatigue and road accidents: a literature review and position paper. The Royal Society for the Prevention of Accidents Driver Fatigue and Road Accidents, Feb 2001
3. Traffic safety facts: drowsy driving. United States Department of Transportation, Drowsy Driving: Avoid Falling Asleep Behind the Wheel. NHTSA. <https://www.nhtsa.gov/risky-driving/drowsy-driving>
4. Laouz H, Ayad S, Terrissa LS (2020) Literature review on driver's drowsiness and fatigue detection. In: 2020 international conference on intelligent systems and computer vision (ISCV), Fez, Morocco, pp 1–7. <https://doi.org/10.1109/ISCV49265.2020.9204306>
5. Furugori S, Yoshizawa N, Iname C, Miura Y (2005) Estimation of driver fatigue by pressure distribution on seat in long term driving. *Rev Automot Eng* 26(1):053–058
6. Analysis of electrocardiogram and photoplethysmogram signals to detect car driver drowsiness using the threshold method (2023). https://doi.org/10.1007/978-981-99-0248-4_43
7. Gupta AS, Kumari M, Shokeen S, Mishra A, Singh A (2022) EEG and ECG-based drowsiness detection: a review on state of the art. In: Gao XZ, Tiwari S, Trivedi MC, Singh PK, Mishra KK (eds) *Advances in computational intelligence and communication technology. Lecture notes in networks and systems*, vol 399. Springer, Singapore. https://doi.org/10.1007/978-981-16-9756-2_4
8. Omidyeganeh M, Javadtalab A, Shir Mohammadi S (2011) Intelligent driver drowsiness detection through fusion of yawning and eye closure. In: 2011 IEEE international conference on virtual environments, human-computer interfaces and measurement systems proceedings, Ottawa, ON, Canada, pp 1–6. <https://doi.org/10.1109/VECIMS.2011.6053857>
9. Lew M, Sebe N, Huang T, Bakker E, Vural E, Cetin M, Ercil A, Littlewort G, Bartlett M, Movellan J (2007) Drowsy driver detection through facial movement analysis. In: *Human computer interaction*, vol 4796. Springer, Berlin, pp 6–18
10. Yin BC, Fan X, Sun YF (2009) Multiscale dynamic features based driver fatigue detection. *Int J Pattern Recogn Artif Intell* 23:575–589

11. Kumar V, Sharma S, Ranjeet (2022) Driver drowsiness detection using modified deep learning architecture. *Evol Intel*. <https://doi.org/10.1007/s12065-022-00743-w>
12. Brandt T, Stemmer R, Rakotonirainy A (2004) Affordable visual driver monitoring system for fatigue and monotony. In: IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583), vol 7, pp 6451–6456
13. Yusri MF, Mangat P, Wasenmüller O (2021) Detection of driver drowsiness by calculating the speed of eye blinking. <https://doi.org/10.48550/arXiv.2110.11223>
14. Kuwahara A, Nishikawa K, Hirakawa R, Kawano H, Nakatoh Y (2022) Eye fatigue estimation using blink detection based on Eye Aspect Ratio Mapping (EARM). *Cogn Robot* 2:50–59. ISSN 2667-2413. <https://doi.org/10.1016/j.cogr.2022.01.00>
15. Andreassi JL (2006) Psychophysiology: human behavior and physiological response, 5th ed. Psychology Press. <https://doi.org/10.4324/9780203880340>
16. Peters B, Anund A (2004) System for effective assessment of driver vigilance and warning according to traffic risk estimation—preliminary pilot plans—revision II. VTI (Swedish National Road and Transport Research Institute), Linköping
17. Thorslund B, Anund A, Forsman Å, Gustafsson S, Soerensen G (2004) Electrooculogram analysis and development of a system for defining stages of drowsiness. Master's thesis project in biomedical engineering, reprint from Linköping University. Department Biomedical Engineering, LIU-IMT-EX-351, LINKÖPING 2003
18. Hargutt V, Kruger HP (2000) Eyelid movements and their predictive value for fatigue stages. In: International conference on traffic and transport psychology—ICTTP 2000, 4–7 Sept 2000, Berne, Switzerland
19. Singh J (2020) Learning based driver drowsiness detection model. In: 3rd international conference on intelligent sustainable systems (ICISS), Thoothukudi, India, pp 698–701. <https://doi.org/10.1109/ICISS49785.2020.9316131>
20. Chirra VRR, Uyyala SR, Kolli VKK (2019) Deep CNN: a machine learning approach for driver drowsiness detection based on eye state. *Revue d'Intelligence Artificielle* 33(6):461–466. <https://doi.org/10.18280/ria.330609>
21. Albadawi Y, Takruri M, Awad M (2022) A review of recent developments in driver drowsiness detection systems. *Sensors* 22(5):2069. <https://doi.org/10.3390/s22052069> [online]
22. Shekari SS, Wilkinson VE, Cori JM, Westlake J, Stevens B, Downey LA, Shiferaw BA, Rajaratnam SMW, Howard ME (2019) Eye-blink parameters detect on-road track-driving impairment following severe sleep deprivation. *J Clin Sleep Med* 15(9):1271–1284. <https://doi.org/10.5664/jcsm.7918>. PMID: 31538598; PMCID: PMC6760410
23. Schmidt J, Laarousi R, Stolzmann W, Karrer-Gauß K (2018) Eye blink detection for different driver states in conditionally automated driving and manual driving using EOG and a driver camera. *Behav Res Methods* 50(3):1088–1101. <https://doi.org/10.3758/s13428-017-0928-0>
24. Soukupova T, Cech J (2016) Eye blink detection using facial landmarks. In: 21st computer vision winter workshop, Rimske Toplice, Slovenia, p 2, Feb 2016
25. Paul S, Mubarak S, da Vitoria Lobo N (2000) Monitoring head/eye motion for driver alertness with one camera, VL-15, Sept 2000. IEEE Xplore. <https://doi.org/10.1109/ICPR.2000.902999>
26. Pasaribu NTB, Prijono A, Ratnadewi R, Adhie RP, Felix J (2018) Drowsiness detection according to the number of blinking eyes specified from eye aspect ratio value modification. In: Advances in social science, education and humanities research, vol 203. International conference on life, innovation, change, and knowledge (CLICK 2018). <https://doi.org/10.2991/iclick-18.2019.35>
27. Ariel G (2022) Eye-tracker in the car keeps drivers awake and alert. No Camels Weekly Newsletter, 21 Aug 2022
28. Feld H, Mirbach B, Katrolia J, Selim M, Wasenmüller O, Stricker D (2021) DFKI cabin simulator: a test platform for visual in-cabin monitoring functions. In: Commercial vehicle technology. Springer, 417–430

Selection of Robust Text-Based CAPTCHA Using TensorFlow Object Detection Method



R. Menaka and G. Padmavathi

Abstract Nowadays, websites are facing multiple security issues. Today, web pages are protected by specified security alerts to overcome these issues, in that CAPTCHA security is still in use. The logic behind designing the robust text-based CAPTCHA is ‘an increase in the confusion rate indicates the high quality of the CAPTCHA.’ By studying the cracking techniques, the CAPTCHA developer can revise the design of text-based CAPTCHA. So that the cracking rate can be minimized and the confusion rate can be maximized. Open-source text-based CAPTCHAs are used for launching the chosen plain text attack. The choice of open-source CAPTCHA library in generating the unlimited CAPTCHA dataset can reduce the cost. To select the robust text-based CAPTCHA from the open-source CAPTCHA library, a framework is developed with TensorFlow object detection and convolutional neural network. The proposed method is tested using 40 numeric and 12 alphanumeric CAPTCHA along with special character.

Keywords Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA) · Security · Chosen plain text attack · Convolutional Neural Network (CNN) · TensorFlow Object Detection (TOD)

1 Introduction

In 2003, Von Ahn et al. introduced the term CAPTCHA, a Turing test tool that distinguishes the human user and machine from their response. The purposes of CAPTCHA security are to prevent the customers’ websites from attacks and to retain the end users’ uninterrupted flow. The successful growth of E-Commerce depends

R. Menaka (✉)

Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India
e-mail: 21phcsf005@avinutty.ac.in

G. Padmavathi

Dean School of Physical Sciences and Computational Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India

on software intelligence to detect malicious activities and network traffics on the customers’ websites constantly. To overcome the online CAPTCHA security issues, modern approach is to be used to design the CAPTCHA to minimize the cracking rate. Figure 1 shows the classification of CAPTCHA and its features [1].

- The main objective is to evaluate the robustness of the text-based CAPTCHA based on the following key points.
- Checking the character feature of DISTORTED CAPTCHA will tighten the cracking rate.
- Analyze whether the open-source Python CAPTCHA library used as the chosen plain text attack is well suited for developing the cracking model in TOD and CNN environments. The choice of open-source library is to reduce the cost, particularly in small business sites.

The remaining section of the paper is as follows: Sect. 2 discusses the related work, Sect. 3 discusses the proposed methodology, Sect. 4 discusses the dataset used, and Sects. 5 and 6 follow the result and conclusion.

2 Related Works

The existing literature on CAPTCHA is separated as CAPTCHA-In earlier stage, CAPTCHA-In Cracking Strategy, and CAPTCHA-Using Known plain text attack.

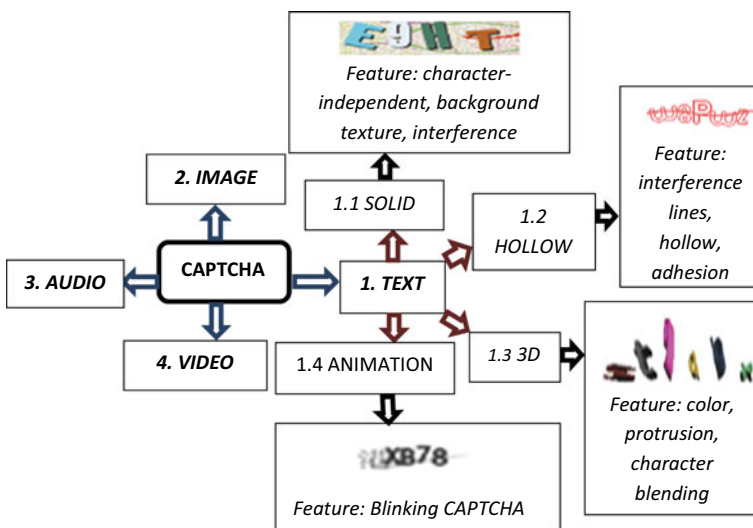


Fig. 1 CAPTCHA classification flow

2.1 CAPTCHAs-In Earlier Stage

- In 2005, Chellapilla et al. developed a model with adequate machine learning methods that enhances the recognition rate that updates the segmentation issues [2]. In 2008, Kluever et al. proposed a work in which CAPTCHA extracts the videos from public domains and compares them with bots entry [3].

2.2 CAPTCHA-In Cracking Strategy

- In 2014, Jaderberg et al. proposed a model for text spotting from an intact image with the updated CNN architecture that allows effective features for detecting the text, case-sensitive and insensitive classification in character, and bigram classification [4]. In 2017, Martin Kopp et al. proposed a CAPTCHA recognition approach, which can fully replace the state-of-the-art scheme in precise flow. The study experimentally compared the ability of Single Hidden Layer Perceptron (SLP), Multi-hidden Layer Perceptron (MLP), and CNN to record characters from CAPTCHA images [5]. In 2018, Bostik et al. exposed that a Feed-Forward Network shows a better implementation rate than other neural network models in evaluating the CAPTCHA database. In 2020, Chunhui Li et al. proposed an end-to-end attack on text-based CAPTCHA to break the scheme [6]. In 2021, Zhong Wang and Peibei Shi tried to enhance the quality of the CAPTCHA recognition section with a deep learning network [7].

2.3 CAPTCHA-Using Known Plain Text Attack

- In 2018, Ye et al. used the CAPTCHA sample dataset using Generative Adversarial Network support to simulate the small sample input and generate a large dataset of synthetic CAPTCHAs for training the deep learning method [8]. To enhance the characters recognition rate, ‘confusion class’ is introduced by Chunhui et al. in the year 2019 using Selective Learning Confusion Class [6].

3 Methodology

With the existing framework [1], a cracking method is studied, and suitable techniques are incorporated within the framework as shown in Fig. 2.

In the research, CAPTCHA schemes, Proprietary CAPTCHA schemes (e.g., Google Re-CAPTCHA), and open-source-based CAPTCHA libraries (e.g., Python CAPTCHA library) are in use. Using the open-source-based libraries on CAPTCHA faces higher security issues than potential AI-based machines.

3.1 Distorted CAPTCHA

The encryption machine acts as a CAPTCHA generator and generates unlimited text-based CAPTCHA image that acts as chosen plain text attack with the help of an open-source library. Figure 3 shows the generated CAPTCHA character display based on rendering the features of distortions like overlapping, skewing, interference line, and background dots.

3.2 Preprocessing

The colored image will be converted into a black-and-white format, as shown in Fig. 4. Image binarization uses a global-to-local approach where the threshold value is supported to binarize the selected image [9].

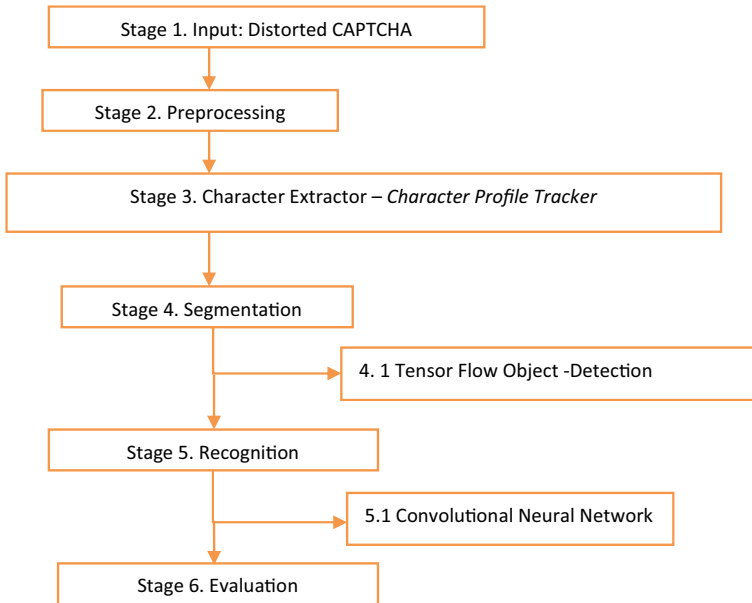


Fig. 2 Proposed methodology of cracking the text-based CAPTCHA

Fig. 3 Distorted CAPTCHA sample





Fig. 4 Distorted CAPTCHA before and after preprocessing

3.3 Character Extractor

From preprocessed CAPTCHA image, the character profile structure given in Table 1 is getting extracted from the source code of the Python CAPTCHA library. The feed-forward batch system is used to accept the upper case alphanumeric character except ‘I’ and ‘O.’

3.4 Segmentation

A bounding box process is used to tighten each character relatively close to the character text shown in Fig. 5. (Number of character label present in the CAPTCHA image is = Number of the Bounding box). Each character in the CAPTCHA label is expressed from the margin-left and end to the margin-right. Then, the entire CAPTCHA label gets minimized into the horizontal axis (x) and accumulates the values in the vertical axis (y) to act as the average line. To record the approximation location of the character, set the default threshold as pixel value to 3 and then remove the below average line. Table 2 records the value of the character profile tracker used to locate the character’s position [10–23].

Table 1 Parameter used in CAPTCHA character profile tracker

Variable	Data type
Location, label	Character array
Height, width, X-axis Y-axis	Integer array Integer

```

Algorithm 1. Segmentation procedure of a CAPTCHA [16]
Characterindex = []; // Holds the executed list generated
by segment.
Startposition = 0; // Set x-axis position
Setread = false;
Charcount = 0;
//Following expression is to reset all the values of the y-
axis to zero in case the average value is not met
Yaxis = [k if k>or=average else 0 for k in Yaxis]
forCharacterindex, k in enumerate(Yaxis) do
    If k!=0 then
        Charcount = 0;
        If not Setread then
            Setread = true;
            Startposition = Characterindex;
        end
    else
        Charcount = Charcount +1
        //Threshold is a dynamic value from the user
        If Setread = false;
            Characterindex.append((Startposition,charindex));
            Charcount = 0;
        end
    endend return Characterindex; // Finally, character indices
returned from the iteration.
    
```



Fig. 5 Bounding box used as characters’ position prediction in CAPTCHA image

Table 2 CAPTCHA character profile tracker for Fig. 5

Location	Width	Height	x-min	y-min	x-max	y-max
TRAIL_1-0/claptcha-0.png	33	55	14	0	47	55
TRAIL_1-0/claptcha-0.png	38	55	47	0	85	55
TRAIL_1-0/claptcha-0.png	28	55	85	0	113	55
TRAIL_1-0/claptcha-0.png	26	55	113	0	139	55
TRAIL_1-0/claptcha-0.png	28	55	139	0	167	55

Table 3 Different losses occurred during object detection [24]

Type of loss function	Description
Grouping loss	Loss between the set of characters and detection
Positional loss	Loss between the character and the box covered the each character
GP loss	Loss combining the last grouping and positional loss
Bounding object loss	Loss of grouping in the bounding box
Net loss	Loss between grouping, positional, GP, and bounding object

3.4.1 TensorFlow Object Detection (TOD)

In this stage, the text-based image is breakdown into 2D. One dimension indicates the location, and the other dimension is the sum of vertical pixels based on the respective location (X -axis direction). The characters are segmented based on their position lists which act as the character index. It is used as a character detection model using Google's object detection module within the TensorFlow open-source module. Five loss functions are observed over 10k iterations while training the model (Table 3).

3.5 Recognition

In this phase, existing methods used in the text-based CAPTCHA cracking model are template matching, character feature, and machine learning [1].

3.5.1 Convolutional Neural Network (CNN)

The performance gets optimized by multi-layer perceptron (MLP) to accomplish the feed-forward artificial neural network in the model structure. By taking advantage of MLP, the construction of the CNN model is designed with 128 batch sizes with 30 epochs and 0.2 as the dropout rate. The Deep Neural Network architecture is applied to a 28×34 input shape transformation to ensure the metrics in the normalized inputs. Next to this MaxPool-2D layer is used to reach the dropout layer. Finally, the softmax layer helps to optimize the training accuracy to 99% (Fig. 6). The input layer is the user response to execution time.

4 Dataset

There are two datasets used: (i) Internal Dataset and (ii) External Dataset.

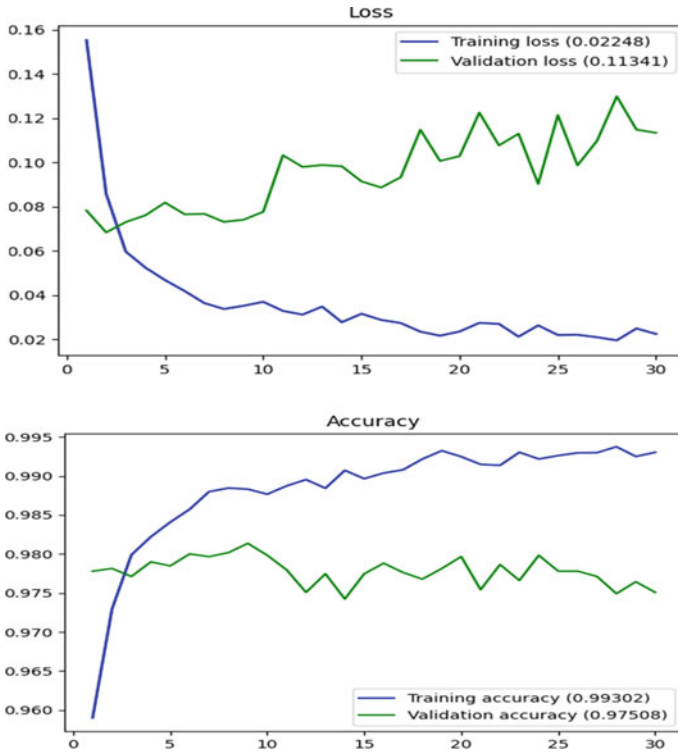


Fig. 6 Training and validation-accuracy and loss-graph over single batch set under 30 epochs

4.1 Internal Dataset

The unlimited cipher text (CAPTCHAs) generated by the CAPTCHA generator using chosen plain text attack acts as the internal dataset. In developing a model, 42 characters are used [24 alphabets (excluding O and I), 10 numeric characters, and 8 special characters (!@#\$\$%^&*)]. Based on the iteration count, cipher texts (CAPTCHAs) were generated.

4.2 External Dataset

Based on related literature, the study has used Hashkiller and Delta Airlines. The Hashkiller dataset contains 13 CAPTCHA image files with a length of 6. Delta Airlines dataset contains 40 CAPTCHA image files length of 5 characters. Delta Airline dataset: Numeric characters, Hashkiller dataset: Alphanumeric characters.

5 Results and Discussion

The TOD + CNN combination suits well in cluttering and tightening the bounding box in the selected image from the dataset (internal and external) character cracking rates shown in Figs. 7 and 8. Figures 9 and 10 show the overall confusion rate chart summary based on the equation, $i = \frac{TF}{2}$, if $(i < FI)$ then c.r. = true, else c.r. = false here, TF = Total Frequency Of Character; FI = Frequency Of Incorrect Recognition; c.r. = confusion rate.

Guideline By referring Figs. 8 and 9, the following character increases in confusion rate, {'F,' 'H,' 'N,' 'T,' 'W,' 'Y,' '5,' '\$'}. Hence, this statement supports to revise of the design of robust text-based CAPTCHA.

Fig. 7 Evaluation metrics for cracking rate on internal dataset

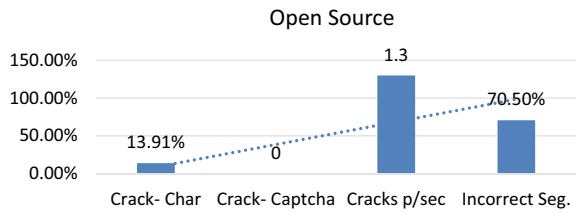


Fig. 8 Evaluation metrics for cracking rate on external dataset

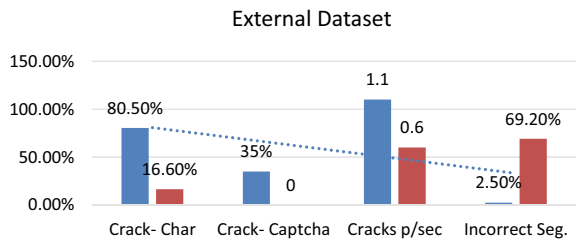


Fig. 9 Confusion rate chart summary for 24 alphabet character

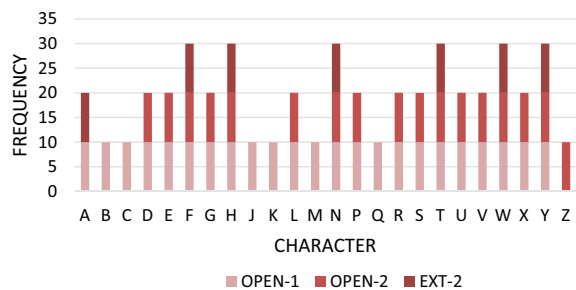
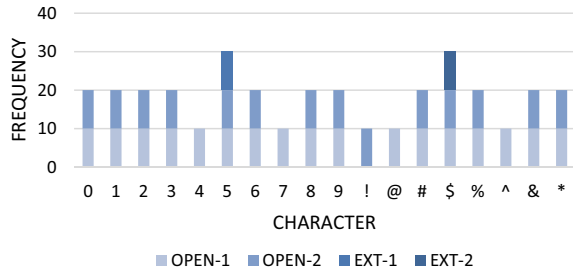


Fig. 10 Confusion rate chart summary for 18 numeric and special character



6 Conclusion

In this paper, work concentrates more on testing the open-source CAPTCHA library that generates unlimited text-based CAPTCHAs with the help of chosen plain text attacks. The approach proposed based on the TensorFlow Object Detection method suits well for text cluttering in the segmentation phase. By applying deep learning techniques, the patterns of images trained and recognized to explore the semantic details of the image dataset and connect with many more new patterns that support to achieve the minimum cracking rate and maximum confusion rate of the model.

Limitations: The upcoming AI-based CAPTCHA system uses essential guide lines to overcome the recent challenges in cracking rate with other types of CAPTCHA features. To fool the machine learning bots, an unpredictable CAPTCHA technique is the need of the hour.

References

1. Chen J, Luo X, Guo Y, Zhang Y, Gong D (2017) A survey on breaking technique of text-based CAPTCHA, Wiley, Hindawi. Secur Commun Netw 2017:6898617
2. Chellapilla K, Simard PY (2005) Using machine learning to break visual human interaction proofs (HIPs). Adv Neural Inf Process Syst 17:265–272
3. Kluever KA (2008) Video CAPTCHAs: usability vs security. In: IEEE workshop on image processing, pp 1–4
4. Jaderberg M, Vedaldi A, Zisserman A (2014) Deep features for text spotting. In: Proceedings of the European conference on computer vision, Zurich, Switzerland, 6–12 Sept 2014
5. Kopp M, Nikl M, Holeña M (2017) Breaking CAPTCHAs with convolutional neural networks. CEUR Works Proc 1885:93–99
6. Chen J, Luo X, Liu Y, Wang J, Ma Y (2019) Selective learning confusion class for text-based CAPTCHA recognition. IEEE Access 7:22246–22259
7. Wang Z, Shi P (2021) CAPTCHA recognition method based on CNN with focal loss, vol 2021. Wiley, Hindawi, Article ID 6641329, Jan 2021
8. Ye G, Tang Z, Fang D, Zhu Z, Feng Y, Xu P, Chen X, Wang Z (2018) Yet another text captcha solver: a generative adversarial network based approach. In: Proceedings of the ACM conference on computer and communications security, Toronto, ON, Canada, 15–19 Oct 2018

9. Jyotsna SC, Sharma E, Doegar A (2016) Binarization techniques for degraded document images—a review. In: International conference on reliability, Infocom technologies and optimization (ICRITO), AIIT, Amity University Uttar Pradesh, Noida, India, 7–9 Sept 2016
10. von Ahn L, Blum M, Langford J (2004) Telling humans and computers apart automatically. *Commun ACM* 47(2)
11. Chow YW, Susilo W, Thorncharoensri P (2019) CAPTCHA design and security issues. In: advances in cyber security: principles, techniques, and applications. Springer, Berlin, Germany, pp 69–92
12. Bostik O, Klecka J (2018) Recognition of CAPTCHA characters by supervised machine learning algorithms. *IFAC-PapersOnLine* 51:208–213
13. Zhang Y, Gao H, Pei G, Luo S, Chang G, Cheng N (2019) A survey of research on CAPTCHA designing and breaking techniques. In: 2019 18th IEEE international conference on trust, security and privacy in computing and communications/13th IEEE international conference on big data science and engineering
14. Moradi M, Keyvanpour M (2015) CAPTCHA and its alternatives: a review. *Secur Commun Netw* 8:2135–2156
15. Sivakorn S, Polakis I, Keromytis AD (2016) I am robot: (deep) learning to break semantic image CAPTCHAs. In: Proceedings of the 2016 IEEE European symposium on security and privacy (EuroS P), Saarbrücken, Germany, 21–24 Mar 2016, pp 388–403
16. Nguyen VD, Chow YW, Susilo W (2012) Attacking animated CAPTCHAs via character extraction. In: Proceedings of the international conference on cryptography and network security, Paraty, Brazil, 20–22 Nov 2012, pp 98–113
17. Huang S-Y, Lee Y-K, Bell G, Ou Z (2009) An efficient segmentation algorithm for CAPTCHAs with line cluttering and character warping. Springer Science, 1 Aug 2009
18. Gao H, Wang W, Qi J, Wang X, Liu X, Yan J (2013) The robustness of hollow CAPTCHAs. *ACM*, Berlin, Germany, 04–08 Nov 2013
19. Referenced Website: <https://www.cloudways.com/blog/ecommerce-security-tips/>
20. Referenced Website: <https://owasp.org/www-project-automated-threats-to-web-applications/>
21. Thomas C, Fraga-Lamas P, Fernandez-Carames TM (2020) Computer security threats. *IntechOpen*, 130 pp, 9 Sept 2020. ISBN 978-1-83880-240-0, EBOOK (PDF) ISBN 978-1-83962-381-3
22. https://assets.barracuda.com/assets/docs/dms/Bot_Attacks_report_vol1_EN.pdf
23. Li C, Chen X, Wang H, Zhang Y, Wang P (2020) An end-to-end attack on text based CAPTCHAs based on cycle-consistent generative adversarial network, 26 Aug 2020
24. Yu N, Darling K (1998) A low cost approach to crack python CAPTCHAs using AI-based chosen plain text attack. In: Tang S, King M (eds) Applied sciences, vol II. Jiaoda Press, Xian, pp 158–176

Performance Analysis of ECC-Based Security Solutions for Internet of Medical Things



Anuj Kumar Singh and Sachin Kumar

Abstract Significant technological breakthroughs in the field of health care have been seen recently, incorporating automatic data collection, patient monitoring, self-care tools, laboratory tests, and many others. Because of advances in technology in health sector, health services are increasingly interwoven with user-friendly, accessible devices and extend beyond the confines of hospital settings. The Internet of Medical Things abbreviated as IoMT is a suite of e-healthcare tools and software that can interconnect with medical systems through network technologies. IoMT has been very helpful in providing the real-time data of patient to the medical practitioners and other parties like hospital. But, at the same time the wireless nature of communication and data transmission poses multiple threats, and thus, there is a need of robust and efficient IoMT security protocol providing highest level of security and consuming less cost. In this work, an analysis of performance of recent elliptic curve based IoMT security schemes has been carried out in terms of security properties they satisfy, the attacks they counter, computational cost, storage cost, and communication cost. The comparative analysis shows that maximum number of these schemes are susceptible to security attacks while the others incur high cost. Thus, the study presented here lays out the foundation for the development of futuristic security solutions for IoMT.

Keywords IoMT · Security · Privacy · ECC · Authentication · Attacks · Solutions

A. K. Singh (✉)
Amity University Madhya Pradesh, Gwalior, India
e-mail: anujbtechcs@gmail.com

S. Kumar
South Ural State University, Chelyabinsk, Russian Federation

1 Introduction

Advancements in technology have given rise the usage of embedded devices and sensors in the area of medical science. With the progression of IoMT which is an interconnection of hardware, software, and medical devices, it has become possible to communicate sensitive medical data securely over the Internet rapidly. Moreover, the significance of IoMT is evident from its impact on the healthcare industry and its deployment in-hospital, in-body, in-community, and in-home. IoMT combines Internet of Things with healthcare systems to provide remote patient monitoring and treatment in real time. The components building the IoMT ecosystem are shown in Fig. 1. The following four architecture layers [1] are present in IoT and Internet of Medical Things system architectures, respectively.

- Perception Layer—exemplified by the variety of smart clinical equipment gathering many types of health information.
- Connectivity Layer—accountable for leveraging connectivity technologies such as networking and gateways to convey data from the cloud to the perceptual layer and vice-versa.
- Processing Layer—enabled by IoT platforms or cloud middleware to maintain and store data.
- Application Layer—delivering technology solutions that enable end users to access data analytics, device control, and reporting.

The interconnection of critical medical equipment with other systems at various network layers, on the other hand, opens up new chances for remote opponents and more vulnerabilities [2]. With respect to the sensitivity of the medical data and regulations imposed by the governments, there is a need to provide more comprehensive security to IoMT infrastructure. IoMT data is typically transmitted via the public network, exposing it to enhanced security risks than transmitting data over a private

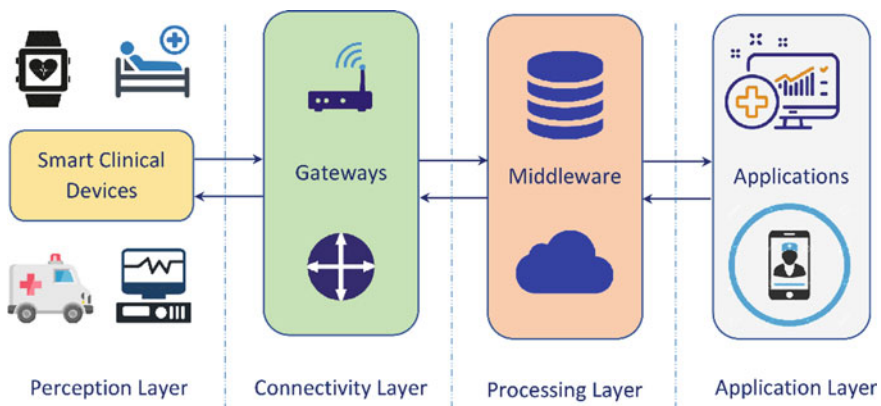


Fig. 1 Components in IoMT ecosystem

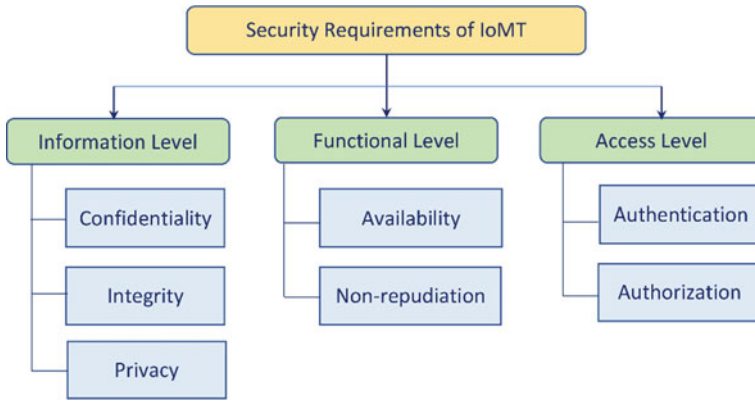


Fig. 2 Classification of levels of security requirements in IoMT

network guarded by a firewall. The fact that the data is shared throughout numerous systems that increases to the threat, enabling multiple attack avenues. The major security requirements of IoMT includes confidentiality, integrity, forward security, backward security, access control, authentication, availability, and key agreement. Alsaeed and Nadeem [3] have carried out a detailed survey on the requirements of security properties in IoMT and classified these in three levels namely information level, functional level, and access level. This classification is shown in Fig. 2. In addition to these security requirements, many attacks can be attempted over IoMT which include eavesdropping, spoofing, masquerade, traffic analysis, physical attacks, Denial of Service (DoS) attacks, impersonation attacks, and replay attacks. Moreover, attacks to violate the security attributes are also attempted. These includes, attacks on integrity, attacks on confidentiality, attacks on authentication, and attacks on availability [4]. Recently, Hireche et al. [5] made a study on the types of attacks on the different layers, namely perception, network, and application in the IoMT environment which has been summarized in Table 1.

2 Related Work

A variety of security solutions for IoMT environment have been proposed by many researchers. But due to the strength of the computational problems involving elliptic curves and less key-bits requirements, ECC-based authenticated key agreement schemes have been a favorite choice for the researchers [6, 7]. Numerous authenticated key agreement schemes and protocols for e-health involving IoMT have been designed and presented in the past decade. An authentication protocol for pervasive healthcare monitoring system was developed by Yeh et al. in 2013 [8]. Their protocol offers mutual authentication between the patient’s device and the medical server instead of end-to-end security, but this protocol performs modular exponentiation

Table 1 Attacks on IoMT environment

Layer	Attack	Elucidation
Perception	Side channel	Obtaining information from indirect sources like power consumption, time in computation, etc.
	Tag cloning	Obtaining the information from other tag by some means like side channel attack and using this information to clone the tag
	Tracking	This attack exploits the privacy of the patient by stealing the location information from the IoMT devices
	Tampering	The data within the IoMT device is modified by maliciously accessing the device
	Eavesdropping	Attacker listens to the communication media to extract confidential information
Network	Replay	The attacker captures a signed authentication message in transit and sends it to multiple parties
	Sniffing	Data interception between two devices
	Sinkhole	In this attack, a fraudulent node draws traffic by promising a higher connection quality. And from this point then other attacks are launched
	Man-in-the-middle	The attacker sits between two devices and modifies the communication between them to obtain the secret information
	Selective forwarding	Any message or part of it can be changed, deleted, or forwarded to other nodes in the network by a rogue node
	Rogue access	The access to legitimate nodes is given by a fake gateway to get confidential information
	DoS/DDoS	Submerging a legitimate node with fake messages and requests
	Brute force	The attacker attempts all possible combinations to crack the password of the user
Application	Account hijacking	Capturing authentication data from IoMT devices and hijacking the user's account
	SQL injection	Injecting a malformed SQL statement into the backend database of an application
	Ransomware	Attacker locks vital health information with encryption, retains the decryption key, and demands a hefty fee to unlock it

operations which is highly time intense. Zaho [9] presented an effective anonymous authentication method for WBANs based on identification that uses ECC. However, this scheme was found to be insecure against known session-specific temporary information assault and cannot ensure clock synchronization. Kumar et al. [10] devised a two-factor mechanism for medical sensor network and claimed that it is secure against known attacks, but He et al. [11] demonstrated that this mechanism is vulnerable to insider attacks. A new authentication technique for sensor enabled medical

networks was developed by Amin et al. in [12]. Afterward, Jiang et al. [13] highlighted that the technique of Amin et al. is susceptible to desynchronization, stolen mobile device, and sensor key disclosure attacks. An authentication system for wearable devices in a WBAN setting was designed by Das et al. [14]. Later, Chaudhry et al. [15] discovered that the authentication method of Das et al. is susceptible to a man-in-the-middle attack and device impersonation. An authentication mechanism for WBAN was suggested by Li et al. [16], but He et al. [17] found it to be vulnerable from impersonation attacks. Additionally, He et al. [17] designed a method for anonymous authentication for WBANs, but Sowjanya et al. [18] revealed that it is vulnerable to clock synchronization, insiders, and known session-specific temporary information attacks. A lightweight method offering end-to-end authentication for WBAN using ECC was developed by Li et al. [19]; however, it has been evidenced Sowjanya et al. [18] that it cannot offer forward security, key control, and clock synchronization. The authors of [18] also proposed a security mechanism for IoMT based on ECC; but in the recent work presented by Pirmoradian et al. [20], it has been shown that the scheme of [18] is defenseless against replay attack and insider secret disclosure. Singh et al. [21] developed a mutual authentication mechanism for IoT and cloud and proved that it provides highest security level. This scheme can also be made to work with IoMT environment. In addition to these schemes, the ECC-based protocols mentioned in [22–25] can also be utilized for WBAN and IoMT settings since they are efficient in terms of security properties and costs.

From the discussion made under related work, it can be observed that though a variety of security schemes and methods have been designed by different authors, some of these schemes have serious security flaws while some of them are not efficient. Thus, to analyze these schemes critically performance evaluation must be carried out regarding the security functions they provide and the cost they incur.

3 Performance Analysis of Security Schemes for IoMT

The performance evaluation of security schemes for IoMT can be done on the basis of five parameters: (i) security attributes they provide, (ii) the attacks they can resist, (iii) computational cost they incur, (iv) communication cost they take, and (v) storage cost they require. However, there has been a trade-off between the security functions a scheme can offer and the different costs associated with the scheme, but a security scheme that fulfills maximum number of security features has always been on priority for the researchers and developers. This section that evaluates the performance of the ECC-based IoMT schemes and protocols developed in the recent years mentioned in [9, 14, 17–20, 25] has been done with respect to these five parameters.

Table 2 Security attributes of ECC-based IoMT protocols

Protocol	Security attributes									
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Zhao [9]	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{F}
Das [14]	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{F}
He [17]	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{P}
Sowjanya [18]	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}
Li [19]	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{P}	\mathcal{P}
Sowjanya [18]	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}
Pirmoradian [20]	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}
Khatoon [25]	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{P}	\mathcal{F}	\mathcal{P}	\mathcal{P}

S_1 —confidentiality, S_2 —integrity, S_3 —privacy, S_4 —mutual authentication, S_5 —availability, S_6 —non-repudiation, S_7 —authorization, S_8 —forward security, S_9 —anonymity, S_{10} —non-traceability, \mathcal{P} —provides, \mathcal{F} —fails to provide

3.1 Analysis of Security Attributes

It is desired that the security protocol for developed for IoMT should satisfy the security properties of integrity, privacy, confidentiality, authorization, authentication, availability, and non-repudiation as listed in Fig. 2. A comparison of security attributes offered by the ECC-based schemes and protocols for IoMT mentioned in [9, 14, 17–20, 25] has been performed, and the highlights of this comparison are presented in Table 2.

3.2 Analysis of Attack Resistance Capability

In addition to satisfy security properties mentioned in Table 2, the security protocol for IoMT must also be capable of resisting attacks against it. More specifically, denial of service, impersonation, session-specific temporary information, insider, modification, replay, stolen verifier, and man-in-the-middle attack must be countered by an IoMT security scheme. A relative study of the attack confrontation capability of the IoMT security protocols in [9, 14, 17–20, 25] has been done and given in Table 3.

3.3 Computational Cost Analysis

The two major parties in the IoMT authentication process are the client device (CD) and the application provider (AP). In this computational cost analysis, only the time

Table 3 Attack resistance ability of ECC-based IoMT protocols

Protocol	Attack resistance ability									
	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3	\mathcal{A}_4	\mathcal{A}_5	\mathcal{A}_6	\mathcal{A}_7	\mathcal{A}_8	\mathcal{A}_9	\mathcal{A}_{10}
Zhao [9]	C	C	F	C	C	C	C	C	C	C
Das [14]	C	F	C	C	C	C	F	C	C	C
He [17]	C	F	F	F	C	C	F	C	F	F
Sowjanya [18]	C	F	C	F	C	F	C	C	C	C
Li [19]	C	F	C	C	C	C	F	C	F	C
Sowjanya [18]	C	F	C	F	C	F	C	C	C	C
Pirmoradian [20]	C	C	C	C	C	C	C	C	C	C
Khatoun [25]	C	C	F	C	C	C	C	C	C	C

\mathcal{A}_1 —denial of service attack, \mathcal{A}_2 —device impersonation attack, \mathcal{A}_3 —session-specific temporary information attack, \mathcal{A}_4 —insider attack, \mathcal{A}_5 —modification attack, \mathcal{A}_6 —replay attack, \mathcal{A}_7 —man-in-the-middle attack, \mathcal{A}_8 —stolen verifier attack, \mathcal{A}_9 —server impersonation attack, \mathcal{A}_{10} —key security attack, C—counters, F—fails to counter

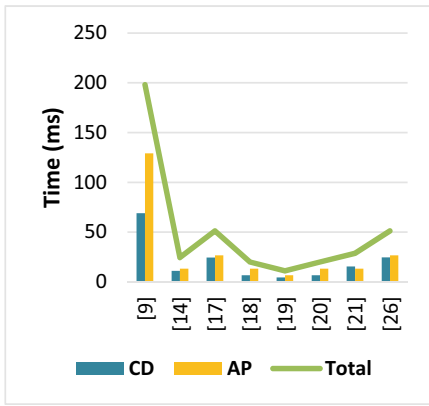
consumed at the side of CD and AP in mutual authentication process has been considered because the registration and other steps are executed once only. In has been established that on PXA270 processor of 624 MHz having 128 MB memory it takes 20.04 ms to execute one bi-linear pairing operation, 2.21 ms for a single point multiplication on elliptic curve, 0.002 ms for hash computation, and 0.005 ms in computing to compute a single symmetric encryption/decryption operation [18]. Based on these facts, an analytical comparison of the computational time of the IoMT authentication schemes in [9, 14, 17–20, 25] has been performed and illustrated in Table 4. The graphical representation of the computational time analysis is shown in Fig. 3a. The computational cost has been computed after counting the number of pairing operations, point multiplication on elliptic curve, hash computation, and encryption/decryption operations for both CD and AP. The other operations have been ignored assuming that they incur negligible time as compared to these key operations.

3.4 Communication Cost Analysis

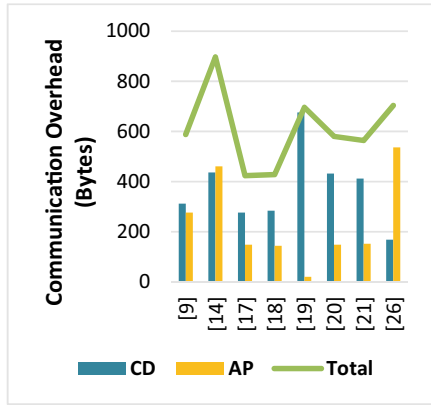
The communication overhead involved in the mutual authentication process of IoMT security protocol can be calculated by identifying the number of bits transmitted between both the parties in the communication, i.e., the CD and the AP. Here, the communication overhead involved in the IoMT authentication schemes mention in [9, 14, 17–20, 25] has been analyzed, and the summary of this analysis has been presented in Table 5. The pictorial representation of the analysis in Table 5 is mentioned in Fig. 3b. While comparing the communication cost of the ECC-based IoMT schemes, the parameter sizes mentioned in [18] have been considered.

Table 4 Computational cost of ECC-based IoMT protocols

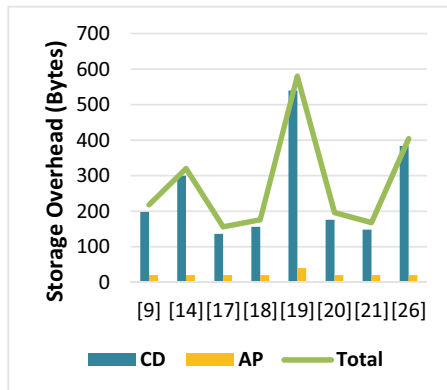
Protocol	Computational time (ms)		
	CD	AP	Total
Zhao [9]	68.973	129.095	198.068
Das [14]	11.054	13.268	24.322
He [17]	24.464	26.674	51.138
Sowjanya [18]	6.632	13.262	19.894
Li [19]	4.429	6.634	11.063
Sowjanya [18]	6.639	13.271	19.91
Pirmoradian [20]	15.481	13.271	28.752
Khattoon [25]	24.469	26.679	51.148



(a) Computational cost analysis



(b) Communication cost analysis



(c) Storage cost analysis

Fig. 3 Cost analysis of different IoMT authentication protocols

Table 5 Communication cost and storage cost of ECC-based IoMT protocols

Protocol	Communication cost (Bytes)			Storage cost (Bytes)		
	CD	AP	Total	CD	AP	Total
Zhao [9]	312	276	588	198	20	218
Das [14]	436	461	897	300	20	320
He [17]	276	148	424	136	20	156
Sowjanya [18]	284	144	428	156	20	176
Li [19]	676	20	696	540	40	580
Sowjanya [18]	432	148	580	176	20	196
Pirmoradian [20]	412	152	564	148	20	168
Khatoon [25]	168	536	704	384	20	404

3.5 Storage Cost Analysis

The storage overhead is the estimate of the memory needed to store different data items within a device. In the mutual authentication schemes for IoMT, different data items and global system parameters are to be stored in the memory of CD as well as the AP. Based on the parameters mentioned in [18], a comparative analysis of the storage requirements of various ECC-based IoMT authentication schemes in [9, 14, 17–20, 25] has been carried out and summarized in Table 5 and also illustrated graphically in Fig. 3c.

T_{PM} —ECC point multiplication, T_P —pairing computation, T_H —hashing, $T_{E/C}$ —symmetric encryption/decryption

4 Discussion

In this section, a discussion over the information presented in the analysis given in Tables 2, 3, 4, and 5 has been made. From Table 2, it is clear that only the scheme of Pirmoradian et al. [20] satisfies all the security attributes, while other schemes in [9, 14, 17–19, 25] fail to satisfy at least one. Similarly, Table 3 shows that none of the schemes in [9, 14, 17–19, 25] can counter all the attacks over IoMT except the scheme of in [20]. Therefore, scheme in [20] is the safest scheme to be used for IoMT authentication. However, from the view point of computational time, Table 4 tells that the scheme of Li et al. [19] consumes minimum computational cost as compared to the other schemes. It can be deduced by Table 5 that the protocol of He et al. [17] least storage cost as well as the communication overhead in comparison with the other schemes in [9, 14, 18–20, 25]. Thus, there is a clear trade-off between the fulfillment of security functions and the costs associated with the IoMT authentication protocols. By this discussion, the researchers and developers can carry out an insight that the problem of developing secure and efficient IoMT mutual authentication mechanism is still open.

5 Conclusion

Security in IoMT communications has always been a challenge for both practitioners and researchers. IoMT security has gained a lot of attention, but constructing a trusted mutual authentication protocol has remained challenging since security requirements must be balanced against computational and communication expenses. Recent ECC-based IoMT security schemes' performance has been examined in this paper with respect to the security requirements they meet, the threats they thwart, computational time, storage overhead, and communication cost. The comparative research reveals that while some schemes are expensive, the majority are susceptible to security concerns. The findings support the conclusion that there is a clear trade-off between the cost of the IoMT authentication procedures and the achievement of security functions. Therefore, the researchers are still working on the challenge of establishing a reliable and effective IoMT mutual authentication system. The future scope of this research is to develop robust and efficient IoMT authentication mechanism using latest techniques like blockchain and hyper-elliptic curve cryptography.

References

1. Ashfaq Z, Rafay A, Mumtaz R, Zaidi SMH, Saleem H, Zaidi SAR, Mumtaz S, Haque A (2022) A review of enabling technologies for Internet of Medical Things (IoMT) ecosystem. *Ain Shams Eng J* 13(4):101660
2. Papaioannou M, Karageorgou M, Mantas G, Sucasas V, Essop I, Rodriguez J, Lymberopoulos D (2020) A survey on security threats and countermeasures in Internet of Medical Things (IoMT). *Trans Emerg Telecommun Technol*: e4049
3. Alsaeed N, Nadeem F (2022) Authentication in the Internet of Medical Things: taxonomy, review, and open issues. *Appl Sci* 12(15):7487
4. Sun Y, Lo FPW, Lo B (2019) Security and privacy for the internet of medical things enabled healthcare systems: a survey. *IEEE Access* 7:183339–183355
5. Hireche R, Mansouri H, Pathan ASK (2022) Security and privacy management in Internet of Medical Things (IoMT): a synthesis. *J Cybersecur Privacy* 2(3):640–661
6. Singh AK, Patro BDK (2020) Elliptic curve signcryption based security protocol for RFID. *KSII Trans Internet Inf Syst (TIIS)* 14(1):344–365
7. Singh AK, Patro BDK (2017) Performance comparison of signcryption schemes—a step towards designing lightweight cryptographic mechanism. *Int J Eng Technol (IJET)* 9(2)
8. Yeh CK, Chen HM, Lo JW (2013) An authentication protocol for ubiquitous health monitoring systems. *J Med Biol Eng* 33(4):415–419
9. Zhao Z (2014) An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem. *J Med Syst* 38:1–7
10. Kumar P, Lee SG, Lee HJ (2012) E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks. *Sensors* 12(2):1625–1647
11. He D, Kumar N, Chen J, Lee CC, Chilamkurti N, Yeo SS (2015) Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimedia Syst* 21:49–60
12. Amin R, Islam SH, Biswas GP, Khan MK, Kumar N (2016) A robust and anonymous patient monitoring system using wireless medical sensor networks. *Futur Gener Comput Syst* 80:483–495

13. Jiang Q, Ma J, Yang C, Ma X, Shen J, Chaudhry SA (2017) Efficient end-to-end authentication protocol for wearable health monitoring systems. *Comput Electr Eng* 63:182–195
14. Das AK, Wazid M, Kumar N, Khan MK, Choo KKR, Park Y (2017) Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE J Biomed Health Inform* 22(4):1310–1322
15. Chaudhry SA, Yahya K, Al-Turjman F, Yang MH (2020) A secure and reliable device access control scheme for IoT based sensor cloud systems. *IEEE Access* 8:139244–139254
16. Liu J, Zhang Z, Chen X, Kwak KS (2013) Certificateless remote anonymous authentication schemes for wirelessbody area networks. *IEEE Trans Parallel Distrib Syst* 25(2):332–342
17. He D, Zeadally S, Kumar N, Lee JH (2016) Anonymous authentication for wireless body area networks with provable security. *IEEE Syst J* 11(4):2590–2601
18. Sowjanya K, Dasgupta M, Ray S (2021) Elliptic curve cryptography based authentication scheme for internet of medical things. *J Inf Secur Appl* 58:102761
19. Li X, Peng J, Kumari S, Wu F, Karupiah M, Choo KKR (2017) An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. *Comput Electr Eng* 61:238–249
20. Pirmoradian F, Safkhani M, Dakhilalian SM (2023) ECCPWS: an ECC-based protocol for WBAN systems. *Comput Netw*: 109598
21. Singh AK, Nayyar A, Garg A (2022) A secure elliptic curve based anonymous authentication and key establishment mechanism for IoT and cloud. *Multimedia Tools Appl*: 1–52
22. Singh AK, Patro DB (2019) A novel security protocol for wireless sensor networks based on elliptic curve Signcryption. *Int J Comput Netw Commun (IJCNC)* 11
23. Singh AK, Patro BDK (2020) Signcryption-based security framework for low computing power devices. *Recent Adv Comput Sci Commun (Formerly: Recent Patents Comput Sci)* 13(5):845–857
24. Singh AK, Solanki A, Nayyar A, Qureshi B (2020) Elliptic curve signcryption-based mutual authentication protocol for smart cards. *Appl Sci* 10(22):8291
25. Khatoon S, Rahman SMM, Alrubaian M, Alamri A (2019) Privacy-preserved, provable secure, mutually authenticated key agreement protocol for healthcare in a smart city environment. *IEEE Access* 7:47962–47971

Water Quality Monitoring and Evaluation Using Internet of Things and Machine Learning



Pravin Vilasrao Sawant and Y. M. Patil

Abstract Water is an important factor for all the living creatures. There are several factors that can affect water quality, both natural and artificial processes involved. Environment, aquatic ecosystems, and human health can all be negatively impacted by poor water quality. This paper presents real-time water quality monitoring using wireless sensor network and evaluation using machine learning model. Water quality is monitored with help of sensors like pH, temperature, dissolved oxygen, and turbidity. Collected data is sent on cloud platform with the help of NodeMCU ESP 8266 module using http protocol. Water parameters are monitored on ThingSpeak platform and simultaneously passed to machine learning (ML) model which is deployed in Amazon web services' (AWSs) platform. ML model is used to determine water quality without human interventions. This ML model detects real-time contamination in water and also infers which parameter is responsible for contamination. Machine leaning model using decision tree algorithm provides average accuracy of 98.28%.

Keywords Internet of Things · Machine learning model · AWS

1 Introduction

Water quality refers to the physical, chemical, and biological condition of water. There are different water parameters like dissolved oxygen, pH, amount of chloride, temperature, electrical conductivity, and turbidity. For drinking water, permissible range of each water parameter is given by World Health Organization (WHO). In earlier system, the water quality is monitored using sensors and decision has

P. V. Sawant (✉)

Department of Technology, Shivaji University Kolhapur, Kolhapur, Maharashtra, India
e-mail: pravin1173@gmail.com

Y. M. Patil

KIT College of Engineering, Kolhapur, Maharashtra, India

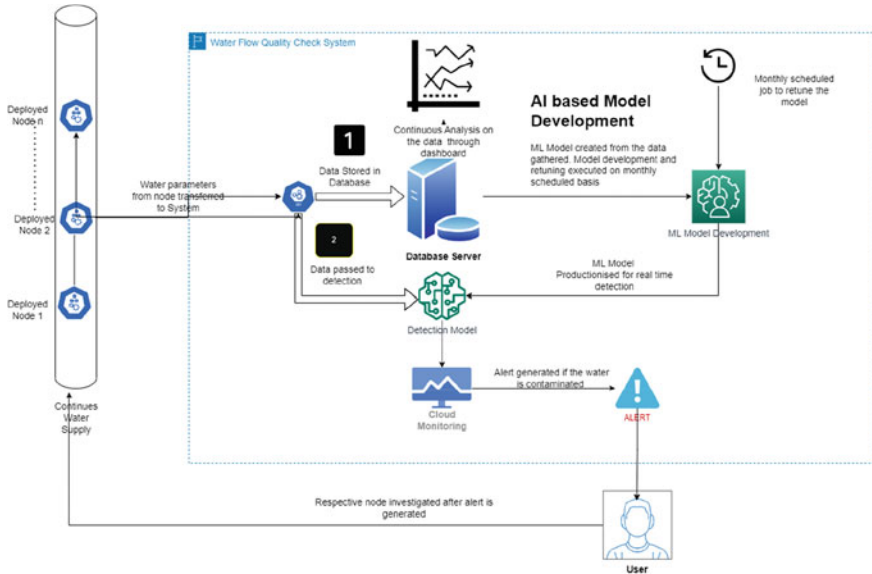


Fig. 1 System architecture of water quality monitoring and evaluation

been made manually by referring water parameter standard. This process is time-consuming and requires continuous human intervention. The architecture of proposed system is shown in Fig. 1. Sensor nodes deployed in water source collect water parameters. These parameters are processed by processing unit and sent on cloud platform. Cloud platform stores this data for continuous analysis as well as evolution of machine learning model as shown in Fig. 1. This data simultaneously hits machine learning model deployed on Amazon web services (AWSs) and indicates that water is safe for drinking or contaminated. If water is contaminated then it identifies suspected water parameters. In this system, machine learning model is continuously evolved by variety of data provided to it by proposed system. The proposed system is only used for Internet of Things application. In case of network failure, recorded water parameters by system are stored in data logger.

- This system uses Internet of Things technology and machine learning to determine the quality of water.
- The system also indicates water parameters responsible for contamination.

2 Background

An indexing approach was introduced [1] for water quality assessment. There are different machine leaning and deep learning algorithms which are used. Water samples for development of model have been taken from wells in North Pakistan.

Data consists of limited water parameters. Machine learning model provides highly accurate results. This system is basically designed to take preventive actions after contamination in water source. Water quality monitoring system is developed to monitor water quality of Ghatprabha River [2]. This system consists of pH, temperature, dissolved oxygen, electrical conductivity, biochemical oxygen demand, nitrate, and total dissolved solids sensors to collect real-time water parameters of river. Internet of Things technology is used as it overcomes on traditional water quality monitoring system. Linear regression model is used to indicate that water is safe or unsafe and find out relation between different water parameters.

A network for monitoring and assessing water quality [3] was developed. A system uses LoRa protocol for wide connectivity. There are total of four layers in this system. pH, turbidity, temperature, and conductivity sensors are part of the sensing layer. The edge layer is responsible for processing data from sensors. The cloud layer is in charge of applying machine learning to classify water samples, while the application layer handles the user interface and provides the ultimate judgment on whether the water is suitable for drinking or irrigation. SVM, logistic regression, and random forest are the three algorithms that were used for classification. Logistic regression was best fit for drinking water and SVM was suited for irrigation water. A system is developed for rural water monitoring based on wireless sensor network [4]. Using pH, dissolved oxygen, and conductivity sensors, water quality has been observed. The water parameters that are obtained from the actual sensor node are compared to the findings of the lab tests. Average error range for dissolved oxygen is 2.0%, and for pH sensors, it is in range of 1.08%–1.86%. The GPRS module is used to transport data to the server due to issues with Internet connectivity in rural areas. A system is proposed using decision tree and support vector machine for anomaly detection in water distribution networks [5]. Water treatment plant provides the database needed to develop machine learning models. All physiochemical and microbiological samples were taken from Tunisian rivers. This paper gives performance analysis of decision tree and support vector machine. Linear SVM found more adequate for water quality monitoring systems. Data is insufficient for development of machine learning model. Water quality monitoring system using IOT and machine learning [6] system collects temperature and turbidity of water. This data is sent on cloud for continuously monitoring. System contains LM35, DS18B20 (temperature sensor), microcontroller unit, web application, power Bi Azure Stream. Linear regression model is used to detect anomaly in water. Water parameters are stored and displayed on Microsoft Azure platform. Designed system is less accurate. Use of Internet technology for transmission of data on cloud platform is not mentioned in this paper. IoT for automated water quality monitoring system [7]. Designed system is used to monitor water quality of river. System uses Raspberry Pi 3, YST 600 K water sensor, 4G, and IoT technology. Conductivity, dissolved oxygen and pH these water parameter are transmitted on cloud using MQTT protocol. In the water monitoring application user knows the status of system, whether offline (sensor connections are not ready) or online. In the next stage of research, authors wish to combine the IoT platform and big data system to classify river water quality.

The system is designed to find out real-time contamination in water pipeline [8]. The system measures pH, conductivity, turbidity, temperature, dissolved oxygen and transmitted to cloud platform with the help of ZigBee module. Multiple sensor nodes are used to detect contamination source. Master node generates alerts when water level from slave sensor nodes is not appropriate. There is no use of any algorithm for detection of contamination.

The proposed system is designed using the ATMEGA 32 microcontroller, along with pH, turbidity, conductivity, and temperature sensors [9]. Water parameters are transmitted with the help of Bluetooth module on smart phone. The signal conditioning circuit is designed for sensors used. The paper gives brief idea about calibration and interfacing of sensors to the Arduino board. The accuracy of sensors is measured and compared with actual values. Range of Bluetooth is very less as compared with other wireless devices. The designed system is only limited for domestic use. Cloud-based water contamination system [10] is designed using pH, temperature, and turbidity sensors. Real-time water parameters are stored on cloud using GSM module. System is designed for farmers to find best place for agricultural based on condition of water source. There is no decision-making mechanism provided in this system.

In the existing system, real-time water parameters are captured and calculate water quality index (WQI). There are some limitations of WQI as follows:

- Limited parameters: WQI is based on a limited number of parameters that may not provide a complete picture of water quality.
- Regional variability: WQI may not be applicable to all regions, as the factors that affect water quality may vary depending on the location. Weighting of parameters: WQI assigns weights to different parameters based on their importance, but these weights may not always reflect the true impact of each parameter on water quality.
- Difficulty in interpretation: WQI can be difficult to interpret for non-experts. The numerical values generated by the WQI may not be readily understandable or meaningful to the general public.
- Sampling frequency: WQI is typically based on periodic water quality sampling, which may not capture sudden changes or variations in water quality.
- In existing system, WQI is popular tool used for examining water quality. But it has above limitations, so in this system water parameters have been assessed individually. All the captured water parameters are passed through machine learning model to determine condition of water. A machine learning model examines water quality and displays suspected water parameters if water is contaminated.

3 Materials and Methods

As shown in Fig. 2, this system contains different water sensors to record live water parameters.

pH Sensor: Amount of Potential of Hydrogen (pH) in water is determined by this sensor. Interface pH sensor to ESP8266 with help of signal conditioning device. For

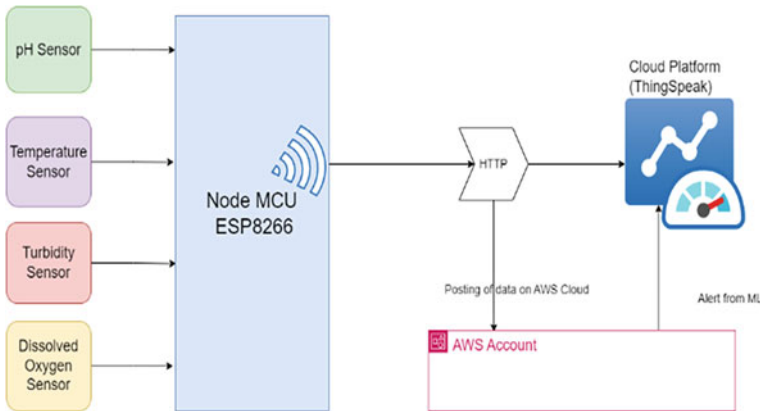


Fig. 2 Block diagram of proposed system

drinking water, range of pH must be in between 6.5 and 8.5. If the pH of water is not in between this range, then water is not suitable for drinking. There is no permissible range for pH. pH value of below 7 is acidic in nature, and pH value above 7 is basic in nature.

Temperature Sensor: PT 100 is Resistance Temperature Detector-based temperature sensor. This temperature sensor provides wide range of temperature. This temperature sensor is either used Whetstones Bridge or Max31865 temperature controller module which is used to get values in an analog form. We need to check other parameters with respect to temperature of water. Some water parameters may vary with respect to temperature.

Turbidity Sensor: It measures amount of suspended particles in water. Turbidity of drinking water should be less than 1. Permissible range for turbidity is below 5 [1].

Oxygen Sensor: The amount of oxygen is an important factor while deciding quality of water. As water contains less amount of oxygen, it means more amount of organic matter present in water. DO in healthy water should be more than 6 mg/l.

There are multiple platforms available for Internet of Things application Intel boards, Raspberry pi and ESP8266 based on application we may choose any one of following (Table 1).

In short, Intel Galileo is better suited for amateurs and hobbyists which need moderate processing power and wireless connectivity, whereas Raspberry Pi is better

Table 1 Comparison of IoT boards

Board	Processor	Connectivity	Power consumption
Raspberry Pi	ARM	Wi-Fi, Bluetooth	Moderate
Intel Galileo	Intel	Wi-Fi, Ethernet	Low to moderate
ESP8266	ESP	Wi-Fi	Low

suit for Internet of Things applications requiring high computing power and wireless connectivity. Applications requiring wireless communication and low power usage can use the ESP8266. Three boards are priced differently, with the ESP8266 being the least expensive. After comparison of above platform, ESP8266 module is better processor for our system. Libraries are available for every sensors used. The necessary requirement is fulfilled by ESP8266 module. This processor reduces overall cost and power of system.

NodeMCU ESP8266—it is a popular option for IoT applications because it is a flexible and reasonably priced microcontroller board with Wi-Fi connectivity and simple programming. It has a huge developer community that has produced libraries and examples, making it simple to add functionality to the board.

3.1 Internet of Things for Real-Time Water Monitoring

Internet of Things is one of the popular technologies. The moto behind this technology is to bringing dark things into light. In this case, real-time water parameters are collected by sensor and given to processing unit. Processing unit is having ability to convert these parameters into readable form and also able to transmit on cloud platform. It consists of all the necessary components for Internet of Things. The data received from different sensors is visualized on ThingSpeak platform as shown in Fig. 3. Indicator has set on ThingSpeak which gives alert to user. As shown in Fig. 3, turbidity of water is continuously monitored; if turbidity of water sample is above 1 NTU, then indicator will turn on. Likewise, all other parameters are visualized and indicators are set for respective water parameter. When particular parameter is influenced, then indicators are used for providing alerts to user. So on ThingSpeak platform water quality will be monitored, but human intervention is required when parameters are changed and abnormal. Collected water data is exported in .csv format. This recorded data is also used to retrain the machine learning model.

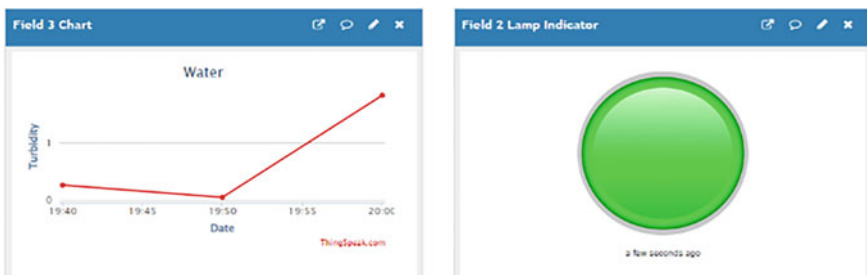


Fig. 3 Turbidity of water sample visualize along with indicator at right side

3.2 AWS Cloud

Provides device management, data analytics, security, and scalability, and is designed to integrate with Amazon’s other facilities. Evaluation of water quality done by ML model is shown in Fig. 4. AWS is used to deploy actual ML model which is in pickle file. Water parameters sent on AWS from NodeMCU are possible using Postman application. This data is collected on Amazon api gateway. This gateway is integrated with AWS lambda, where data from sensors passes through actual ML model. As model hits by real-time data, it produces results as shown in Table 5. This result is transferred to ThingSpeak platform, where user can see condition of water. This result along with current water parameters is stored on AWS database.

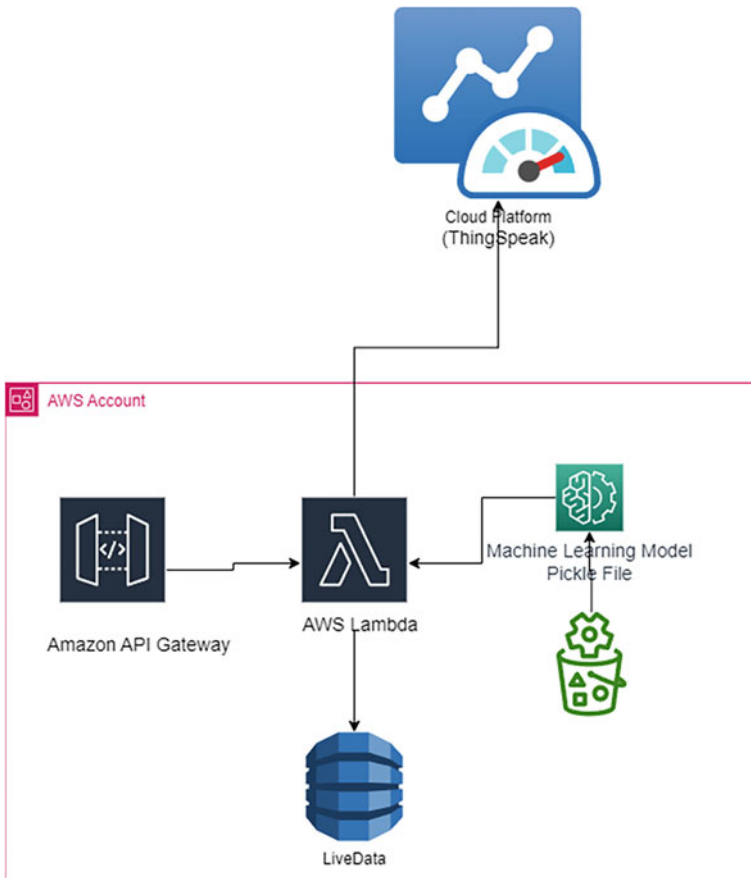


Fig. 4 Deployment of ML model in AWS platform

4 Dataset and Experimental Settings

4.1 Dataset Description

Data for the machine learning model is gathered from the Department of Environment Science, Shivaji University, Kolhapur. The dataset consists of a total of 1000 water samples that were evaluated in 2021 and 2022 in the department. This data is serialized in .csv form.

4.2 Data Labeling

Label is assigned to dataset as shown in Fig. 5. Dataset consists of recorded water samples at Department of Environment Science. Recorded samples are categorized into two classes safe for drinking and unsafe for drinking. This dataset consists of variety of water samples, so water samples which are unsafe are again classified into fifteen different classes. Each unique class indicates result in terms of safe or contamination of water. Figure 5 shows samples with different classes. X-axis represents class and y-axis represents contamination of water with respect to influenced water parameter. Class ‘0’ means water is suitable for drinking. Class ‘1’ means water is contaminated due to pH level of water which is not appropriate.

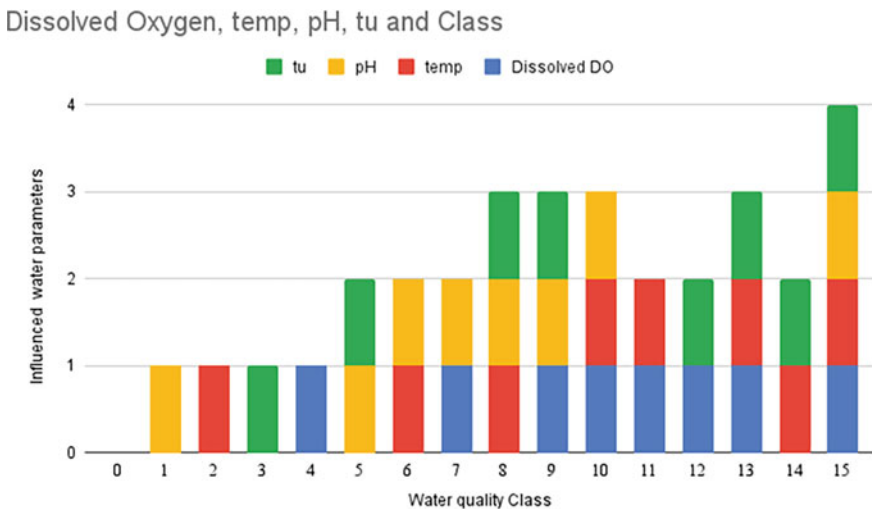


Fig. 5 Variety of data provided to train machine learning model

4.3 Experimental Settings

These examined samples are classified into sixteen unique classes. Using this data and above classification machine learning model is developed with 70% data for training and 30% data for testing. Model is developed and tested in Jupyter Notebook tool.

There are three types of ML algorithms, namely supervised, unsupervised, and reinforcement. In general, water quality determination requires manual intervention to validate whether water is contaminated. Hence, supervised learning algorithms are the best fit for the system. The system contains the following type of labeled data. The decision tree (DT) is one of the supervised learning algorithms. In making decisions, it mimics human reasoning capacity. By considering available data and accuracy of model as shown in Table 4, decision tree is best fit for our system. The system employs a DT model that has been trained using the provided labeled data. Dataset with ratio of 70:30 is used for training and testing of model. It recorded the model accuracy with 98.28%.

A flowchart of the water quality monitoring system is shown in Fig. 6. In the beginning, the data was collected from different sensors. The collected data is processed by the ESP8266 module as appropriate. The data is transmitted by the NodeMCU ESP8266 module on the ThingSpeak platform. This data is also transferred to the AWS cloud platform. The data is organized on the same channel but in different fields. This data is exported in CSV format. Labeling is provided to exported data in accordance with the class defined in Fig. 5. This labeled data is divided into training and testing datasets. A machine learning model is developed in a Jupyter Notebook. The ML model predicted the applied water samples. If all water parameters are in the range mentioned in Table 2, then the water quality is good as per WHO standards. If water parameters are not appropriate, then it indicates the water is contaminated.

5 Results and Discussion

With the use of Internet of Things technology, real-time water parameters have been recorded by the system and visualized, as shown in Figs. 7, 8, 9, and 10. Indicators are also set for water parameters. If dissolved oxygen in water is below threshold level, then indicator will on as shown in Fig. 7. If DO of water is appropriate, then indicator will be in off state. This collected data passed through a machine learning model and produced results as shown in Table 4. When all water parameters are within the range defined in Table 2, the indicator displays the message “Water suitable for drinking”. If water is contaminated, then it examines the reason for contamination and displays an indicator.

Figure 11 indicates the accuracy of different machine learning models. The decision tree provides an accuracy of 98.18% and a low false-positive and false-negative rate. Hence, a decision tree is good for our system.

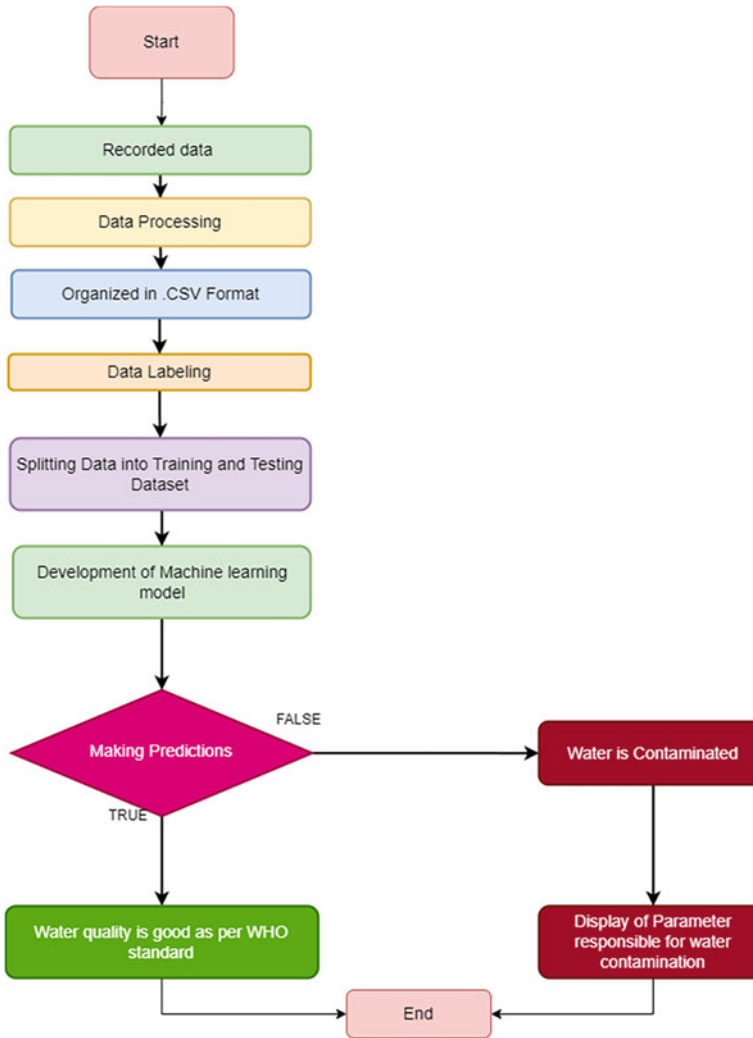


Fig. 6 Flowchart of water quality monitoring system

Table 2 Water parameter permissible range for drinking [11–13]

Water parameter	Permissible range for drinking water
pH	6.5–8.5
Turbidity	0–1 NTU
Dissolved oxygen	Above 6.5 mg/l
Temperature	10–25 °C

Table 4 Results of proposed model

S. No.	pH	DO in Mg/l	Temp. in °C	Turbidity in NTU	Indicator
1	7	8.5	16	1	Water suitable for drinking
2	2.5	7	15	0.9	Water contaminated due to pH level of water is not appropriate
3	7.5	1	20	1	Water contaminated due to oxygen level of water is not appropriate
4	8	7	7	0.8	Water contaminated due to temperature level of water is not appropriate
5	7.9	7.5	16	6	Water contaminated due to turbidity level of water is not appropriate
6	8.9	3.5	18	0.7	Water contaminated due to pH and oxygen levels of water is not appropriate
7	11	4.6	7	1	Water contaminated due to pH, oxygen, and temperature levels of water is not appropriate
8	7.8	6.8	30	3	Water contaminated due to temperature and turbidity levels of water is not appropriate
9	6.9	4	28	5	Water contaminated due to oxygen, temperature, and turbidity levels of water is not appropriate
10	9.9	8	20	4.5	Water contaminated due to pH and turbidity levels of water is not appropriate
11	7.8	2.8	22	3.8	Water contaminated due to oxygen and turbidity levels of water is not appropriate
12	5.5	7.8	7.5	2.5	Water contaminated due to pH, temperature, and turbidity levels of water is not appropriate
13	6.9	5	8.8	1	Water contaminated due to oxygen and temperature levels of water is not appropriate

(continued)

Table 4 (continued)

S. No.	pH	DO in Mg/l	Temp. in °C	Turbidity in NTU	Indicator
14	2.5	8.8	32.5	4	Water contaminated due to pH, temperature, and turbidity levels of water is not appropriate
15	9.9	7.6	29.9	0.7	Water contaminated due to pH and temperature levels of water is not appropriate
16	6	4.3	6.8	4.9	Water contaminated due to pH, temperature, oxygen, and turbidity levels of water is not appropriate

Bold means the proposed algorithm results.

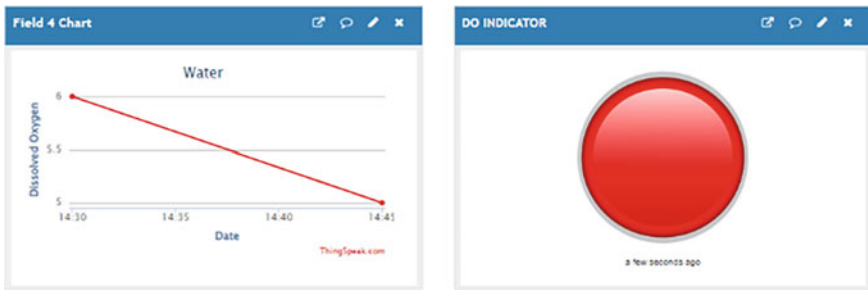


Fig. 7 Visualization of dissolved oxygen sensor in ThingSpeak

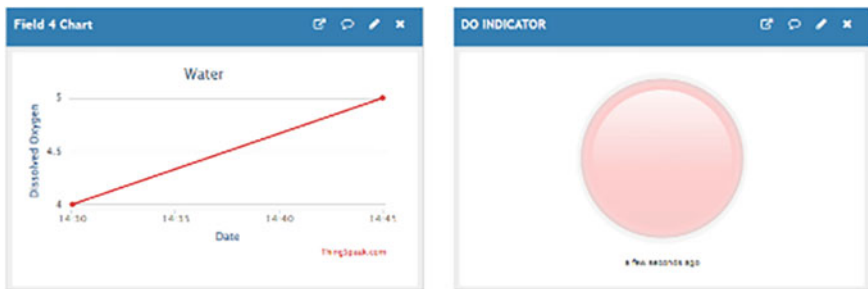


Fig. 8 Visualization of dissolved oxygen level and indicator in ThingSpeak

Sensors need to be calibrated before deployment in water. In this system, pH, temperature, turbidity, and dissolved oxygen sensors are used and out of dissolved oxygen sensor is very costly. After periodic time, sensors should be replaced.

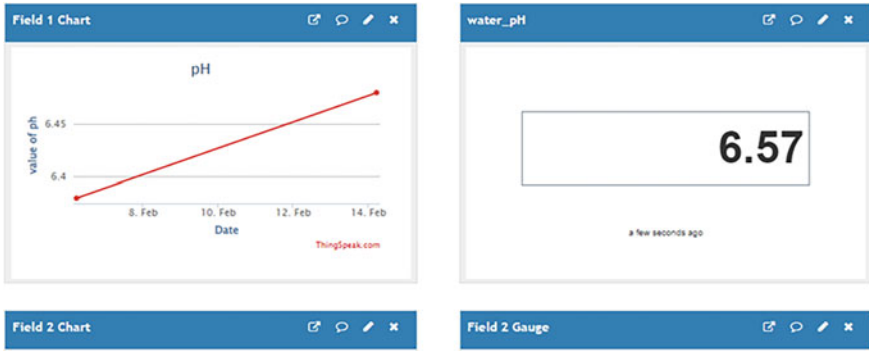


Fig. 9 Visualization pH sensor in ThingSpeak

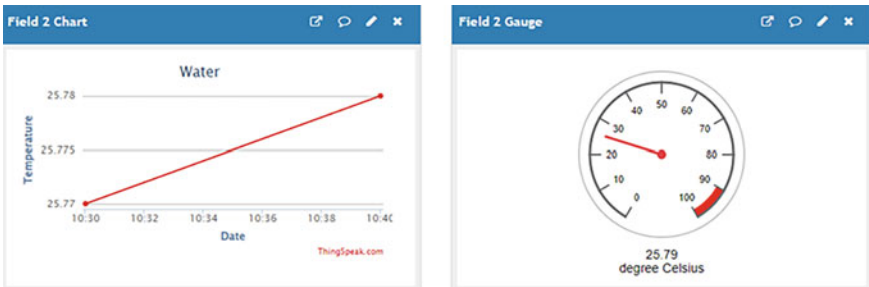


Fig. 10 Visualization of temperature sensor in ThingSpeak

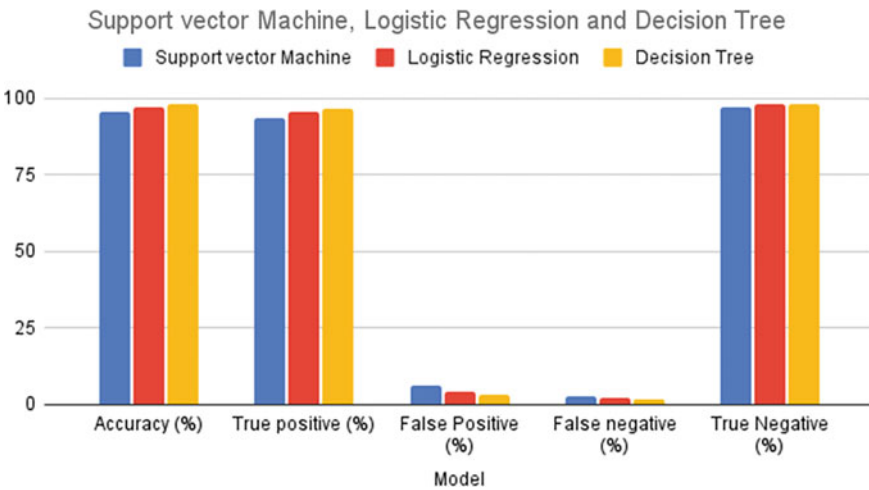


Fig. 11 Results of different models with comparison

6 Conclusion

In this paper, we have proposed water quality monitoring using the Internet of Things and a machine learning model. This technique is highly effective and user-friendly compared with the WQI technique for detecting water contamination. There is no need to calculate WQI. In this system, individual water parameters have been assessed. The evaluation of water quality is done by a machine learning model. The precise water factors that cause pollution are found using this approach. Retraining the model as it evolves continuously produces results that are superior to those of individual tunings. Alerts that are raised can be used to understand trends over time. The model will get more accurate as more and more diverse data is provided to it. The response time of the system is good when we use the Internet of Things and machine learning combination.

Acknowledgements The authors extend their appreciation and gratitude to the Shivaji University, Kolhapur, Maharashtra, for providing required infrastructure and facilities for carrying out this research work.

References

1. Aslam B, Maqsoom A, Cheema AH, Ullah F, Alharbi A, Imran M (2022) Water quality management using hybrid machine learning and data mining algorithms: an indexing approach. *Open Access IEEE 10*
2. Kenchannavar HH, Pujar PM, Kulkarni RM, Kulkarni UP (2022) Evaluation and analysis of goodness of fit for water quality parameters using linear regression through the Internet-of-Things-based water quality monitoring system. *IEEE Internet of Things J* 9(16):14400–14407. <https://doi.org/10.1109/JIOT.2021.3094724>
3. Ajayi OO, Bagula AB, Maluleke HC, Gaffoor Z, Jovanovic N, Pietersen KC (2022) Water net: a network for monitoring and assessing water quality for drinking and irrigation purposes. *Open Access IEEE 10*
4. Lin Z, Yin H, Jiang S, Wang W, Jiao G, Yu J (2017) Design of monitoring system for rural drinking water source based on WSN. In: 2017 international conference on computer network, electronic and automation. *IEEE*. <https://doi.org/10.1109/ICCNEA.2017.106>
5. Jalal D, Ezzedine T (2020) Decision tree and support vector machine for anomaly detection in water distribution networks. *IEEE*. 978-1-7281-3129-0/20/2020
6. Koditala NK, Pandey PS (2018) Water quality monitoring system using IoT and machine learning. *IEEE*. 978-1-5386-2599-6/18/2018
7. Tjahjono RPNBA, Hariadi M, Purnomo MH (2019) Development of IoT for automated water quality monitoring system. In: *Proceedings of ICOMITEE 2019*. *IEEE*, 16th–17th Oct 2019, Jember, Indonesia. 978-1-7281-3436-9/19/2019
8. Kavi Priya S, Shenbagalakshmi G, Revathi T (2018) IoT based automation of real time in-pipe contamination detection system in drinking water. In: *International conference on communication and signal processing*, 3–5 Apr 2018, India. 978-1-5386-3521-6/18/\$31.00©2018
9. Feng C, Yuan J, Sun Y, You J (2020) Design of water quality monitoring system. In: *2020 international conference on artificial intelligence and computer engineering (ICAICE)*. <https://doi.org/10.1109/ICAICE51518.2020.00057>

10. Raghavan SS, Loganathan V, Rathod V, Sharvani GS (2017) Cloud enabled water contamination detection system. In: 2nd IEEE international conference on computational systems and information technology for sustainable solutions. IEEE. 978-1-5386-2044-1/17
11. Indian Standard. Drinking water—specification, 2nd Rev. ICS 13.060.20
12. Water quality requirement for different uses hydrology and water resources information system for India
13. Omar NH. Water quality parameters, water quality science, assessments and policy. Intechopen. <https://doi.org/10.5772/intechopen.89657>

A QoS Enabled Automatic Fallback Handover Mechanism for Future Generation Wireless Networks



Ronitt Mehra, Palash, Reshav Kalyani, and Manjeet Kumar

Abstract The emergence of high-performance machine learning (ML) computing and the Internet of Things (IoT) is driving a paradigm shift in wireless communication. With the increasing use of multimedia applications and high-speed data transfer, there is a significant rise in demand for bandwidth. However, with increasing number of cellular users, the challenge is to effectively manage the limited spectrum allotment for wireless communication while maintaining satisfactory quality of service. Hence, different multiplexing techniques have been used to effectively use the available bandwidth. Recently, the concept of automatic fallback in receivers are gaining popularity due to high mobility in vehicular networks and IoT. Automatic fallback and handover mechanisms often utilize the channel state information (CSI) of the radio and can switch between technologies to provide the best available quality of service for spatial and temporal channel conditions. With the advent of machine learning and deep learning methods, estimating the channel state information has become computationally efficient and feasible thereby improving the performance metrics of the system. OFDM and NOMA are two of the most widely used data transmission methods in modern day networks. This paper presents a Quality of Service (QoS) enabled handover mechanism between OFDM and NOMA for future generation wireless networks. The performance metric for the proposed system is the bit error rate of the system.

Keywords Wireless networks · Handover · Quality of service (QoS) · Channel state information (CSI) · OFDM · NOMA

1 Introduction

Wireless communications beyond 5G have emerged as new paradigm with enormous new possibilities such as metaverse, digital clones, large scale automation and internet of things to name a few [1]. However, all these new age concepts critically depend

R. Mehra (✉) · Palash · R. Kalyani · M. Kumar
Delhi Technological University, Delhi 110042, India
e-mail: mehronitt@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
A. Swaroop et al. (eds.), *Proceedings of Data Analytics and Management*, Lecture Notes in Networks and Systems 787, https://doi.org/10.1007/978-981-99-6550-2_28

365

on the bandwidth availability and spectrum management in wireless networks. As bandwidth is limited, hence, effectively using the bandwidth is critically important to cater to the following needs [2]:

- (1) Increasing number of users.
- (2) Increased bandwidth requirement owing to multimedia data transfer.
- (3) Need for high data rates.
- (4) Limited available bandwidth.

The problem becomes even more critical with the necessity of internet of things (IoT) and fog computing networks where multiple devices are connected over internet and send data to a centralized server [3]. The pervasive nature of future wide area networks, IoT, Fog networks and cellular communications along with the necessity of higher data rate would need compliance to high quality of service (QoS) metrics. While different multiple access techniques are available at our disposal to accommodate the increasing number of users, yet sticking to one technique may not render desirable QoS metrics. Hence, there is an inevitable need for handover mechanisms which can be used to automatically switch from one technique to the other in case one of the technique's parameters starts degrading [4].

Some of the major contributions are as follows: -

- In the paper, we have analyzed the conditions for co-existence of NOMA and OFDM for wireless networks, by primarily establishing a comparable range of BER over varying SNR, for both these techniques, through our simulations.
- We have designed near and far user models based on the path loss co-efficient values. The Far User model considers a higher path loss factor, and the Near User model considers a lower path loss factor.
- We have generated a vertical handover condition for NOMA and OFDM based on the bit error rate of the system. The system created switches from NOMA (Primary Technique) to OFDM (Secondary Technique), as the BER of the system degrades.
- We have attained lesser Bit Error Rate compared to existing work, over a similar range of SNR.

Conventional wireless networks are being re-configured as software defined networks (SDNs). Thus, automatic handover mechanisms would be better suited through fallback in SDNs.

The paper is organized in the subsequent sections: Sect 1 introduces the background of future generation wireless networks, the need of automated handover and its applications to cellular networks, IoT and Fog networks. Section 2 presents the theoretical background for handover under different conditions. Section 3 presents the existing techniques in the domain based on QoS metrics. Section 4 presents the proposed approach followed for generating the simulations. Section 5 presents the simulation results. Section 6 represents concluding remarks and future directions of research.

2 Theoretical Background for Handover

The main objective of handover is to maintain a satisfactory quality of service metric. The outage of the system is measure of the quality of service of the systems. The outage means the chance of unacceptable quality of service. The outage primarily depends on the signal to noise ratio and the bit error rate of the system [5]. The system outage often is represented in terms of the complementary cumulative distribution function or the CCDF. The need for using a probabilistic model for the description of the outage of the system is since neither the BER nor the SNR of the system can be used to ascertain the outage since both are subjective performance metrics [6]. In general, it is shown that the outage is a function of the signal to noise plus interference ratio, the distance, and the channel fading effects. The outage in terms of absolute parameters $q(\lambda)$ is given by [7]:

$$q(\lambda) = \exp \left\{ - \frac{2\pi}{\sin\left(\frac{2\pi}{\eta}\right)} K_k^2 (\text{SINR})_k^{2/\eta} \lambda_i \right\} \quad (1)$$

Here,

$K_k = C_k R_k^2 (\text{SNR})_k^{2/\eta}$ is a constant depending on system and channel parameters. SINR represents the signal to noise plus interference ratio.

λ_j is the device density in a network.

$q(\lambda)$ is the absolute outage.

The major challenge looming large on SDNs is the multipath propagation and varying media (channel conditions) in terms of fading [8]. This results in the following problems [9]: reduced strength resulting in poor quality of service, increased bit and packed error rates resulting in SDN system outage, and large latencies and relatively low throughput.

A typical SDN usually has the ability of handover or automatic fallback. Handover may occur between two systems when the performance of once system starts to deteriorate compared to the other system [10].

3 QoS Enabled Handover

There can be various handover techniques which can be considered in wireless networks. The most predominant ones can be among:

- (1) Multiplexing or multiple access techniques: Common techniques can be orthogonal frequency division multiplexing (OFDM), Orthogonal time frequency space (OTFS) and Non-Orthogonal multiple access (NOMA). OFDM and NOMA often exhibit similar SNR-BER characteristics [11]. Cellular systems commonly feature adaptive fallback or automatic fallback capabilities that allow

for a switch from one technology to alternative coexisting or parallel technology in reply to variations in system parameters, including Bit Error Rate (BER).

- (2) Cellular to device to device (D2D): Rapid increase in the number of users using cellular users has resulted in the increase of load on the cellular network. The base station which routes the data from devices is becoming more and more loaded with data [12]. With evolving technologies of 5G and 6G on the forefront, a new technical solution to the aforesaid problem is inevitable. One of the major contenders for the same is the Device-to-Device Network (D2D) model. In this model, the base station is completely bypassed and the data is communicated among the devices directly.
- (3) Connectivity: While WIFI can be adept to most of the conventional urban environments, yet wide area networks may encounter small as well as large scale fading along with Doppler effects for vehicular networks. For that purpose, WiMax based last mile connectivity measures may be effective [13].
- (4) Spectrum Allocation: While many networks such as IoT and Fog networks may operate in the 2.4 GHz freelance band, yet spectrum congestion and poor channel response may result in opting for switching over to the licensed band [13]

Wireless channels exhibit different behaviors at different frequencies. Channel state information (CSI) denotes to the state of the wireless channel at any given time, which is typically a function of time. The diverse fading effects experienced by signals give rise to varying bit error rate (BER) and outage patterns at the output, depending on the attenuation constant. This attenuation constant relies on material constants of the channel, namely conductivity, permittivity, and permeability as well as the transmission frequency. Hence, the fading pattern is highly dependent on the transmission frequency and the characteristics of the channel, which evolves as channel characteristics change. The attenuation constant can be mathematically expressed as:

$$\alpha = \frac{\omega\pi}{2} \sqrt{\left(\frac{1}{\sigma + \omega\varepsilon}\right)^2 - 1} \quad (2)$$

Here,

α denotes attenuation constant.

ω denotes angular frequency.

μ denotes medium permeability.

ε denotes medium permittivity.

σ denotes medium conductivity.

Meanwhile, α (attenuation constant) depends on ω (angular frequency), which in turns depends on the frequency as per:

$$\omega = 2\pi f \quad (3)$$

Here, f denotes frequency,

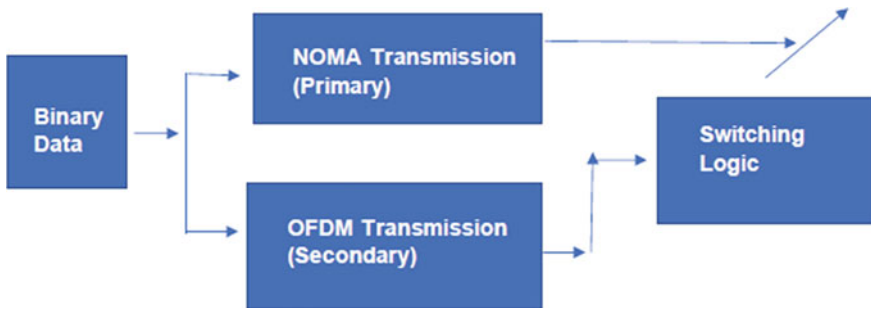


Fig. 1 System architecture

Signal fading effects result in outages and poor quality of service. As far as 5G and onward technologies are concerned, OFDM and NOMA are suitable candidates due to their high spectral efficiency. In case NOMA is the preferred candidate, an automatic fallback candidate can be OFDM. However, the choice of candidates to implement handover should satisfy the conditions of co-existence.

Showing that an identical SNR-BER curve can be achieved using OFDM and NOMA, thereby can justify co-existence of NOMA-OFDM for a cellular network which can lead to a possible vertical handover in case of system requirements. Non-identical BER performance in the SNR range would mean different characteristics for NOMA and OFDM thereby hindering handover. It has been discussed that a major challenge of NOMA based multiple access technique is the fact that small scale fading effects and multipath propagation make the amplitude of the power variable at the receiving end. This results in difficulty of separating the signals of different users with equal reliability. The metric for obtaining equal reliability and quality of service (QoS) is bit error rate (BER) of the system. The mathematical condition for the handover process can be represented as:

Estimate the BER of the system for NOMA and OFDM (Fig. 1).

$$\text{if}(\text{BER}_{\text{NOMA}} < \text{BER}_{\text{OFDM}})$$

{ Choose NOMA as the transmission technique }

else

{ Fall back to OFDM }

4 Proposed Approach

- 1 The first step involves generation of the data to be transmitted. Random complex signals are generated to practically represent the generation of signals to be sent by the transmitter.

$$x(t) = K_1 \cos(\omega t) + j K_2 \sin(\omega t)$$

- 2 1,600,000 bits have been considered for transmission. The length of a data frame has been chosen to be 32 bits.
- 3 The serial data is converted into parallel data. The data is then converted into time domain. Further, cyclic prefixes are added to prevent loss of information.
- 4 The conversion of signal to Time Domain has been made possible with Inverse FFT, allowing for the generation of orthogonal carriers.
- 5 The output of the channel for the transmitted data is then calculated and polluted with AWGN noise.
- 6 The resultant signal is converted to frequency domain, and frequency-domain channel equalization has been performed with the channel. After equalization and subsequent QAM decisions for decoding at the receiver, the BER is computer for varying values of SNR (in dB). BER computed using below formula.

$$\left(\sum_{i=1}^n \text{Bit}_{RX} \neq \text{Bit}_{TX} \right) \forall \text{ all bits } n \text{ as a function of SNR}$$

$$\text{BER} = \frac{\text{Number of error Bits}}{\text{Total Number of Bits}}$$

- 7 For the near and far user scenarios, adjustments must be made to the analytical BER. For Far user, naturally, the path loss is greater than that for the near user.
- 8 A multi-path channel has been created for NOMA using matrix, wherein 1 denotes no reflection and -1 denotes reflection. A random channel is also generated for NOMA.
- 9 The power matrix is then generated, and the Kronecker Tensor Product is evaluated for the random complex signals generated and the power matrix. The tensor product provides all the possible combinations for a bit.
- 10 AWGN noise is added to the channel output, after which the de-spreading and the decorrelation has been done.
- 11 Now, SIC Algorithm is used at the receiver end to obtain the signals at the receiver. And, as the algorithm suggests, the signals are obtained from the strongest to the weakest.
- 12 Finally, the analytical BER for NOMA is computed. Plots are generated for co-existence of OFDM and NOMA (non-overlapping curve), for NOMA-OFDM handover in near user scenario and for NOMA-OFDM handover in far user scenario.

5 Results

The following section comprises of the simulation results of the proposed method. The simulations have been run on MATLAB 2019. The handover mechanism has been employed among NOMA and OFDM.

For a practical scenario, as different users send their signals at different power levels. In general, the user farthest away from the receiver would face the most severe fading. On the contrary, the user nearest to the base station would face the least fading, Hence, the users can be categorized into 2 major categories which are: Near and Far User.

The near and far user situation can be differentiated based on the path loss factor. The noise condition considered in this case is AWGN with a constant noise psd for all frequencies. Mathematically,

$$\text{Noise}_{psd} = \frac{N_0}{2} \forall f$$

Here,

psd denotes the power spectral density.

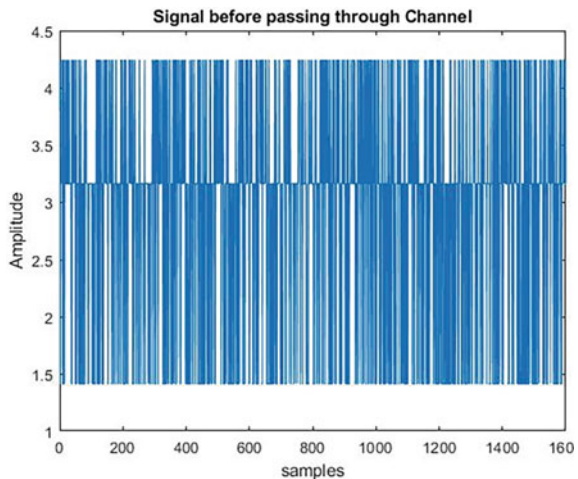
f signifies frequency.

$\frac{N_0}{2}$ characterizes the two-sided AWGN psd .

Based on the automatic fall-back approach, choose the system BER as the metric to decide upon handover. For the near and far user scenario, we obtain the overlapping BER curves. The lesser among the two BER curves would be the technology to use. The results are presented subsequently (Figs. 2, 3, 4 and 5).

Figure6 depicts the performance of BER for far and near users in OFDM. The far users have a higher BER compared to the near users.

Fig. 2 Transmitted signal



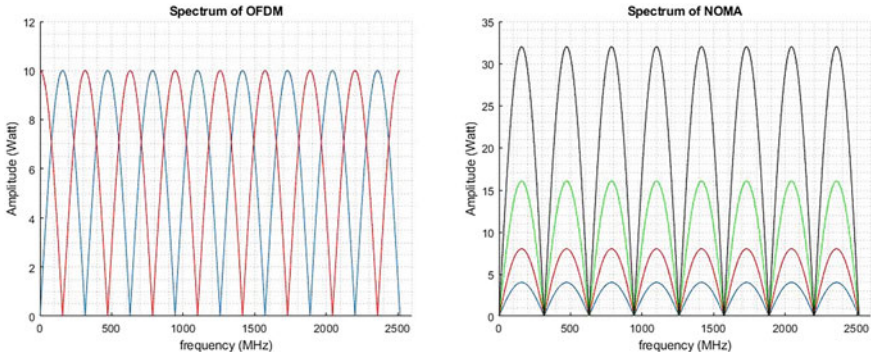


Fig.3 Spectrum of OFDM and NOMA

Fig. 4 Addition of noise in the channel

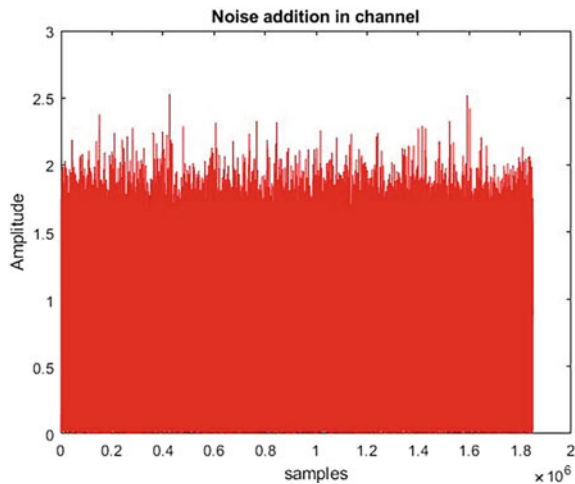


Figure 7 depicts the performance of BER for far and near users in NOMA. As in the case of OFDM, for NOMA to the far users have a higher BER compared to the near users.

Figure 8 shows the BER condition for the co-existence of NOMA and OFDM.

Figure 9 shows the condition for switching among NOMA and OFDM for the near user scenario. It can be seen that prior to the intersection point, NOMA performs better while after the intersection point, OFDM performs better in terms of system BER.

Figure 10 shows the condition for switching among NOMA and OFDM for the far user scenario. A similar pattern is seen with the difference that the BER now falls slower compared to the near user condition implying the fact that SNR is relatively less.

Fig. 5 Signal after passing through channel

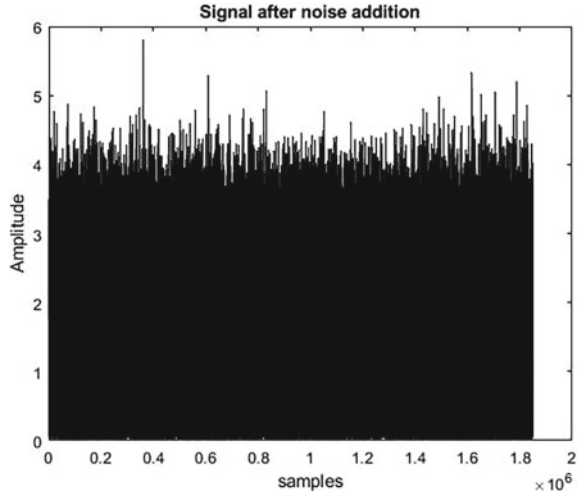
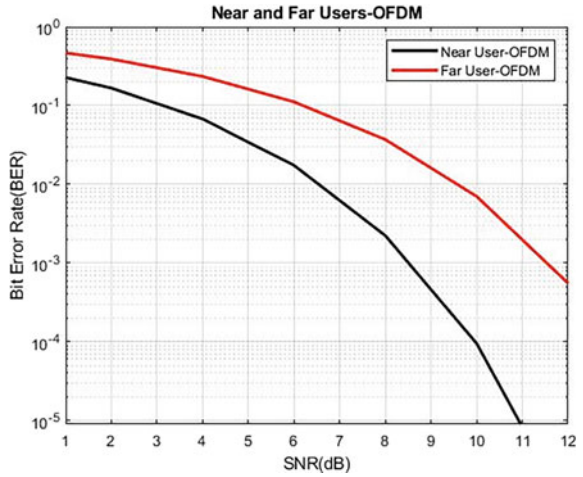


Fig. 6 Near and far user condition for OFDM



6 Comparative Analysis

On comparison with existing work, A. Tusha et al., IEEE 2020 [1] (Table 1).

Hence, lesser SNR requirement needed by at least 3.5 dB by proposed scheme for near user and 9.5 dB for far user.

Fig. 7 Near and Far user scenarios for NOMA

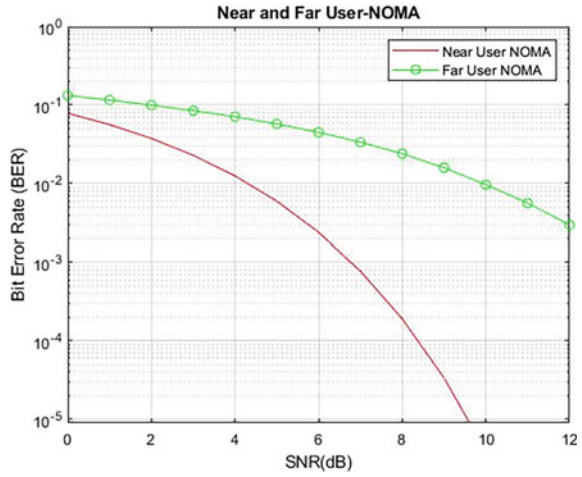
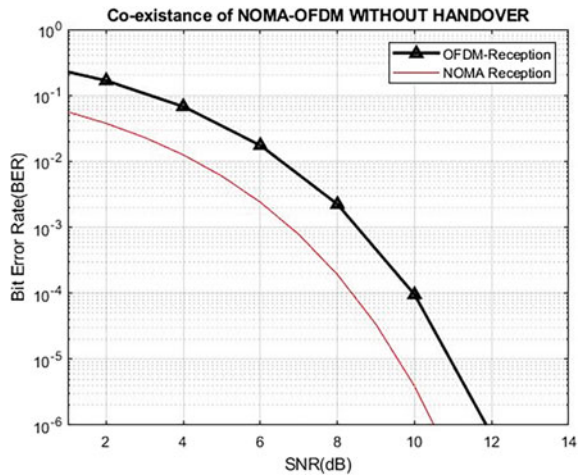


Fig.8 Co-existence of NOMA-OFDM



7 Limitations

The limitations of the proposed work are:

- 1 In the implementation of our proposed scheme, Error Detection through hamming codes has not been considered. Employing hamming codes would help in further reduction of the BER of our proposed system.
- 2 The proposed model does not offer the possibility to switch from a Cellular Network to a Device-to-Device Network. Therefore, in a practical scenario, with a very large number of users in the system, the Base Station will be over-loaded with data.

Fig. 9 Vertical handover for near user condition

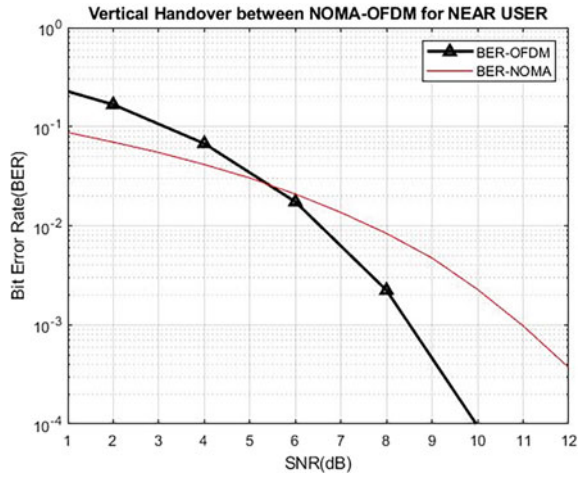


Fig. 10 Vertical handover for far user condition

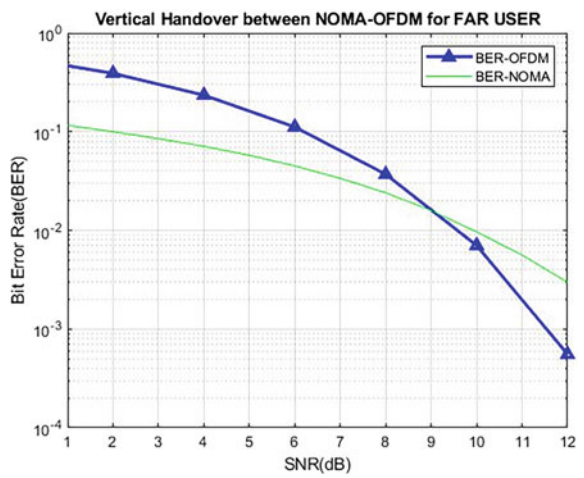


Table 1 Comparison with existing work

Parameter	Value (dB)
SNR range for BER (OFDM-NOMA) (Near User)	12–15
SNR range for BER (OFDM-NOMA) (Far User)	20
SNR range for BER (OFDM-NOMA) (Proposed Technique Near User)	8.5
SNR range for BER (OFDM-NOMA) (Proposed Technique Far User)	11.5

Table 2 Handover characteristics for proposed approach

Parameter	Value (dB)
SNR range for lower BER (NOMA) (Near User)	0–5.3
SNR range for lower BER (OFDM) (Near User)	5.3–12
Switching SNR Value (Near User)	5.3
SNR range for lower BER (NOMA) (Near User)	0–9
SNR range for lower BER (OFDM) (Far User)	9–12
Switching SNR Value (Far User)	9

8 Conclusion

This paper presents a comprehensive review on the current trends in wireless networks pertaining to modulation techniques, handover mechanisms and automatic fallback, fading effective and channel sensing through latest machine learning and deep learning algorithms for cognitive networks. Moreover, IoT (internet of things), device to device networks, fog computing and their co-existence in underlay cellular networks have also been discussed.

Channel sensing mechanisms through channel sensing and estimation has also been cited and discussed in detail. It has been shown that in case of non-intersecting BER curves, the condition remains to be that of non-handover since one of the techniques for transmission continuously outperforms the other in terms of the performance metric (BER). In case of handover, concurrent BER curves for OFDM and NOMA intersect to create a point of intersection (Table 2).

9 Future Scope

Future enhancements of the proposed work can be:

- Devising a strategy to switch between cellular and device to device (D2D) modes of transmission.
- Employing error detection and correction codes along with the proposed scheme

References

1. Thompson J et al (2014) 5G wireless communication systems: prospects and challenges [Guest Editorial]. *IEEE Commun Mag* 52(2):62–64. <https://doi.org/10.1109/MCOM.2014.6736744>
2. Albreem MAM (2015) 5G wireless communication systems: vision and challenges. In: 2015 international conference on computer, communications, and control technology (I4CT), pp 493–497. <https://doi.org/10.1109/I4CT.2015.7219627>

3. Vaezi M et al. (2022) Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G. *IEEE Commun Surv Tutor* 24(2):1117–1174, Secondquarter. <https://doi.org/10.1109/COMST.2022.3151028>
4. Liu S, Yu G, Wen D, Chen X, Bennis M, Chen H Communication and energy efficient decentralized learning over D2D networks. *IEEE Transact Wireless Commun*. <https://doi.org/10.1109/TWC.2023.3271854>
5. Sönmez Ş, Shayea I, Khan SA, Alhammadi A (2020) Handover management for next-generation wireless networks: a brief overview. *2020 IEEE Microwave theor tech wireless commun (MTTW)* 35–40, <https://doi.org/10.1109/MTTW51045.2020.9245065>
6. Novlan TD, Ganti RK, Ghosh A, Andrews JG (2012) Analytical evaluation of fractional frequency reuse for heterogeneous cellular networks. *IEEE Trans Commun* 60(7):2029–2039. <https://doi.org/10.1109/TCOMM.2012.061112.110477>
7. Zhang H, Wen X, Wang B, Zheng W, Sun Y (2010) A novel handover mechanism between Femtocell and macrocell for LTE based networks. *Second Int Conf Commun Softw Netw* 2010:228–231. <https://doi.org/10.1109/ICCSN.2010.91>
8. Yannuzzi M, Milito R, Serral-Gracià R, Montero D, Nemirovsky M (2014) Key ingredients in an IoT recipe: fog computing, cloud computing, and more fog computing, In: *2014 IEEE 19th international workshop on computer aided modeling and design of communication links and networks (CAMAD)* 325–329. <https://doi.org/10.1109/CAMAD.2014.7033259>
9. Alrawais A, Alhothaily A, Hu C, Cheng X (2017) Fog computing for the internet of things: security and privacy issues. *IEEE Internet Comput* 21(2):34–42, Mar–Apr. <https://doi.org/10.1109/MIC.2017.37>
10. Han S et al (2020) Artificial-intelligence-enabled air interface for 6G: solutions, challenges, and standardization impacts. *IEEE Commun Mag* 58(10):73–79. <https://doi.org/10.1109/MCOM.001.2000218>
11. Nain G, Das SS, Chatterjee A (2018) Low complexity user selection with optimal power allocation in downlink NOMA. *IEEE Wireless Commun Lett* 7(2):158–161. <https://doi.org/10.1109/LWC.2017.2762303>
12. Guerreiro J, Dinis R, Montezuma P, Campos M (2020) On the receiver design for nonlinear NOMA-OFDM systems. In: *2020 IEEE 91st vehicular technology conference (VTC2020-Spring)* pp 1–6. <https://doi.org/10.1109/VTC2020-Spring48590.2020.9129559>
13. Cai Y, Qin Z, Cui F, Li GY, McCann JA (2018) Modulation and multiple access for 5G networks. *IEEE Commun Surv Tutor* 20(1):629–646, Firstquarter. <https://doi.org/10.1109/COMST.2017.2766698>

Bone Fracture Detection Using CNN



Sai Prudhvi Vallurupalli and T. Anuradha

Abstract Bone fractures are a prevalent issue faced by many individuals, often resulting from accidents. X-rays are commonly used by doctors to predict fractures. However, manually interpreting X-rays can be challenging, as small fractures may be overlooked, leading to potential future harm. This project aims to address this issue by utilizing artificial intelligence applications, such as ML and DL techniques, to analyze and classify images of hand, leg, chest, fingers, and wrist fractures in a precise manner. Specifically, Convolutional Neural Networks (CNNs) are employed to develop various models that offer a step-by-step image analyzing algorithm to accurately predict whether a bone is fractured or normal, providing a better solution for fracture detection.

Keywords Fracture classification · Convolution neural network (CNN) · ReLU and Softmax activation functions · Bone fracture · Fracture · Bones-rays · X-rays

1 Introduction

Fractures are a common issue faced by all human beings, and even minor fractures can lead to more serious injuries if overlooked by doctors. Although X-rays are available, it can still hard to determine if a bone is broken or not by normal persons. It is crucial to treat a fracture bone as a medical emergency and seek care promptly. Fractures can vary in shape and size. Traumatic fractures are more common and can be caused by falls, high pressure, fights, accidents, or other reasons, while pathological fractures are caused by underlying medical conditions. The goal of this project is to find the most accurate CNN model, which is a step-by-step picture analyzing algorithm that can provide improved results in detecting bone fractures. The project involves analyzing and classifying X-ray images of hand, leg, chest, fingers, and wrist fractures, as well as normal images, to achieve clearer and more accurate results.

S. P. Vallurupalli (✉) · T. Anuradha
Department of IT, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, AP,
India
e-mail: vsprudhvi1999@gmail.com

Motivation: Fractures can occur as a result of sudden accidents such as falls, vehicle collisions, or sports injuries. In these cases, immediate symptoms may include swelling and severe pain. However, it is important to note that not all accidents will result in fractures that require treatment with a cast or splint. Some fractures may resolve within 2–3 days and can be managed with pain balms and tablets.

Identifying whether a bone is fractured or not requires a medical examination by a doctor, usually involving an X-ray and a consultation fee. Waiting for 2–3 days to see if it is a fracture or not can potentially worsen the condition and lead to more severe consequences. Additionally, it is worth noting that in some cases, doctors may not be immediately available, especially if the accident occurs during odd hours.

Contribution: This paper introduces a user-friendly web application that utilizes deep learning to detect fractures in X-ray images. Previous research in this field has primarily focused on developing machine learning and deep learning models, which achieved less than (90%). In contrast, the proposed model achieved a higher accuracy of more than (90%). Additionally, it is important to note that existing literature has mainly concentrated on developing classification models without emphasizing the creation of a user-friendly application.

2 Related Works

The field of image classification for fracture detection, various techniques and models have been explored in the literature. Support Vector Machine (SVM) is commonly used for classification, which has also been employed in some studies [1]. NLP techniques, such as text processing, have been used for feature detection in some papers. A survey of existing neural networks in the literature was conducted, and a need to identify and implement a (CNN) for image classification was identified [2]. Some studies have proposed fracture recognition using X-ray images by dividing them into small regions and extracting local binary histograms, which are combined into a single vector representing the image [3, 4]. Support Vector Machine (SVM) and Multi-layer Perceptron (MLP) have been used for fracture classification in some studies [5].

Several studies have utilized X-ray and craniofacial development methods for fracture analysis [6]. In the recent years, there are significant advancements in many fields through the use of Convolutional Neural Networks (CNNs) based on deep learning, which have the capability to find and extract features to enhance image classification accuracy [7, 8]. Intelligent system development can successfully learn and recognize objects which is a primary focus of research in pattern recognition and automatic classification domains. The objective of this work is to develop a system for estimating fractures and rods from X-ray images using CNNs [9, 10]. Three CNN models were used in this study with different architectures (e.g., number of filters, number of convolution layers) which were created and validated using the IMDB and WIKI datasets [11]. The results demonstrated that CNN networks significantly

improved the system's performance and recognition accuracy. The use of deep CNNs for learning representations has been shown to greatly enhance performance in image classification tasks [12]. The innovative CNN method employed in this study is designed to classify images without any restrictions and achieve accurate fracture classification [13, 14].

A CNN-based To find the area, an image segmentation algorithm has been presented of bone fractures using a developed GUI application. The output of image processing, shown in The Affected Area Localization, demonstrates that the proposed method accurately detects the bone structure and fracture edges, even in the presence of noise, surpassing other established edge detection methods such as Sobel and Prewitt and Canny [15, 16]. The proposed algorithm, Convolutional Neural Networks-based, clearly highlights the fractured area in an image. The SFCM clustering technique is utilized to approximate the fractured region and calculate the impacted area percentage, employing the DWT edge detection method within the algorithm [17, 18].

X-ray images of both healthy and fractured human bones, collecting a total of 100 original images from various sources, address the issue of overfitting in deep learning using less data. Techniques for data augmentation were used to increase the dataset, resulting in a final size of 400 images [19]. The classification accuracy of the model for distinguishing between healthy and fractured bones was found to be which is notably higher than achieved by other methods. Further improvements in accuracy can be achieved by exploring alternative deep learning models. However, validation on larger datasets is necessary to thoroughly evaluate the system's performance [20]. The studies under investigation have shown that transfer learning can be effective even when data are limited [21, 22]

3 Proposed Work

The architecture of purposed work is shown in Fig. 1. Data came from a bone fracture imaging repository, which contains a large database of publicly available datasets on various types of bone fractures. The collection of data was taken from Kaggle [23] which consist of 400 images of X-rays, and then, the data were divided into two parts training and testing in 70–30% ratio. Image sizes are of different sizes like (600 × 600) (700 × 700), all images are converted into 300 × 300 size, and Convolutional Neural Networks' model was used for feature extraction and classification (Fig. 2).

Convolutional Neural Network Algorithm steps

The model was developed using Convolutional Neural Networks' algorithm with five convolutional Hidden layers, each with different filter sizes and poolings.

Step 1: Load dataset.

Step 2: Divide the dataset into training and testing data and preprocessing is done 300 × 300 × 3 pixels.

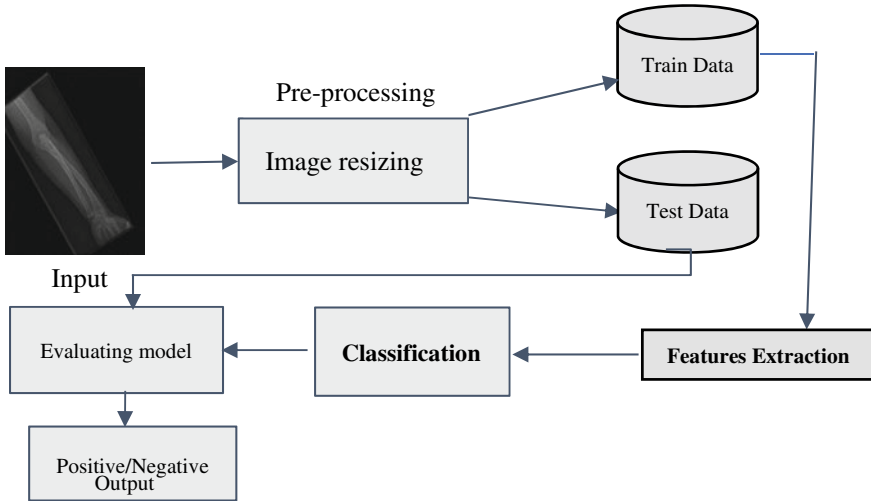


Fig. 1 Convolutional Neural Network architecture for bone fracture determination

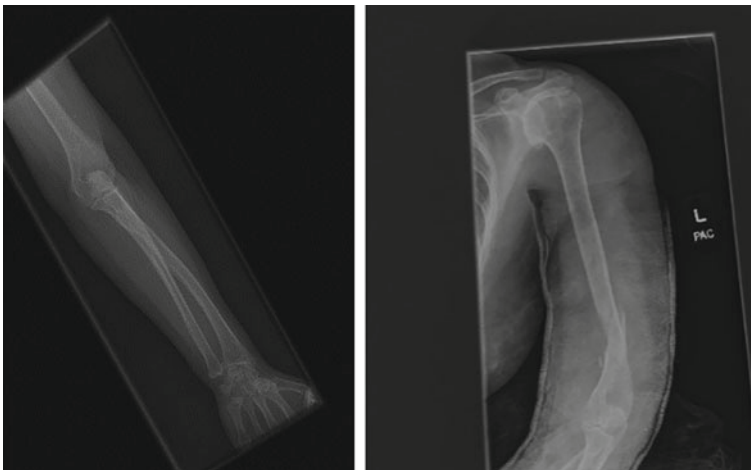


Fig. 2 Sample data

Step 3: There are five Hidden layers in this CNN model; the classification is done. The first Hidden layer uses 16 kernels of size 3×3 to determine fracture based on X-ray images Max pooling and ReLU activation function.

Step 4: The second Hidden layer uses 32 kernels of size 3×3 . Max pooling and ReLU activation function.

Step 5: The third Hidden layer uses 64 kernels of size 3×3 . Max pooling and ReLU activation function.

Table 1 Comparison of our results to the state-of-the-art in terms of fracture classification efficiency

Model	Fracture prediction accuracy %
SVM [20]	79.3
CNN [2]	88.18
Proposed approach	90.56

Step 6: The fourth Hidden layer uses 32 kernels of size 3×3 . Max pooling and ReLU activation function.

Step 7: The fifth Hidden layer uses 32 kernels of size 3×3 . Max pooling and ReLU activation function.

Step 8: After the output of seventh step, the matrix is converted into a vector.

Step 9: Then final output is displayed depending up on the input of the X-ray image either positive or negative.

4 Results and Discussion

The CNN model was run with initially with 400 epochs and obtained an accuracy of 70.45%, then with 500 epochs and got an accuracy of 80.69%. Finally, the model was run with 600 epochs and got an accuracy of 90.56%. The proposed model obtained better results compared to other similar models in literature. The results' comparison was shown in Table 1 and Fig. 5

After the model was built, a web application was developed where the users can upload an X-ray image and the application will give results as positive if fracture, negative if no fracture. The web application is shown in Fig. 3; the accuracy and model loss are shown in Fig. 4.

Limitations: The proposed model can only detect bone fractures in the parts of leg, wrist, elbow with more accuracy; for other parts, accuracy is slightly less. The model cannot detect whether there are multiple fractures.

Future scope: In future, model will be extended to detect fractures in other parts also by collecting more feature images of other parts. Accuracy of the model may be further improved by adding more Hidden layers and different combinations of convolution, pooling layers, and activation functions (Fig. 5).

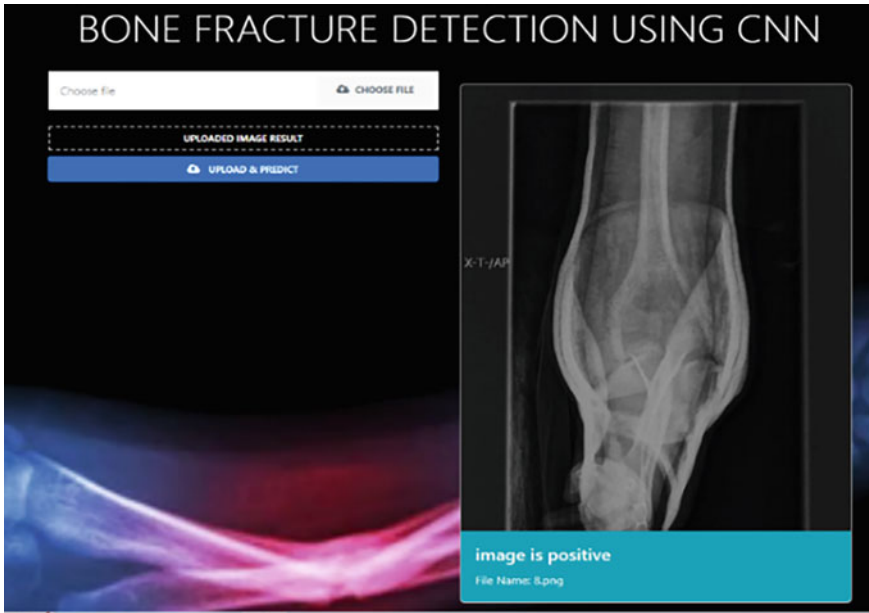


Fig. 3 Output page

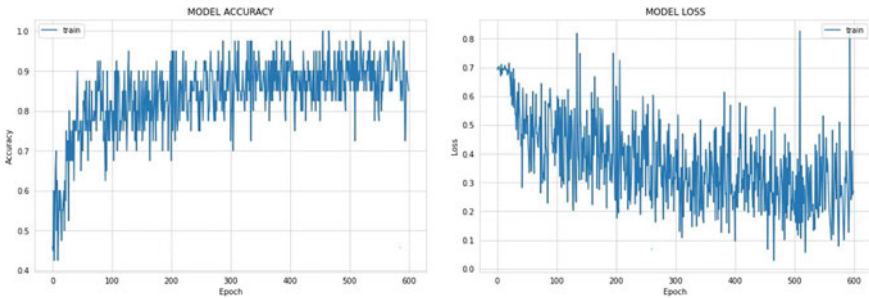
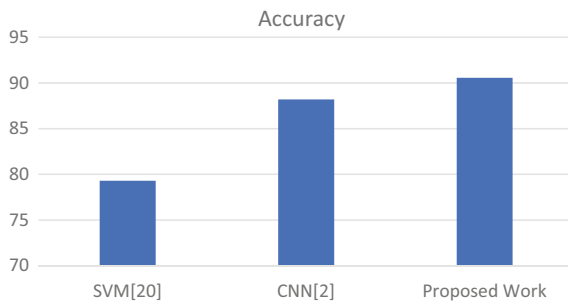


Fig. 4 Model accuracy and model loss

Fig. 5 Comparison of accuracy



5 Conclusion

In this paper, the use of Convolutional Neural Networks (CNNs) for bone fracture detection has shown great promise in improving the accuracy and efficiency of diagnosing fractures from medical images with 90.56% accuracy. While challenges and limitations exist, further research and development in this area hold great potential for advancing the field of radiology and improving the diagnosis and treatment of bone fractures.

References

1. Ma Y, Luo Y Bone fracture detection through the two-stage system of crack-sensitive, convolutional neural network. University of Science and Technology of China, Hefei, 230026, PR China
2. Wang X, Xu Z, Tong Y, Xia L, Jie B, Ding P, Bai H, Zhang Y, He Y (2022) Detection and classification of mandibular fracture on CT scan using deep convolutional neural network. *Clin Oral Investig* 26:4593–4601
3. Tanzi L, Vezzetti E, Moreno R, Moos S (2020) X-ray bone fracture classification using deep learning: a baseline for designing a reliable approach. Department of Management and Production Engineering, Politecnico di Torino, 10129 Torino, Italy; enrico.vezzetti@polito.it (E.V.); sandro.moos@polito.it (S.M.), 31 January 2020; Accepted: 20 February 2020; Published: 22 February 2020
4. Jacobs IS, Bean CP (1963) Fine particles, thin films and exchange anisotropy. In: Rado GT, Suhl H (eds) *Magnetism*, vol III. Academic, New York, pp 271–350
5. Shin HC, Roth HR, Gao M et al (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans Med Imaging* 35(5):1285–1298
6. Johari N, Singh N (2018) Bone fracture detection using edge detection technique. *Adv Intell Syst Comput* 584:11–19
7. Castro-Gutierrez E, Estacio-Cerquin L, Gallegos-Guillen J, Obando JD (2019) Detection of acetabulum fractures using X-ray imaging and processing methods focused on noisy images. In: *Proceedings—2019 Amity international conference on artificial intelligence (AICAI)*, pp 296–302
8. Upadhyay AM, Rajput AS, Singh AP, Kumar B (2017) Automatic detection of fracture in femur bones using image processing. In: *2017 international conference on innovations in information, embedded and communication systems (ICIIECS)*, pp 1–5. IEEE
9. Chai HY, Wee KL, Swee TT, Salleh SH, Ariff AK, Kamarulafizam (2011) Gray-level co-occurrence matrix bone fracture detection. *Am J Appl Sci* 8:26–32
10. Myint S, Khaing AS, Tun HM (2016) Detecting leg bone fracture in X-ray images. *Int J Sci Res* 5:140–144
11. Vegi VD, Patibandla SL, SKavikondala S, Basha Z (2016) Computerized fracture detection system using X-ray images. *Int J Control Theory Appl* 9:615–621
12. Kim DH, MacKinnon T (2018) Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. *Clin Radiol* 5:439–445
13. Dimililer K (2017) IBFDS: intelligent bone fracture detection system. *Elsevier Proc Comput Sci* 120:260–267
14. Yang AY, Cheng L (2019) Long-bone fracture detection using artificial neural networks based on contour features of X-ray images. *arXiv: 1902.07897v1*, 21

15. Korfiatis VC, Tassani S, George KM (2018) A new ensemble classification system for fracture zone prediction using imbalanced micro-CT bone morphometrical data. *IEEE J Biomed Health Inform* 22(4):1189–1196
16. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Proceedings of the 25th international conference on neural information processing systems*, Stateline, NV, USA, pp 1097–1105
17. Yadav DP, Sharma A, Singh M, Goyal A (2019) Feature extraction based machine learning for human burn diagnosis from burn images. *IEEE J Transl Eng Health Med* 7:1–7
18. Rathor S, Jadon RS (2019) The art of domain classification and recognition for text conversation using support vector classifier. *Int J Arts Technol* 11(3):309–324
19. Mahendran SK, Baboo SS (2011) An enhanced Tibia fracture detection tool using image processing and classification fusion techniques in X-ray images. *Global J Comput Sci Technol* 11:27–28
20. McBee MP (2018) Deep learning in radiology. *Acad Radiol* 25(11):1472–1480
21. Kim HE, Cosa-Linan A, Santhanam N, Jannesari M, Maros ME, Ganslandt T (2022) Transfer learning for medical image classification: a literature review. *BMC Med Imaging* 22:69
22. Sarvamangala DR, Kulkarni RV (2022) Convolutional neural networks in medical image understanding: a survey. *Evol Intell* 15:1–22
23. <https://www.kaggle.com/datasets/vuppalaadithyasairam/bone-fracture-detection-using-X-rays>

Smart Parking System Using YOLOv3 Deep Learning Model



Rishabh Tater, Preeti Nagrath, Jyoti Mishra,
Victor Hugo C. de Albuquerque, and José Wally M. Menezes

Abstract When it comes to vehicle surveillance, the Smart Parking System is considered indispensable, and it is now making its mark in the parking management sector. The errors caused by manual entry of vehicle registration information are absolutely eliminated. The process is made absolutely smooth and safe by the Smart Parking System. This not only stores reliable vehicle registration data in the database, but it also verifies the vehicle at exit points automatically. The Automatic Number Plate Recognition (ANPR) system is a crucial component of smart cities as it uses image processing and optical character recognition (OCR) technology to read vehicle number plates. ANPR enables traffic control and law enforcement through an automated, fast, reliable, and robust vehicle plate recognition system. This paper proposes an enhanced OCR-based plate detection approach that utilizes YOLOv3 deep learning model and an object-based dataset trained by convolutional neural network (CNN) to detect alphanumeric data from the identified license plate. The project will produce a Dataframe containing vehicle's registration details, entry time, exit time, and fees for the total duration of parking. To boost accuracy, a blended

R. Tater (✉) · P. Nagrath
Bharati Vidyapeeth's College of Engineering, New Delhi, India
e-mail: rishabh_tater14@gmail.com

P. Nagrath
e-mail: preeti.nagrath@bharatividyaapeeth.edu

J. Mishra
Department of Mathematics, Gyan Ganga Institute of Technology and Sciences, Jabalpur, M.P., India

Federal Institute of Education, Science and Technology of Ceara, Fortaleza, Brazil

J. Mishra
e-mail: [jyotimishra@ggits.org](mailto: jyotimishra@ggits.org)

V. H. C. de Albuquerque
Department of Teleinformatics Engineering (DETI), Fortaleza, Brazil
e-mail: victor.albuquerque@unifor.br

J. W. M. Menezes
Federal Institute of Ceará, Fortaleza, CE, Brazil
e-mail: wally@ifce.edu.br

algorithm for license plate detection and recognition is proposed and compared to current methodologies.

Keywords Automatic number plate recognition (ANPR) · Optical character recognition (OCR) · YOLOv3 · Convolutional neural networks (CNN) · Pytesseract

1 Introduction

We propose a solution to the inconvenience and security issues caused by incomplete or incorrect parking details using Automatic Number Plate Recognition (ANPR) with YOLOv3 deep learning model. ANPR is a vital aspect of Smart Parking Systems with various applications such as stolen car detection, parking management, and traffic flow monitoring. However, its performance suffers in complex scenarios.

While predicting parking location has been studied in detail, ANPR using deep learning is still an evolving field. The current quality of smart parking solutions needs improvement compared to other intelligent city services. Innovative and cost-effective solutions are necessary as parking and transportation are vital aspects of daily life.

The ANPR method involves license plate detection, segmentation, and optical character recognition (OCR). Choosing the appropriate model is crucial for object detection. Our proposed paper employs YOLOv3 and VGG16 models for license plate detection and compares their accuracies using a chosen dataset. YOLOv3 outperformed the other model, and we chose it as our prime detector. We were motivated by previous research proposed by Adarsh et al. [1].

ANPR faces challenges such as uneven lighting, weather, distortion of images, and image blurring, making it difficult to implement worldwide. Multinational ALPR systems have been proposed, but they involve a complex three-stage process. Our study proposes a cost-effective and accurate deep ALPR method that can be used on license plates from different geographical areas, filling the research gap in this field.

The paper is organized as follows: Sect. 2 provides a comprehensive review of various techniques employed for license plate identification along with their limitations. The proposed methodology is described in Sect. 3, followed by the presentation of experimental results and simulations in Sect. 4. Finally, the conclusion, findings, and future scope of work are discussed in Sects. 5 and 6.

2 Literature Review

Automatic Number Plate Recognition (ANPR) is a crucial technology in various applications such as law enforcement, traffic monitoring, and parking management. The three major steps involved in ANPR are license plate detection, segmentation,

and character recognition. Traditional image processing methods and deep learning approaches are the two main categories of ANPR techniques.

Deep learning models, particularly CNN-based approaches, have shown promising results in improving the accuracy and robustness of ANPR. These models have been used for license plate detection, enabling the estimation of the location of the license plate. YOLO, a fast and efficient object detector that predicts boundary boxes and class probabilities directly from the whole image, has been used in various ANPR systems for detection and character recognition. However, YOLO has limitations in generalizing objects with new or unusual configurations.

To overcome the limitations of individual techniques, hybrid methods combining both traditional image processing and deep learning approaches have been proposed [2–4]. These methods aim to leverage the strengths of both techniques to achieve better accuracy and robustness. For instance, some hybrid methods have utilized a CNN-based license plate detector in combination with traditional image processing techniques for character segmentation and recognition, achieving high accuracy in detecting and recognizing license plates under various lighting and weather conditions.

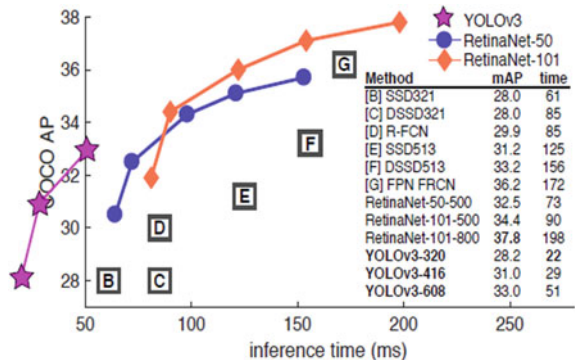
In summary, ANPR is a crucial technology that involves license plate detection, segmentation, and character recognition. Deep learning models, particularly CNN-based approaches, have improved the performance of ANPR. Hybrid methods combining traditional image processing and deep learning approaches have shown promising results and are a potential direction for future research in ANPR.

In [1], a new approach for object detection is proposed, which detects and recognizes new objects in a single stage while achieving high precision without sacrificing speed. The authors categorized existing object detection methods into two classes: two-stage detectors and single-stage detectors, and identified the need for a more efficient single-stage model. They introduced YOLOv3-Tiny as an improvement upon the existing YOLOv1, YOLOv2, YOLOv3, and SSD architectures, which increases object detection speed while maintaining precise results. YOLO was used as an object detector in [5–8].

The YOLOv3 paper [9] also proposes updates to the YOLO object detection system as shown in Fig. 1, achieving similar accuracy as SSD but at three times the speed. The proposed multinational license plate recognition system using YOLOv3 [10] incorporates a Space Pyramid Pooling block in the second step for more efficient character identification in ALPR. After the localized license plate is input into the network, the character recognition network returns the bounding boxes of the predicted characters, and a layout algorithm is used to extract the correct license plate number sequence. The authors emphasize the importance of obtaining the correct sequence for multinational license plates. The application of ANPR as finding occupancy in parking area slots is proposed in [11] that used inception and MobileNet.

Template matching and neural networks are two different approaches to license plate detection and recognition. Template matching is a method that matches a pre-defined template of a license plate to the input image, and then identifies each character of the plate using optical character recognition (OCR) [12–14]. On the

Fig. 1 YOLOv3: an Incremental Improvement [9]



other hand, neural networks use machine learning algorithms to detect and recognize license plates. Several types of neural networks have been used for license plate recognition, including Artificial Neural Network (ANN), BP Neural Network, and Probabilistic Neural Networks [15–17]. Detection of license plate enables an estimation of the site of the license plate using CNN based approach [18, 19]. This model a function that provides a score for each image sub regions and enables us, by combining the results obtained by sparsely overlapping regions, to estimate where the detected license plates are locations. In [20], author has proposed, a single neural network that predicts boundary boxes and class probabilities in one evaluation directly from the whole image.

Symmetric wavelets and multi-level genetic algorithms are two other techniques that have been used for license plate detection [14, 21]. In symmetric wavelets, statistical measurements such as RMSE and PSNR are used to locate license plate data after preprocessing the input image [14]. Multi-level genetic algorithms, on the other hand, can identify and locate multiple license plates in a single image at excellent precision rates [21].

Overall, the use of neural networks for license plate recognition has become increasingly popular in recent years. However, other techniques such as template matching and genetic algorithms continue to be used in the field, each with its own strengths and weaknesses.

YOLOv3 and SSD are both popular single-stage object detection algorithms that have been used in various computer vision applications. YOLOv3’s advantages include multi-label classification, logistic regression for target score computation, and the use of three scales for small to large-scale detections. SSD, on the other hand, uses anchor boxes with different aspects to learn offset rather than the box and has pre-defined anchor boxes for each location on the function map. YOLOv3-Tiny is a lightweight variant of YOLOv3 that requires less operating time but has lower precision. Overall, these algorithms offer different trade-offs in terms of detection accuracy and computational efficiency and can be used in various real-world applications depending on the specific requirements.

3 Methodology of Proposed Work

This section aims to provide specific guidance on how the methodology of the proposed work should be implemented as shown in Fig. 2. The method proposed consisted of number plate detection, identification of the characteristics of the number plate detected and data storage on the database system. We used YOLOv3 CNN for the detection of the license plate, using the images as input and returning the license plate location annotations. The recognized area of the vehicle license plate is then pre-processed and applied to OCR, which successfully acknowledges the number plate character, and then saves data in the data frame for the same number plate.

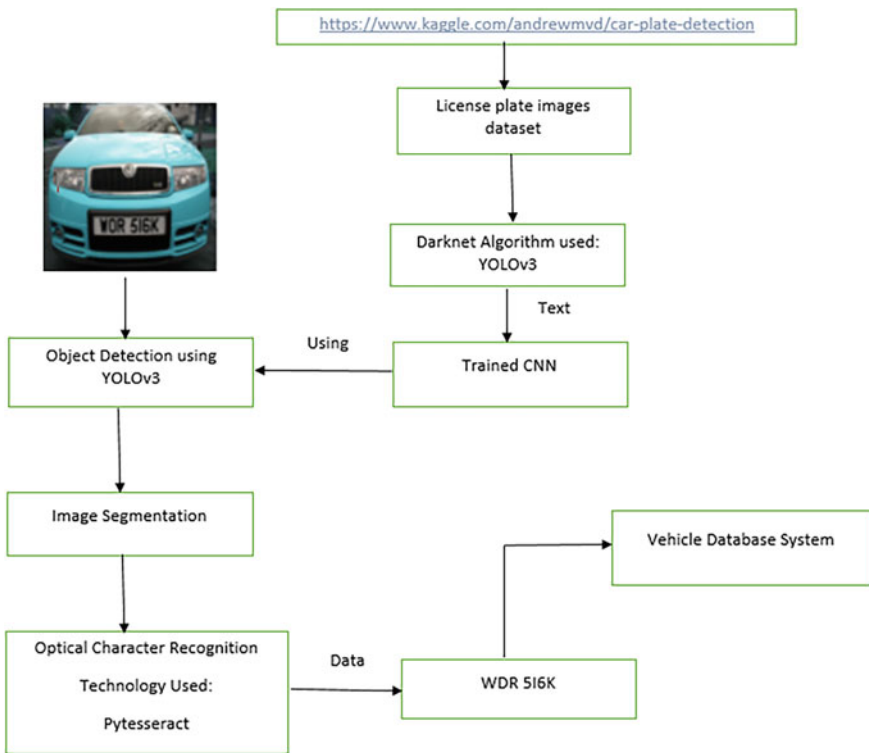


Fig. 2 Proposed methodology flow chart



Fig. 3 Sample images from the dataset

3.1 Dataset

First, a dataset was chosen containing 433 images. This dataset contains bounding box annotations of car license plates within the image. Annotations are provided in the PASCAL VOC format.

Name of the dataset is Car License Plate Detection [22]. The chosen dataset contains 433 images and 433 annotations. With the same name in the same directory the xml files of every images are also present, which contains the coordinates of the images as $\langle x_center \rangle \langle y_center \rangle \langle width \rangle \langle height \rangle$. Sample images from dataset are shown in Fig. 3.

3.2 Training the Model Using Darknet Framework

For number plate detection, the device was trained, and the program was written in Python. We used Darknet framework, an open source neural network framework, for training the detector. In the proposed work the detector is YOLOv3 deep learning model.

You only look once (YOLO) uses an image as an input, runs it via a neural network and predicts the bounding boxes. The prediction for each bounding box includes five components, i.e., x , y , w , h , and confidence. (x, y) represents the center of the bounding box, while (w, h) the width and the height of the boxes, and confidence is the estimated accuracy of the object prediction. Training is done just by placing the YOLOv3 model as input and annotations as output on image data, i.e., x , y , w , h , and confidence.

YOLOv3 is a model which, thanks to its rapidity and precision, has a variety of applications. In this paper, our model is based on YOLOv3. The more convolutional layers we use, the better the outcome. Based on the foregoing, the model provided in this work has a more complex structure that is not only more suitable for our database, but also allows us to recognize targets at a finer level. The Darknet is used to extract characteristics in the original YOLOv3.

3.3 CNN Architecture for Detecting License Plates Using YOLOv3

In India, various ANPR techniques are used, but their effectiveness is very low. The proposed framework aims to increase and optimize ANPR performance. YOLOv3 was used to train the machine at first for number plate detection, using convolutional neural network (CNN), which is capable of detecting objects and entities. CNN applies filter on the input which generates the feature map. Then in the input, the presence of detected features is summarized by the generated feature map.

Within a given image, this model is in charge of locating and identifying a license plate. To train this network, we employ a collection of real-world license plate photos as well as license plate annotations. The network is meant to adapt to different scenarios and be adaptable to regional changes in license plates because the training data we utilized covers a wide range of variations.

Of the last three convolutionary layers shown in Fig. 4, logical should generate the 104×104 , 52×52 , 26×26 feature map. Since high-level characteristics have more semantic information, we still want to collect finer-grained level characteristics. The low level features have more spatial details, on the other hand. The updated model adds a new convolution layer behind the 26×26 function plan to achieve a smaller feature map with a stride of 2. The 13×13 feature is then linked to the 26×26 feature map, to perform the maximum pooling process. The results detect a high-level target. The results will be updated and linked with the 26×26 feature map and 52×52 feature map, to be used following a concatenation method for max-pooling. The findings are a medium-level goal. The intermediate-level function map samples and fits in with the 52×52 function map.

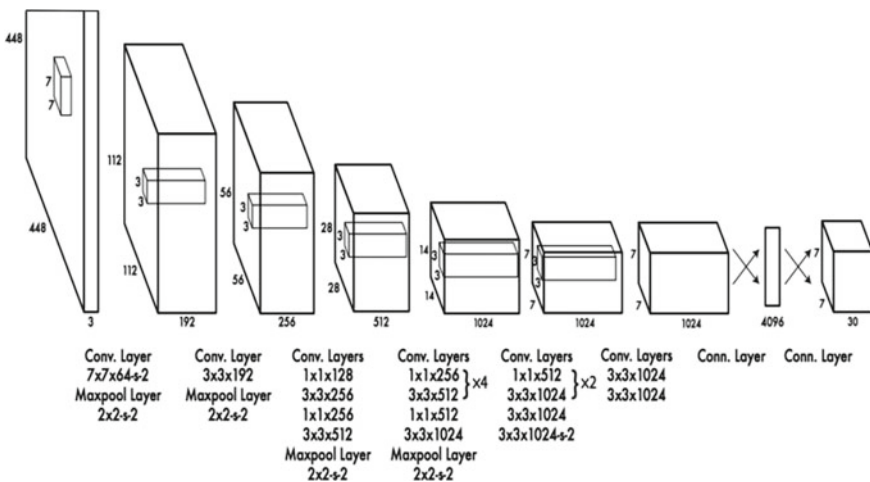


Fig. 4 Convolutional neural network architecture [10]

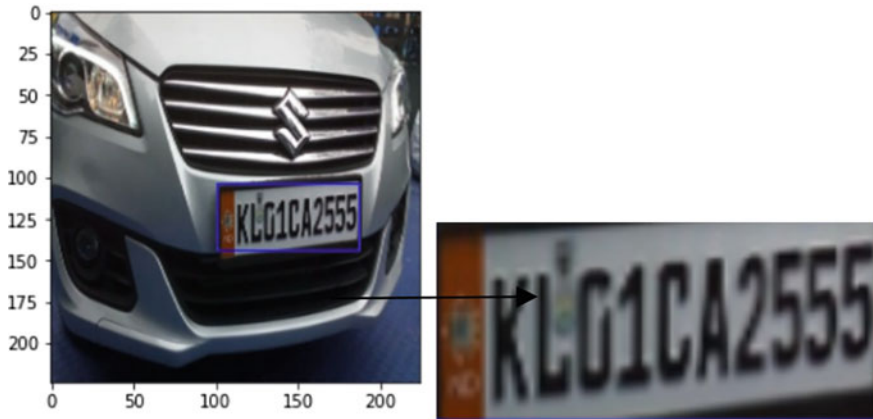


Fig. 5 Segmented number plate from the original image

3.4 Image Segmentation

The next step is to segment the number plate out of the image after successfully detecting the number plate. It is also possible to do this using OpenCV library, by cropping the number plate region and then saving it as the new image. Segmentation serves as a link between character recognition and number plate extraction as shown in Fig. 5. Boundary box analysis is another name for segmentation and the characters are extracted by using this analysis.

3.5 Optical Character Recognition Using Pytesseract

Conversion of manually printed or printed text images into computer text is known as the optical character recognition (OCR). There are a number of OCR engines available, the proposed work uses Python-Tesseract also called Pytesseract. Python-Tesseract is a python-based optical character recognition (OCR) application. It can recognize and interpret text embedded in pictures. This will be our most effective method for recognizing license plates.

Tesseract contains a new neural network component that can recognize text lines. It is based on OCRopus' Python-based LSTM implementation; however it has been rewritten in C++ for Tesseract. Tesseract's neural network system predates TensorFlow, but it is compatible with it because it uses the Variable Graph Specification Language as a network description language (VGSL).

Pytesseract OCR accepts the segmented image as input, and then the characters in the image of the number plate will then be recognized. The collected data is saved in a database or a data file as shown in Fig. 6.

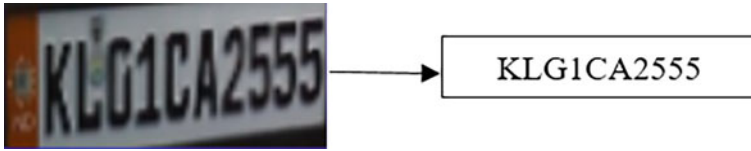


Fig. 6 OCR using Pytesseract

	Licence_Number	Entry time	Exit time	Difference in minutes	Amount
0	/KAO3-MG- 2784	2021-06-16 18:40:42.866059	2021-06-16 18:44:43.005556	4.000000	30
1	DL7C N 5617	2021-06-16 18:40:43.124571	2021-06-16 19:44:18.931102	63.583333	80
2	SRV P	2021-06-16 18:40:43.322041	0	0.000000	0
3	COUInEAGLE W	2021-06-16 18:40:43.523503	0	0.000000	0
4	JAG2 UAR	2021-06-16 18:40:43.735934	2021-06-16 21:16:23.689468	155.650000	100

Fig. 7 Data entry in database system

3.6 Storing Extracted Data in Database System

The characters extracted from the OCR of license plates will be stored with the date, entry and exit time, and vehicle no. in the database system as in Fig. 7 using the Pandas data frame. Also, generating unique User ID of each vehicle entering the parking. With the help of the vehicle’s time of entry and its time of exit in the parking lot will calculate the parking duration and ultimately generating the fees according to the total time consumed by the vehicle in the parking lot.

4 Experimental Results

We conducted our experiment on several features of vehicles with completely various shapes and dimensions all of them subject to different conditions in order to assess their process and precision. The algorithm’s accuracy was limited because for plates at a certain degree and plates at the edge of the image, the segmentation approach did not produce the anticipated results. It needs a proper camera angle setup to be more efficient and effective.

Evaluation Criteria. Formula for the evaluation of accuracy:

$$\begin{aligned}
 \text{Accuracy} &= \frac{\text{number of correct predictions}}{\text{total number of prediction}} \\
 &= \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{1}
 \end{aligned}$$

where TP stands for True Positives, TN stands for True Negatives, FP stands for False Positives, and FN stands for False Negatives.

Formula for Mean squared error,

$$MSE = \sum_{i=1}^n \frac{(W^t x(i) - y(i))^2}{n} \tag{2}$$

The proposed method yielded the following outcomes:

The above graph in Fig. 8 shows the evaluation scores when the dataset was trained using YOLOv3 detector. Score apprentissage depicts the training data score whereas score validation depicts validation score. (Works as a part of test data from the same training dataset.) It yielded an accuracy score on training data and on validation to be 94.2% and 80%, respectively, for 50 epoch.

The graph in Fig. 9 shows the evaluation scores when the dataset was trained using VGG16 detector. It yielded an accuracy score to be 79.13% on 200 epoch.

Fig. 8 Evaluation scores of smart parking system using YOLOv3

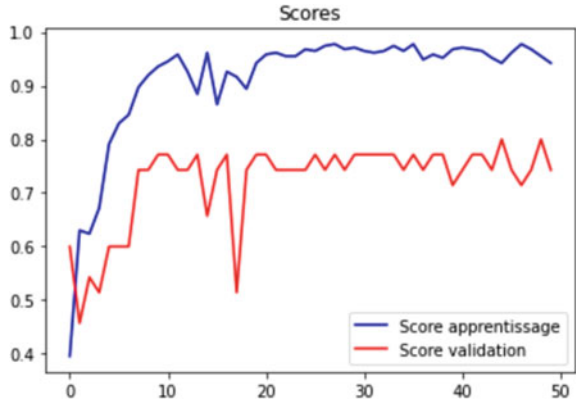
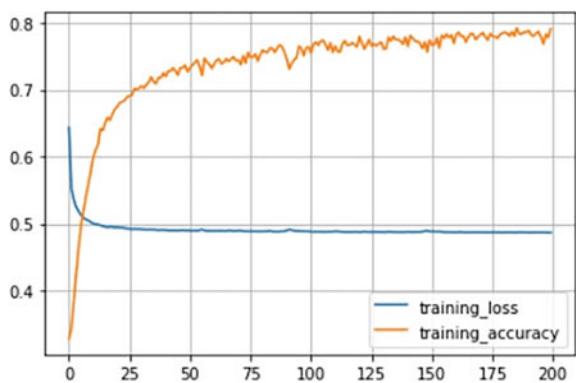


Fig. 9 Loss function of smart parking system using VGG16



5 Conclusion

The proposed algorithm for license plate detection is simple and may successfully categorize various license plate layouts. It can bring a number of benefits, such as traffic safety adherence, safety in the event of susceptibility, ease of use and immediate access to information—compared to the phase of segmentation searching for registration details of vehicle ownership. The lighting, the terminology, the car shade and the non-uniform plate size, the character on the plate, distinct font and the background color are factors that affect ANPR results.

Our system was trained using the YOLOv3-Darknet framework. The model for license plate detection was trained using YOLOv3 with CNN which is capable of detecting object and entities. Then OCR was applied for number plate recognition using Tesseract API available in python called Pytesseract. The results of our method yielded an Accuracy score on training data and on validation to be 94.2% and 80%, respectively. It is clear that due to the complicated ANPR system, it is currently impossible to achieve a 100% overall accuracy since each stage is dependent on the previous step. However, if bounding boxes are accurate, our algorithm is able to extract the correct license plate numbers from an image.

6 Future Scope

In future research, we'll look into employing an applied noise reduction technique to improve license plate recognition accuracy without dramatically increasing calculation time. The disadvantage of using a single class classifier in an ensemble model is that it will significantly increase computation time. We are investigating two options to fix this problem. A proposal-based technology like Fast R-CNN can be utilized to minimize the calculation time of the underlying classifier. Secondly, we can use parallel calculation to simultaneously calculate the basic classifier.

Algorithms such as super resolution of images can be applied for low-resolution images. A coarse to-fine technique may be useful for segmenting multiple vehicle number plates. Since OCR has become a commonly used and common tool in recent years, instead of redesigning the entire ANPR, ANPR developers are focusing on increasing OCR accuracy. Some developers are modifying open sources, like Tesseract, in an attempt to improve their accuracy, as mentioned in the previous section.

References

1. Adarsh P, Rathi P, Kumar M (2020) YOLO v3-tiny: object detection and recognition using one stage improved model. In: 2020 6th international conference on advanced computing and communication systems (ICACCS), Coimbatore, India, pp 687–694
2. Pustokhina IV et al (2020) Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems. *IEEE Access* 8:92907–92917
3. Siddiqui SY et al (2020) Smart occupancy detection for road traffic parking using deep extreme learning machine. *J King Saud Univ Comput Inform Sci*
4. Yépez J, Castro-Zunti RD, Ko S-B (2019) Deep learning-based embedded license plate localisation system. *IET Intell Transp Syst* 13(10):1569–1578
5. Chen R-C (2019) Automatic license plate recognition via sliding-window darknet-YOLO deep learning. *Image Vis Comput* 87:47–56
6. Naren Babu R, Sowmya V, Soman KP (2019) Indian car number plate recognition using deep learning. In: 2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICT), Kannur, Kerala, pp 1269–1272
7. Laroca R, Severo E, Zanlorensi LA, Oliveira LS, Goncalves GR, Schwartz WR et al (2018) A robust real-time automatic license plate recognition based on the YOLO detector. In: *Proceedings of the international joint conference on neural networks (IJCNN)*, pp 1–10
8. Bura H, Lin N, Kumar N, Malekar S, Nagaraj S, Liu K (2018) An edge based smart parking solution using camera networks and deep learning. In: 2018 IEEE international conference on cognitive computing (ICCC), pp 17–24
9. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement
10. Henry C, Ahn SY, Lee S-W (2020) Multinational license plate recognition using generalized character sequence detection. *IEEE Access* 8:35185–35199
11. Karakaya M, Akıncı FC (2018) Parking space occupancy detection using deep learning methods. In: 2018 26th signal processing and communications applications conference (SIU), pp 1–4
12. Ghazali MNB, Rusli MA (2018) Development of car plate number recognition using image processing and database system for domestic car park application
13. Kashyap A et al (2018) Automatic number plate recognition. In: 2018 international conference on advances in computing, communication control and networking (ICACCCN). *IEEE*
14. Himani V et al (2014) Automatic vehicle number plate localization using symmetric wavelets. In: *ICT and critical infrastructure: proceedings of the 48th annual convention of computer society of India*, vol 248 of the series advances in intelligent systems and computing, pp 69–76
15. Zhai X, Bensaali F, Sotudeh R (2012) OCR-based neural network for ANPR. In: *IEEE*, p 1
16. Zhang Z, Wang C (2012) The research of vehicle plate recognition technical based on BP neural network. *AASRI Proc* 1:74–81
17. Öztürk F, Özen F (2012) A new license plate recognition system based on probabilistic neural networks. *Proc Technol* 1:124–128
18. Zain Masood S, Shu G, Dehghan A, Ortiz EG (2017) License plate detection and recognition using deeply learned convolutional neural networks. *arXiv:1703.07330*
19. Delmar Kurpiel F, Minetto R, Nassu BT (2017) Convolutional neural networks for license plate detection in images. In: *Proceedings under IEEE international conference on image processing (ICIP)*, pp 3395–3399
20. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified real-time object detection. In: *Proceedings of CVPR*, pp 779–788
21. Anantha Reddy C, Shoba Bindu C (2015) Multi-level genetic algorithm for recognizing multiple license computer science and engineering
22. Dataset <https://www.kaggle.com/andrewmvd/car-plate-detection>

Crop Prediction Using Machine Learning with CRISP-DM Approach



Lendy Rahmadi, Hadiyanto, Ridwan Sanjaya, and Arif Prambayun

Abstract Machine learning is a successful dynamic tool for forecasting crop yields, as well as for choosing which crops to plant and what to do during the growing season. One of the industrial process models used in data mining is Cross-Industry Standard Process for Data Mining (CRISP-DM), which includes six iterative phases. Crop prediction using various machine learning algorithms proposed by the authors in this study using CRISP-DM as an approach. Machine learning algorithms are used to perform classifications to predict crops. The machine learning algorithms that are used to carry out a classification to be able to predict crops include Random Forest, Naive Bayes, K-Nearest Neighbors, Decision Tree, and eXtreme Gradient Boost (XGBoost). This article contributes to presenting and providing an understanding of crop prediction using machine learning. From the classification and modeling results using crop recommendation data set, the Decision Tree algorithm becomes an algorithm that has the lowest accuracy results compared to other algorithms with value of 0.9 for testing and value of 0.88 for training. Meanwhile, the algorithm that has the highest level of accuracy of the machine learning algorithm used is the XGBoost algorithm with test value accuracy of 0.993 and training value accuracy of 1.0.

Keywords Machine learning · CRISP-DM · Crop prediction · Classification

L. Rahmadi (✉) · Hadiyanto
Diponegoro University, Semarang, Indonesia
e-mail: lendy@lembahdempo.ac.id

Hadiyanto
e-mail: hadiyanto.chm@undip.ac.id

R. Sanjaya
Soegijapranata Catholic University, Semarang, Indonesia
e-mail: ridwan@unika.ac.id

A. Prambayun
Sriwijaya State Polytechnic, Palembang, Indonesia
e-mail: prambayun@polsri.ac.id

1 Introduction

Agriculture has an important role for the development and development of the country's economy. To increase the effectiveness and efficiency of various activities in agriculture, machine learning is applied to different agricultural needs. Various advanced technologies are needed in agriculture such as advanced technology and automation using IoT required in data collection and mining. Machine learning is needed in conducting data analysis, looking for patterns to predicting weather forecasts, making predictions on recommendations for plants to be planted, determining fertilizers and activities to be carried out, to identifying diseases in plants.

In order to increase effectiveness, productivity, the use of advanced agricultural technology is imperative. One of the advanced agricultural technologies is the use of machine learning. Machine learning is a branch of computer science that is part of the field of artificial intelligence, machine learning studies algorithms and data and then uses approaches to solve complex problems which are generally quite difficult to program using traditional methods [1]. Various machine learning techniques have been used in various fields, a simple example is studying consumer behavior in supermarkets. In agriculture, machine learning has been used for many years [2].

Big data technologies and high-performance computing have given rise to machine learning, bringing new perspectives to the interdisciplinary field of agricultural technology [3]. Machine learning is part of the field of artificial intelligence (AI) which focuses on learning through data [4, 5]. One of the dynamic techniques that can be used effectively to decide which crops to plant, predict yields to determine what activities to do during the growing season is machine learning [4]. In machine learning, data can be obtained by applying the required sensors applied [6] then processed and analyzed using machine learning algorithms in order to better monitor and optimize agricultural practices. In addition, by utilizing data from sensors, satellite imagery and climate recordings, machine learning algorithms can be used to predict weather and rainfall [7].

Machine learning is a recent technique that helps farmers avoid crop losses by offering extensive advice and insights into the crops. Various problems in the pre-harvest process such as determining the plants to be planted, the harvest process in monitoring and controlling to the post-harvest process can be overcome by using machine learning. In the field of agriculture machine learning can make agriculture much more effective and efficient, can produce high quality products to reduce the use of human labor [8].

Currently there are several different machine learning frameworks, but there is no single framework that can be used and is suitable for all machine learning purposes. In fact, some frameworks are used more often than other frameworks [9]. Therefore, we wanted to try to find out which machine learning algorithms can be used to analyze and classify crop predictions with better results and accuracy.

Various problems in agriculture today are often sought for solutions to the problem by using data mining. Starting from finding the patterns needed in data sets, conducting analysis in large data sets to classifying them according to needs can be done with data mining. Data mining techniques have the basic objective of extracting information from a data set and turning it into a structure that can be used as needed [10].

Machine learning and data mining often use the same methodology, and sometimes the differences between the two can be confusing [9]. In machine learning the learning process is carried out from the data used, while data mining is the process of obtaining data to carry out learning. The explanation of these two concepts is interconnected and related, it can be analogized that data mining is a task and machine learning is an instrument to achieve this task. In data mining there are two main objectives, namely prediction and description. This goal is achieved by using data mining techniques, one of which is machine learning [11].

One of the industrial process models used in data mining is Cross-Industry Standard Process for Data Mining (CRISP-DM), which includes six iterative phases [12]. CRISP-DM consists of six sequentially different phases and provides a common process model that covers the overall structure and methodological dimensions [13]. CRISP-DM is a standard process model used for data mining projects. Various systematic literature reviews provide summaries and descriptions of how CRISP-DM is used in various current studies to find innovative methods, best practices to research priorities [14].

This article contributes to presenting and providing an understanding of crop prediction using machine learning with the CRISP-DM approach. In this article, it is explained that the stages carried out according to the phases in CRISP-DM are applied to the stages in carrying out classifications to make crop predictions using machine learning. In this study using machine learning algorithms the author will make crop predictions with the CRISP-DM approach. There are several machine learning algorithms used in this study, namely Random Forest, Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, and XGBoost.

2 Theoretical Background

2.1 Machine Learning

Machine learning is something we encounter and use every day. Like when Instagram gives recommendations on which accounts we should follow, when Google gives recommendations for the keywords we use, and Netflix gives recommendations on what movies to watch. This is an example of implementing machine learning in our daily lives. Machine learning (ML) is a multidisciplinary field that focuses on developing computer algorithms capable of extracting predictive information from static or dynamic data sources by employing analytic or probabilistic models and

refining them through training and feedback [15]. Machine learning is a part of computer science that can enable computers to learn through data without being programmed explicitly and specifically [15].

Learning from a training data set and then making the most accurate predictions on test data or unknown data is the main goal in machine learning [16]. When analyzing and interpreting various data sets, machine learning algorithms are used to predict outcomes using test data or new, unexplored data. If the accuracy results are not as expected, the algorithm will be exposed to larger data sets. This allows algorithms to train and learn from experience with large amounts of data. This process continues until the prediction results reach the desired accuracy, after which the algorithm is developed.

Machine learning algorithms can be classified into different classes depending on the type of model, data type and learning process [17]. According to Ayodele, machine learning algorithms are classified into taxonomies based on the expected results of the algorithms used. Examples of common types of algorithms include supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transformation, and learning to learn are examples of commonly used types of algorithms [18]. Machine learning algorithms are used to acquire knowledge throughout this process.

2.2 Data Mining

Data mining is a process carried out to find interesting structures in data. These structures can be found in various forms, such as graphs or networks, trees, one or more equations, a set of rules, and so on [19]. In modern data mining, traditional data analytics technologies are combined with statistics, tools, methods, ideas from computer science, machine learning to database technology [20].

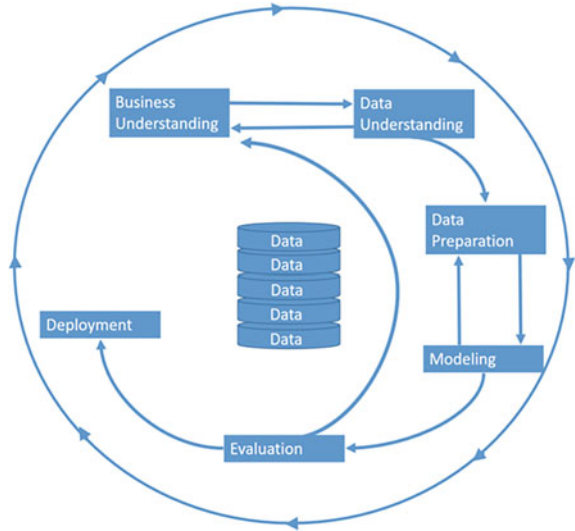
CRISP-DM describes the phases of a project, the work involved in each phase, and the relationships between those tasks, providing an overview of the data mining.

There are numerous models of data analysis approaches that data practitioners might use in data mining. One of them is the CRISP-DM model. As a methodology, CRISP-DM describes the phases of the stages in a project, the work involved in each phase and the explanation regarding the relationship between these jobs and provides an overview of the life-cycle of data mining when viewed as a process model.

Various studies have been conducted related to the application of data mining techniques to agricultural data sets. Through examination of experimental data sets from large soil profiles, to create soil categories applied the Naive Bayes Data Mining Technique [21].

To predict soil fertility, the Decision Tree algorithm in data mining is used [22]. In order to be able to predict the results of the k-means approach analysis was used by Ramesh [23]. Vamanan and Ramar [24] conducted a literature review of several data mining methodologies used in the agricultural domain.

Fig. 1 Six phase in CRISP-DM



2.3 CRISP-DM

CRISP-DM is an industry independent process model for data mining. There are six iterative steps in CRISP-DM (see Table 3). Table 3 provides a brief explanation and description related to tasks, activities, main ideas to the results of this phase, using the CRISP-DM User Guide as a reference [25].

Through Fig. 1 we can see the six different phases showing common examples of the various steps being taken.

When data mining in a particular domain is carried out, there will always be various considerations related to the specificity of that domain. For the field of agricultural machinery, important aspects in machine optimization are indicated by three main characteristics, namely: machine variability, quality index, and environmental conditions. Agricultural machinery systems have a direct impact on profitability and must be configured optimally. Thus, an important step that must be taken is to find the right decision variables, because these decision criteria can determine how the system should behave. Decision variables are needed data-based models in order to drive the desired model behavior. In the first four phases, implicitly these variables are considered CRISP-DM [26].

3 Methodology

In this article, the research methodology begins with preparing the data set used, namely the crop recommendation data set obtained from kaggle.com which is open access. Then the five phases of CRISP-DM are used as guidelines and references for

the stages in making models for crop prediction using machine learning algorithms. The deployment phase is not used in this article because the stages carried out are only up to evaluation, not to deployment. After mapping out the phases in the CRISP-DM that served as a guide, the prepared data set was processed using Python and five machine learning algorithms including Random Forest, KNN, Decision Tree, Naïve Bayes, and XGBoost. Data processing is carried out from the stage of understanding the data to the evaluation stage.

In order for machine learning algorithms to be used optimally and data processing to run properly, various specific libraries are needed to be imported. By using this library the process of understanding data through various visualizations, data preparation to modeling algorithms to make predictions can run efficiently. Various libraries used include pandas for data analysis, numpy for linear algebra, various libraries for visualization such as matplotlib.pyplot, seaborn, plotly.graph_objects, plotly.express, plotly.subplots.

The machine learning algorithm performs an analysis of the data according to the input variables to find out what the possible output patterns are. In this article, a supervised machine learning approach is used to predict plants according to the specified input variables so that the possible output poles can be identified. The predictions made on the machine learning model will be based on a model that is trained using a set of training data, in this case data set crops recommendation which will then be applied to a set of test data in order to produce actual and predicted conditions.

3.1 Business Understanding

In the crop recommendation data set obtained by open access from kaggle.com, there are 8 columns consisting of 7 predictor variables and 1 dependent variable as labels. Users can provide parameters such as N (Nitrogen), P (Phosphorus), K (Potassium), temperature, humidity, pH, rainfall, and label crop names. Then the algorithm will predict which plants are suitable and can be planted according to the input parameter values. Crop recommendation data set can be seen in Fig. 2.

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Fig. 2 Data set crop recommendation

The data set above consists of 8 columns with 7 predictor variables that determine what plants are recommended according to the parameters entered and 1 dependent variable. Nitrogen content ratio in soil (0–140), Phosphorus content ratio in soil (0–150), and Potassium content ratio in soil (0–210). N–P–K is content in the soil that is useful for the growth of leaves, roots, flowers, fruit, and the whole plant. Temperature (temperature in degrees Celsius), humidity (relative humidity value in %), pH (pH value in soil), and rainfall (rainfall in mm) are independent variables that also affect plant conditions.

3.2 Data Understanding

To make the best predictions for the processed data set, it is very important to carry out detailed data analysis at the data understanding phase. Analysis needs to be done to understand the data further both in terms of data form, and data description taking various information that can be obtained through data sets. The analysis carried out in the data understanding phase is Exploratory Data Analysis which aims to be able to understand more deeply related to the data set used. Through Exploratory Data Analysis, data descriptions, data forms, and various other information are presented through graphs, descriptions, plots, and maps so that the data can be better understood.

In Fig. 3 the data description, we can see and understand the values in the data set. Starting from the average value, standard value, minimum and maximum value to the percentage value of each variable in the data set.

The crop recommendation data set consists of 2200 records from 22 varieties of vegetables and fruit. Each vegetable and fruit label consisting of 22 labels has 100 records. So if we add them up we can understand that the form of the data from the crop recommendation data set consists of a total of 2200 records or 2200 rows and there are 8 columns, 7 containing parameters or variables and 1 label. Figure 4 presents a count plot showing the plant labels and the number of records.

	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

Fig. 3 Data description

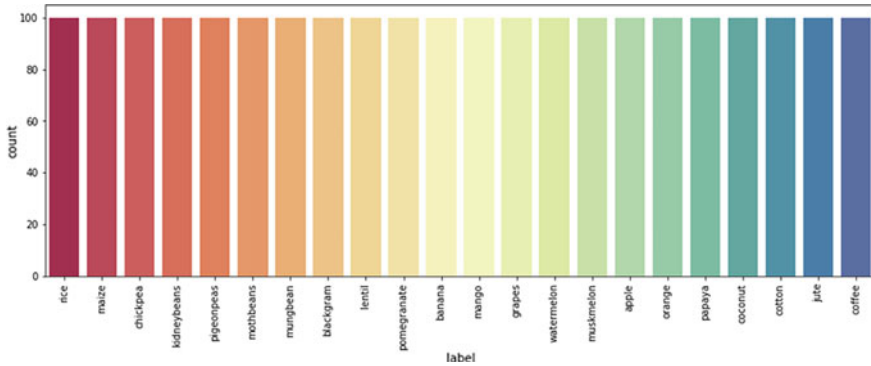


Fig. 4 Count plot label and records

In data understanding, apart from understanding the data, Exploratory Data Analysis (EDA) is also carried out to analyze related existing data sets so that they can be understood further before this data set is continued to the data preparation phase. In Exploratory Data Analysis, various information is presented through graphs, plots, and subplots which are visualizations of each variable in the data set.

Through Fig. 5, we can understand and know the required pH value of each vegetable and plant label in the data set. On average, all plant labels in the data set can grow in a pH value ratio of 5–8. Mothbeans have the widest pH ratio compared to other plants with a pH value between 5.4 and 8.4. This explains that Mothbean plants have the opportunity to grow in a variety of environments because they have a wide ratio of pH values. There are two plants that have the narrowest pH value ratio, namely Kidneybeans with a pH value ratio of 5.7–5.9 and Papaya with a pH value ratio of 6.7–6.9.

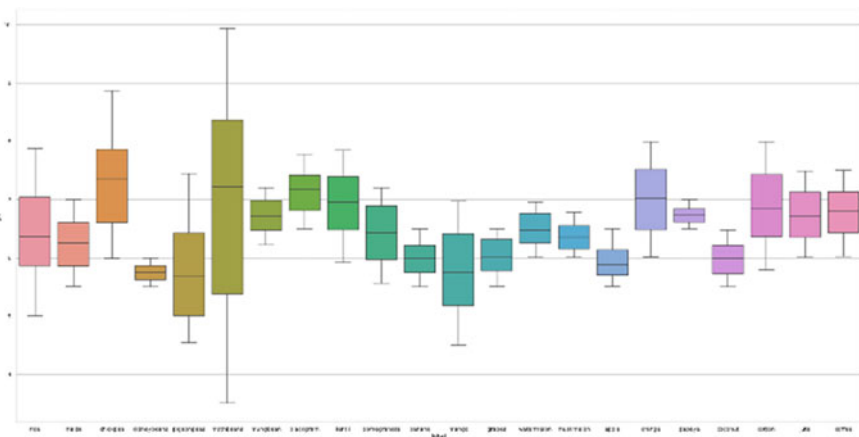


Fig. 5 pH ratio for each label

3.3 Data Preparation

After we have understood the data set used, then we must prepare the data set so that it can be processed properly. Data preparation involves collecting raw data and preparing it for inclusion in an analytics platform. To move to the final stages of preparation, the data must be cleaned, formatted, and transformed into something that analysis tools can handle. One of the main functions of data preparation is ensuring the accuracy and consistency of the raw data prepared for processing and analysis.

In the data preparation phase, several things were done such as data cleaning. Data cleansing is the process of correcting identified data errors and problems in order to create a complete and accurate data set. For example, as part of the data cleansing process, inconsistent entries are aligned, incorrect data can be deleted or corrected, and missing values are filled in. Removing bias starts with checking for zero values in the data to check if there are zero values in the data. After checking for null values, it is known that there are no null values in the data set.

After the data set is cleaned, if necessary the data will be formatted. This step includes solving problems such as multiple date formats in the data or inconsistent abbreviations or data in text format therefore needs to be converted to numeric format. It is also possible that some data variables are not required for analysis and therefore should be removed from the analysis data set. However, in the crop recommendation data set after analysis, null and unique checks on the data set do not find data that needs formatting. So that the data can be continued for the next process.

After the data has been analyzed and prepared, changes to the data set must be made with extreme caution. Algorithms are usually adjusted during analysis and compared with other results. It is difficult to determine whether the difference in results is due to changes in data or algorithms, because changing data can change the results of the analysis. Following the needs of research on data, the data set used can also be transformed, integrated, and even reduced data.

In the data preparation, separating features and targeting labels are visualized in more depth to understand the readiness of the data before modeling. Figure 6 presents a visualization of the correlation matrix between different features which shows the relationship of one feature to another so that the need for one feature to another is visualized.

Figure 6 presents a joint plot that visualizes the correlation between different features explaining the needs of each plant variable for each variable feature in the data set. For example, we can see that the correlation value between N and pH features is 0.097, and the correlation value between P and K features is 0.74. In total there are 49 correlation plots which show the visualization of the need for 1 variable with other variables, there are 7 variables that are related to 7 other variables resulting in 49 correlation plots.

Figure 7 shows the required values of N (Nitrogen), P (Phosphorus), and K (Potassium) for each crop label. N–P–K are nutrients found in soil so all plants need them. However, the need for N–P–K for each plant is different, if the plant experiences a

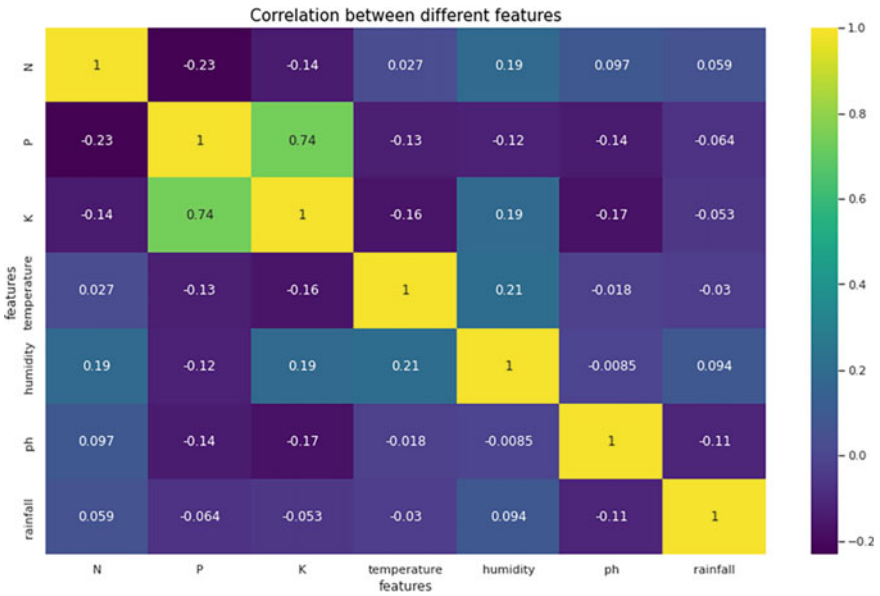


Fig. 6 Correlation between different features

deficiency or excess of these values, the plant will not grow and live effectively. So that the required N–P–K value becomes an independent variable that determines the prediction of what plants are right for planting according to the required value. From Fig. 13 it is known that apple and grape plants have the highest levels of potassium values compared to other plants.

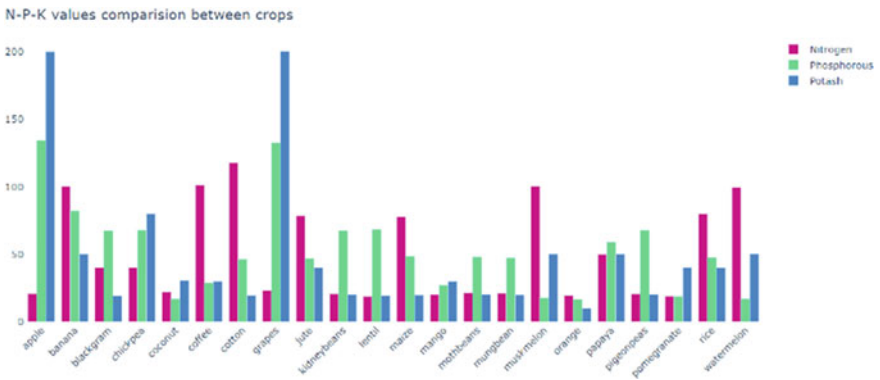


Fig. 7 Comparison of N–P–K requirements between plants

4 Result and Discussion

After the data has gone through the data processing phase, then the data collection will be used in the modeling phase, especially to conduct model training and model testing. This is a mature and widely accepted approach to data mining projects using machine learning algorithms [27]. This methodology offers a form of life-cycle approach in applied artificial intelligence projects [28] so it can be considered as one of the ideal methods for knowledge discovery in database processes (KDD) [29].

Several features are owned by CRISP-DM that can be used to aid evidence mining. This can provide a general process model that includes methodological dimensions and overall structure. This method offers specialization according to various given contexts [13].

4.1 Modeling

In the modeling phase, Random Forest, Naïve Bayes Classifier, K-Nearest Neighbors, Decision Tree, and XGBoost are used to predict crop recommendation. Modeling is carried out by each algorithm model which will then be measured for the accuracy of each algorithm model to find the best algorithm and have the highest accuracy for making predictions on plants. Various stages were carried out in this modeling phase, starting from the imported library, the model was defined, training and model testing was carried out on the data set used until predictions were made and the accuracy of each algorithm used was measured.

In this article, supervised learning is the machine learning model used in crop prediction. Supervised learning is a model machine learning technique, where the model will be given data where each data has a label, in this case a crop label. From these data, the model will be able to predict a label. Usually, in supervised learning there will be training data where all the labels are present and a test data where one of the labels will be removed. After the model learns from the test data, the model will be able to predict data from omitted labels in the test data.

In the process of modeling, predictions are carried out through a collection of data presented after measuring the accuracy value of the training model, measuring the accuracy value of the training model and the accuracy value of the algorithm model using the crop recommendation data set. After the model is trained and training accuracy is measured, then testing is carried out using test data on the model and measuring the accuracy of the test model. After a set or series of data sets suitable for the modeling process is available, through the selected machine learning algorithm, it is necessary to know the characteristics and characteristics of each algorithm, then the validation metrics to be used are determined [13].

K-nearest neighbors (KNN) classification model algorithm is the first algorithm used in the modeling phase, where the KNN algorithm has 20 neighbors with the lowest accuracy at a value of 0.964 for neighbors number 2 and number 4 and the

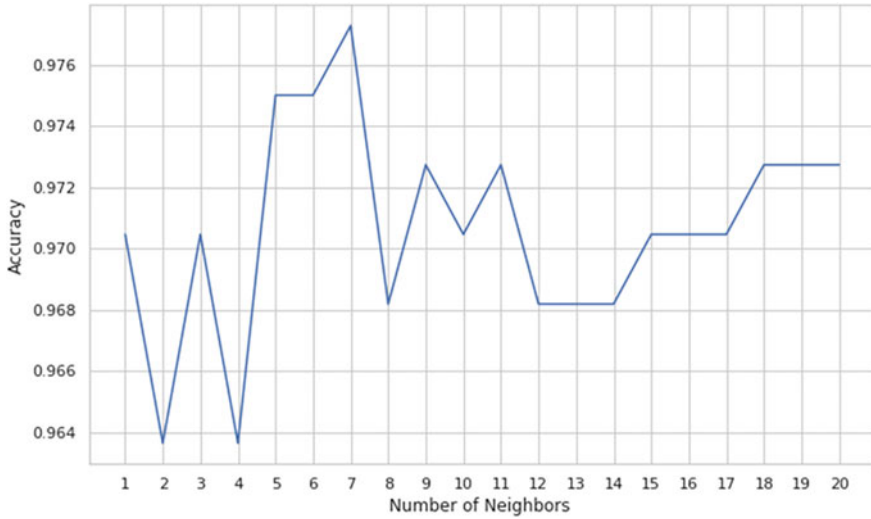


Fig. 8 Number of neighbors and accuracy

highest accuracy at value of 0.978 for neighbors number 7. The number of neighbors and each accuracy for the KNN algorithm can be seen in Fig. 8.

Based on the accuracy measurement results, the KNN algorithm training model has an accuracy value of 0.988 and the accuracy of the test model has a value of 0.975. To be able to improve the accuracy of the measurement results of the training model and model test on the KNN algorithm, it is necessary to do hyperparameter tuning to optimize the KNN algorithm in order to increase its accuracy. Hyperparameter tuning is a key factor that can determine and even find the optimal model architecture. Figure 9 shows a comparison of the accuracy of the KNN algorithm before and after hyperparameter tuning.

After hyperparameter tuning in the KNN algorithm, accuracy measurements are again carried out for training and testing. After hyperparameter tuning is done, there is an increase in the accuracy value of the KNN algorithm, where the accuracy of the training becomes 1.0 and the accuracy of the test becomes 0.972.

The modeling phase is continued by using other classification model algorithms in crop prediction modeling. Random Forest, Naïve Bayes Classifier, Decision Tree,

Fig. 9 Comparison of KNN accuracy before and after

```
Before Hyper Parameter Tuning
knn_train_accuracy = 0.9886363636363636
knn_test_accuracy = 0.975

After Hyper Parameter Tuning
knn_train_accuracy = 1.0
knn_test_accuracy = 0.9727272727272728
```

and eXtreme Gradient Boost (XGBoost) algorithms are used to predict plants with test accuracy above 0.9 for all algorithms. Of these four algorithms, the XGBoost algorithm has the highest accuracy with a test accuracy value of 0.993, and the decision tree algorithm has the lowest accuracy value with a value of 0.9. While the Random Forest and Naïve Bayes Classifier algorithms have the same accuracy value of 0.99.

The modeling phase is continued by using other classification model algorithms in crop prediction modeling. The Decision Tree, Random Forest, Naïve Bayes Classifier, and eXtreme Gradient Boost (XGBoost) algorithms are used to predict plants with test accuracy above 0.9 for all algorithms. Of these four algorithms, the XGBoost algorithm has the highest accuracy with a test accuracy value of 0.993, and the decision tree algorithm has the lowest accuracy value with a value of 0.9. While the Random Forest and Naïve Bayes Classifier algorithms have the same accuracy value of 0.99. Below is a formula to calculate evaluation metrics such as f1 score, recall, and precision.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{\text{precision}} + \frac{1}{\text{recall}} \right)$$

The model must be built by applying the selected technique to the data set. Therefore, the confusion matrix is an alternative to verify and validate model fit. The confusion matrix shows a contingency table of errors and successes created when using a classifier [13]. To be able to calculate and measure evaluation metrics, we need to understand how the confusion matrix works for each machine learning algorithm used. In Figs. 10, 11, 12 and 13, the confusion matrix for Random Forest, Naive Bayes, Decision Tree, and XGBoost algorithm are presented. Through the confusion matrix, we can find out the results of the predicted and actual values for all machine learning algorithm which will be used in measuring accuracy and evaluating metrics.

The measurement results of metric evaluation and modeling accuracy with Random Forest, Naive Bayes, Decision Tree, and XGBoost algorithm are presented from Tables 1, 2, 3 and 4.

From Tables 1, 2, 3 and 4 it is known that the accuracy value of the test model for the Naïve Bayes algorithm is 0.99, the same as the results of the accuracy of the test model for the random forest algorithm. The results of the Naïve Bayes metric evaluation also show that the *F1*-score value for almost all labels has a value of 1.0 except for 2 plant labels, namely jute with a value of 0.93 and rice with a value of 0.86. This shows that the Naïve Bayes algorithm is more effective when compared to the Random Forest and Decision Tree algorithms because only 2 labels have an *F1*-score below 1.0. Table 5 presents a comparison of the metric evaluations of the four machine learning algorithms used.

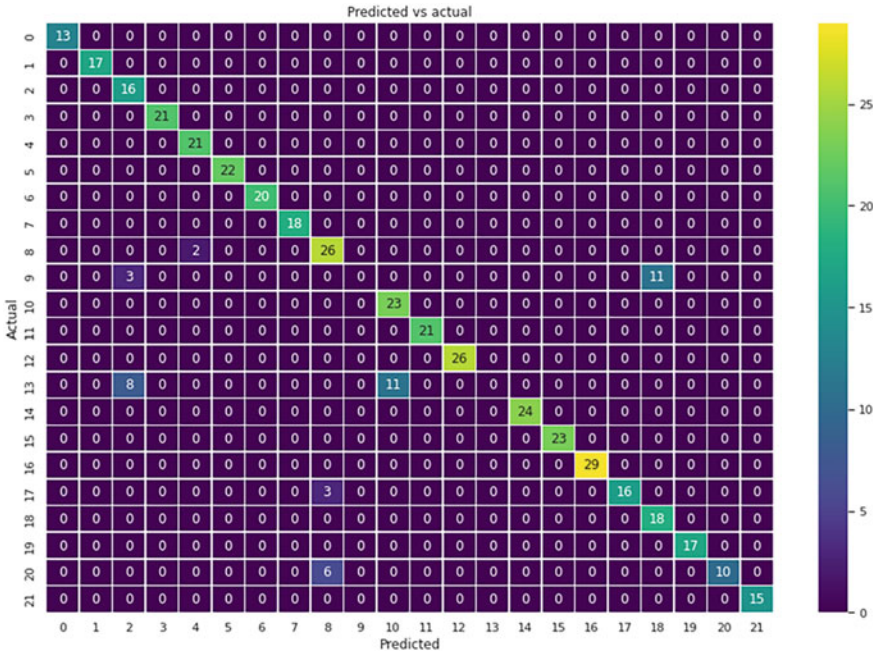


Fig. 10 Confusion matrix decision tree

4.2 Evaluation

In CRISP-DM, the evaluation phase is the final stage of the data mining process, and it is used to assess the quality and effectiveness of the model that has been developed. The model is evaluated by running it on never-before-seen data to assess how effectively it predicts the goal value or addresses the problems discovered. Model validation, model performance evaluation, and result interpretation are all activities carried out during the evaluation stage.

Through the modeling results we can see a comparison of the evaluation of the metrics of the algorithms used. Models are scored based on predefined evaluation metrics, such as accuracy, precision, recall, or *F1*-score. The results of the evaluation are used to determine whether the model is acceptable or not, and if the model is considered bad, the data preparation and modeling stages can be repeated to improve the model.

Machine learning supervised learning models are used in modeling by classifying five machine learning algorithms. Supervised learning is marked by training on the data first before testing which results in accuracy both from the training and test processes for all algorithms. For the KNN algorithm, hyperparameter is used to optimize the model and accuracy. Meanwhile, for other algorithms, it is not implemented. Figure 14 visualizes the comparison of the accuracy of the training model and the test model for each algorithm.

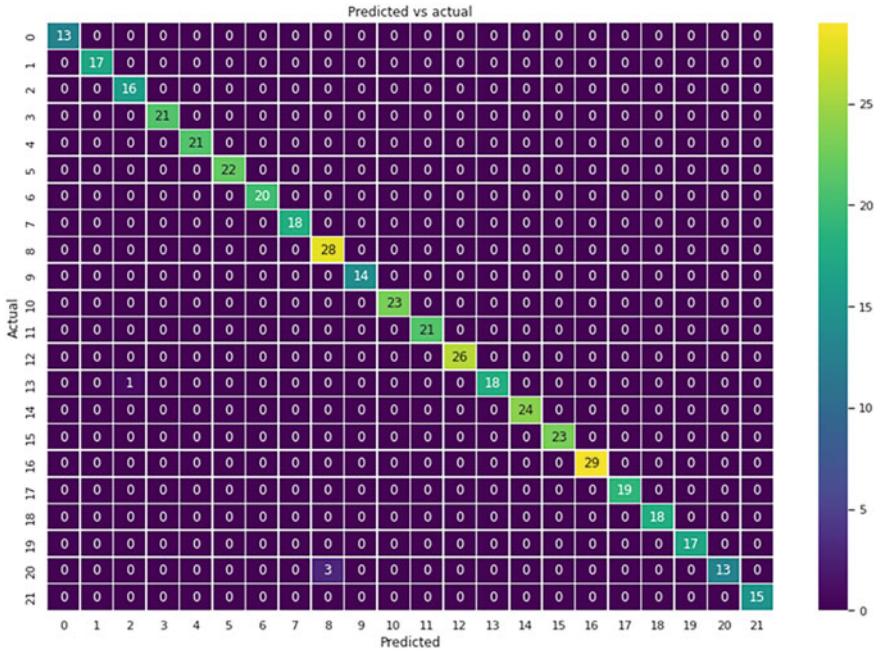


Fig. 11 Confusion matrix random forest

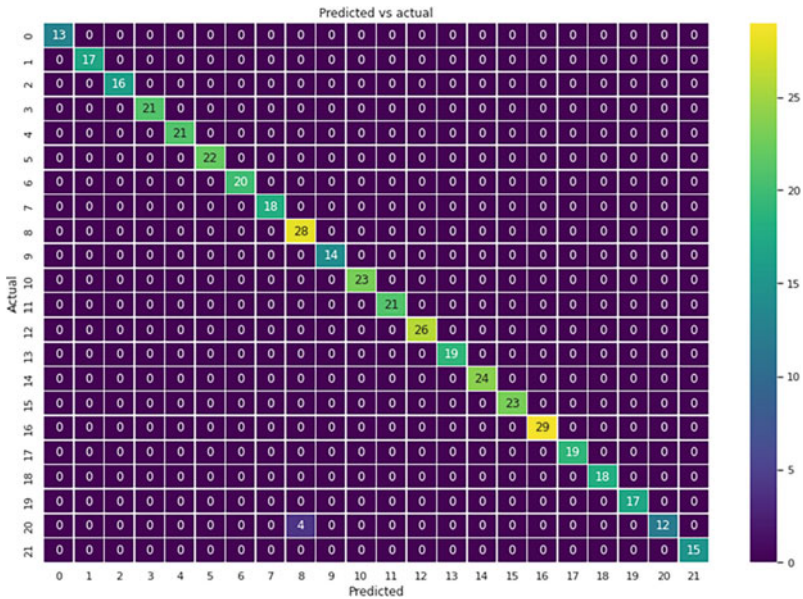


Fig. 12 Confusion matrix Naive Bayes

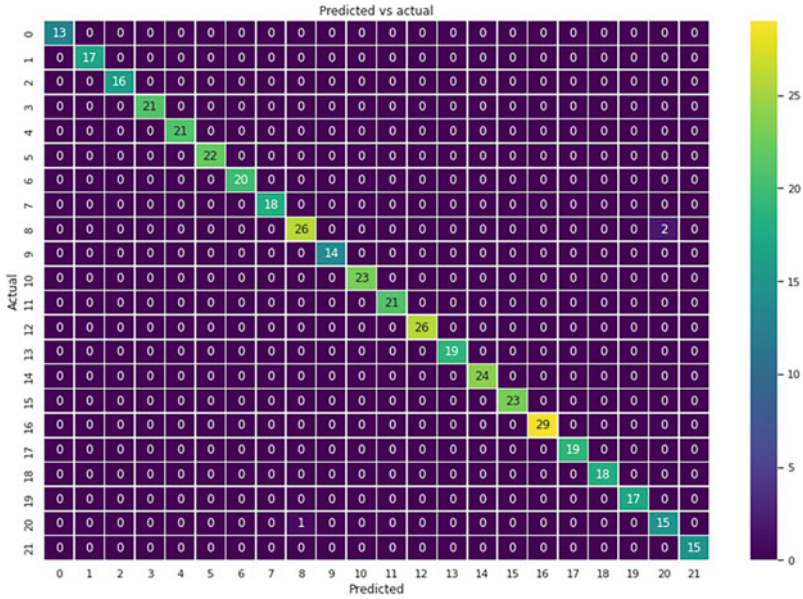


Fig. 13 Confusion matrix XGBoost

From Fig. 15 we can understand the comparison of training accuracy for each machine learning algorithm used. For training accuracy, three algorithms have the highest accuracy with 1.0, namely Random Forest, Naïve Bayes, and XGBoost. The decision tree algorithm gets the lowest training accuracy of the 5 algorithms used with a value of 0.88 while the KNN algorithm before hyperparameter tuning has a value of 0.98. After doing the hyperparameter tuning of the KNN algorithm, the training accuracy value can be optimized to reach a value of 1.0, the same as the Random Forest, Naïve Bayes, and XGBoost algorithms.

Through Table 6 we can see clearly the detailed comparison of the accuracy values of the 5 machine learning algorithms used in making crop predictions. It can be seen that XGBoost is an algorithm with the most optimal accuracy results while the Decision algorithm is an algorithm that has the lowest accuracy.

Table 1 Decision tree metric evaluation

	Precision	Recall	<i>f</i> 1-score	Support
Apple	1.00	1.00	1.00	13
Banana	1.00	1.00	1.00	17
Blackgram	0.59	1.00	0.74	16
Chickpea	1.00	1.00	1.00	21
Coconut	0.91	1.00	0.95	21
Coffee	1.00	1.00	1.00	22
Cotton	1.00	1.00	1.00	20
Grapes	1.00	1.00	1.00	18
Jute	0.74	0.93	0.83	28
Kidneybeans	0.00	0.00	0.00	14
Lentil	0.68	1.00	0.81	23
Maize	1.00	1.00	1.00	21
Mango	1.00	1.00	1.00	26
Mothbeans	0.00	0.00	0.00	19
Mungbean	1.00	1.00	1.00	24
Muskmelon	1.00	1.00	1.00	23
Orange	1.00	1.00	1.00	29
Papaya	1.00	0.84	0.91	19
Pigeonpeas	0.62	1.00	0.77	18
Pomegranate	1.00	1.00	1.00	17
Rice	1.00	0.62	0.77	16
Watermelon	1.00	1.00	1.00	15
Accuracy			0.90	440
Macro avg.	0.84	0.88	0.85	440
Weighted avg.	0.86	0.90	0.87	440

Table 2 Random forest metric evaluation

	Precision	Recall	f1-score	Support
Apple	1.00	1.00	1.00	13
Banana	1.00	1.00	1.00	17
Blackgram	0.94	1.00	0.97	16
Chickpea	1.00	1.00	1.00	21
Coconut	1.00	1.00	1.00	21
Coffee	1.00	1.00	1.00	22
Cotton	1.00	1.00	1.00	20
Grapes	1.00	1.00	1.00	18
Jute	0.90	1.00	0.95	28
Kidneybeans	1.00	1.00	1.00	14
Lentil	1.00	1.00	1.00	23
Maize	1.00	1.00	1.00	21
Mango	1.00	1.00	1.00	26
Mothbeans	1.00	0.95	0.97	19
Mungbean	1.00	1.00	1.00	24
Muskmelon	1.00	1.00	1.00	23
Orange	1.00	1.00	1.00	29
Papaya	1.00	1.00	1.00	19
Pigeonpeas	1.00	1.00	1.00	18
Pomegranate	1.00	1.00	1.00	17
Rice	1.00	0.81	0.90	16
Watermelon	1.00	1.00	1.00	15
Accuracy			0.99	440
Macro avg.	0.99	0.99	0.99	440
Weighted avg.	0.99	0.99	0.99	440

Table 3 Naive Bayes metric evaluation

	Precision	Recall	f1-score	Support
Apple	1.00	1.00	1.00	13
Banana	1.00	1.00	1.00	17
Blackgram	1.00	1.00	1.00	16
Chickpea	1.00	1.00	1.00	21
Coconut	1.00	1.00	1.00	21
Coffee	1.00	1.00	1.00	22
Cotton	1.00	1.00	1.00	20
Grapes	1.00	1.00	1.00	18
Jute	0.88	1.00	0.93	28
Kidneybeans	1.00	1.00	1.00	14
Lentil	1.00	1.00	1.00	23
Maize	1.00	1.00	1.00	21
Mango	1.00	1.00	1.00	26
Mothbeans	1.00	1.00	1.00	19
Mungbean	1.00	1.00	1.00	24
Muskmelon	1.00	1.00	1.00	23
Orange	1.00	1.00	1.00	29
Papaya	1.00	1.00	1.00	19
Pigeonpeas	1.00	1.00	1.00	18
Pomegranate	1.00	1.00	1.00	17
Rice	1.00	0.75	0.86	16
Watermelon	1.00	1.00	1.00	15
Accuracy			0.99	440
Macro avg.	0.99	0.99	0.99	440
Weighted avg.	0.99	0.99	0.99	440

Table 4 XGBoost metric evaluation

	Precision	Recall	f1-score	Support
Apple	1.00	1.00	1.00	13
Banana	1.00	1.00	1.00	17
Blackgram	1.00	1.00	1.00	16
Chickpea	1.00	1.00	1.00	21
Coconut	1.00	1.00	1.00	21
Coffee	1.00	1.00	1.00	22
Cotton	1.00	1.00	1.00	20
Grapes	1.00	1.00	1.00	18
Jute	0.96	0.93	0.95	28
Kidneybeans	1.00	1.00	1.00	14
Lentil	1.00	1.00	1.00	23
Maize	1.00	1.00	1.00	21
Mango	1.00	1.00	1.00	26
Mothbeans	1.00	1.00	1.00	19
Mungbean	1.00	1.00	1.00	24
Muskmelon	1.00	1.00	1.00	23
Orange	1.00	1.00	1.00	29
Papaya	1.00	1.00	1.00	19
Pigeonpeas	1.00	1.00	1.00	18
Pomegranate	1.00	1.00	1.00	17
Rice	0.88	0.94	0.91	16
Watermelon	1.00	1.00	1.00	15
Accuracy			0.99	440
Macro avg.	0.99	0.99	0.99	440
Weighted avg.	0.99	0.99	0.99	440

Table 5 Comparison of metric evaluation

Metric evaluation	Algorithm			
	Decision tree	Random forest	Naive Bayes	XGBoost
Count lowest precision	0.00 (mothbeans)	0.90 (jute)	0.88 (jute)	0.88 (rice)
Count lowest recall	0.00 (mothbeans)	0.81 (rice)	0.75 (rice)	0.93 (jute)
Count lowest <i>F1</i> -score	0.00 (mothbeans)	0.90 (rice)	0.86 (rice)	0.91 (rice)

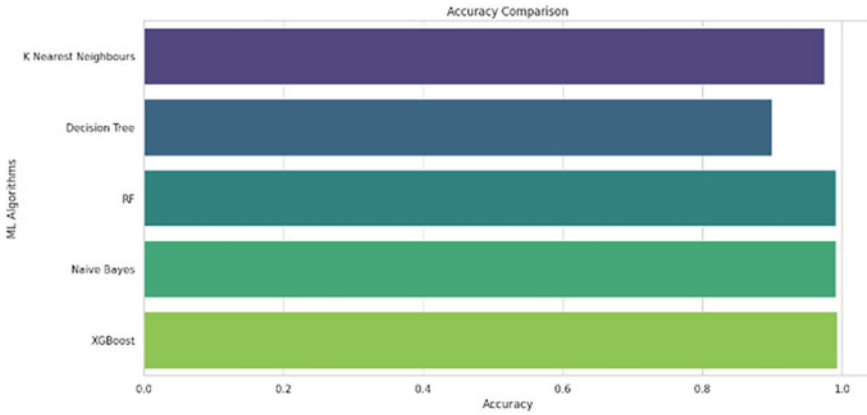


Fig. 14 Comparison of training accuracy

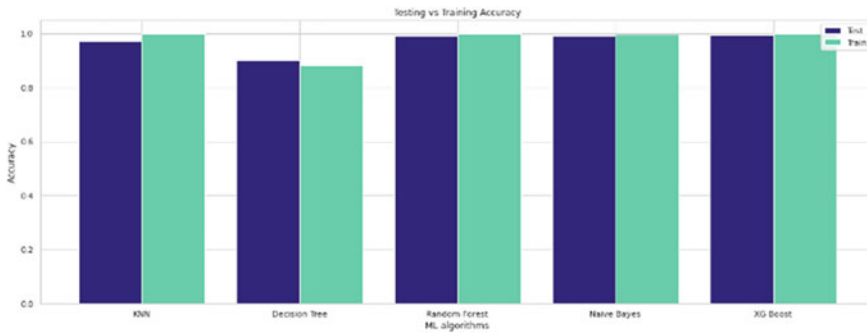


Fig. 15 Comparison of training and test accuracy

Table 6 Detail comparison of training and test accuracy

Accuracy	Algorithm				
	KNN	Decision tree	Random forest	Naive Bayes	XGBoost
Train accuracy	1.0	0.88	1.0	1.0	1.0
Test accuracy	0.972	0.9	0.990	0.990	0.993

5 Conclusion

Crop prediction is carried out using machine learning with the CRISP-DM approach. Each phase in the CRISP-DM is used as a guide for predicting crop recommendations. Starting from the business understanding phase to understand what is the value and meaning behind the data set owned, conducting Exploratory Data Analysis (EDA) in the data understanding phase, preparing and processing data in the data preparation

phase, classifying algorithm models to determine predictions and measuring accuracy in the data preparation phase, and modeling to evaluate and compare the measurement results in the evaluation phase. By using CRISP-DM as a reference and guideline, the steps for making predictions using machine learning are more orderly because the steps in CRISP-DM have the same approach in making predictions using machine learning.

The supervised learning model in machine learning is used to make crop predictions because training will be carried out on the data set before testing. There are five machine learning algorithms used in this study, namely K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Naïve Bayes, and eXtreme Gradient Boost (XGBoost).

Based on the modeling carried out using machine learning algorithms, the accuracy measurement results for each model are obtained for both training and testing. For the accuracy of training the KNN algorithm after hyperparameter tuning has been carried out has the highest accuracy value of 1.0, the same as the three other algorithms which have the highest accuracy of 1.0, namely the Random Forest, Naïve Bayes, and XGBoost algorithms. Meanwhile, the Decision Tree algorithm has the lowest training accuracy value with a value of 0.88. In measuring accuracy testing the XGBoost algorithm has the highest accuracy value with a value of 0.993. While the Decision Tree algorithm has the lowest testing accuracy value with a value of 0.90. The KNN algorithm has the second lowest testing accuracy after the Decision Tree with a value of 0.972. The Random Forest and Naïve Bayes algorithms have the same testing accuracy with a value of 0.990. The results of the comparison of the five machine learning algorithms used, the XGBoost algorithm has the highest training and testing accuracy with a value of 1.0 for training accuracy and 0.993 for testing accuracy.

References

1. Rebala G, Ravi A, Churiwala S (2019) An introduction to machine learning. Springer
2. McQueen RJ, Garner SR, Nevill-Manning CG, Witten IH (1995) Applying machine learning to agricultural data. *Comput Electron Agric* 12(4):275–293
3. Liakos KG, Busato P, Moshou D, Pearson S, Bochtis DD (2018) Machine learning in agriculture: a review. *Sensors* 18
4. van Klompenburg T, Kassahun A, Catal C (2020) Crop yield prediction using machine learning: a systematic literature review. *Comput Electron Agric* 177:105709. <https://doi.org/10.1016/j.compag.2020.105709>
5. Jalandoni A, Zhang Y, Zaidi NA (2022) On the use of machine learning methods in rock art research with application to automatic painted rock art identification. *J Archaeol Sci* 144. <https://doi.org/10.1016/j.jas.2022.105629>
6. Adamchuk VI, Hummel JW, Morgan MT, Upadhyaya SK (2004) On-the-go soil sensors for precision agriculture. *Comput Electron Agric* 44(1):71–91
7. Sharma A, Jain A, Gupta P, Chowdary V (2021) Machine learning applications for precision agriculture: a comprehensive review. *IEEE Access* 9:4843–4873

8. Meshram V, Patil K, Meshram V, Hanchate D, Ramkteke SD (2021) Machine learning in agriculture domain: a state-of-art survey. *Artif Intell Life Sci* 1:100010. <https://doi.org/10.1016/j.aillsci.2021.100010>
9. Däderman A, Rosander S (2018) Evaluating frameworks for implementing machine learning in signal processing: a comparative study of CRISP-DM, SEMMA and KDD
10. Hand DJ (2007) Principles of data mining. *Drug Saf* 30(7):621–622. <https://doi.org/10.2165/00002018-200730070-00010>
11. Kantardzic M (2011) *Data mining: concepts, models, methods, and algorithms*. Wiley
12. Chapman P et al (2000) CRISP-DM 1.0: step-by-step data mining guide. 9(13):1–73
13. Solano JA, Lancheros Cuesta DJ, Umaña Ibáñez SF, Coronado-Hernández JR (2021) Predictive models assessment based on CRISP-DM methodology for students performance in Colombia—Saber 11 test. *Procedia Comput Sci* 198(2020):512–517. <https://doi.org/10.1016/j.procs.2021.12.278>
14. Schröer C, Kruse F, Gómez JM (2021) A systematic literature review on applying CRISP-DM process model. *Procedia Comput Sci* 181(2019):526–534. <https://doi.org/10.1016/j.procs.2021.01.199>
15. Kinghorn D (2022) What is machine learning. <https://www.pugetsystems.com/labs/hpc/What-is-Machine-Learning-758/>. Accessed 21 Nov 2022
16. Das S, Dey A, Pal A, Roy N (2015) Applications of artificial intelligence in machine learning: review and prospect. *Int J Comput Appl* 115(9)
17. Mohd Selamat SA, Prakoonwit S, Sahandi R, Khan W, Ramachandran M (2018) Big data analytics—a review of data-mining models for small and medium enterprises in the transportation sector. *Wiley Interdiscip Rev Data Min Knowl Discov* 8(3):e1238
18. El Naqa I, Murphy MJ (2015) What is machine learning? In: *Machine learning in radiation oncology*. Springer, pp 3–11
19. Ayodele TO (2010) Types of machine learning algorithms. *New Adv Mach Learn* 3:19–48
20. Roiger RJ (2017) *Data mining: a tutorial-based primer*. Chapman and Hall/CRC
21. Paul M, Vishwakarma SK, Verma A (2015) Analysis of soil behaviour and prediction of crop yield using data mining approach. In: *2015 international conference on computational intelligence and communication networks (CICN)*, pp 766–771
22. Bhargavi P, Jyothi S (2009) Applying Naive Bayes data mining technique for classification of agricultural land soils. *Int J Comput Sci Netw Secur* 9(8):117–122
23. Gholap J (2012) Performance tuning of J48 algorithm for prediction of soil fertility. arXiv preprint [arXiv:1208.3943](https://arxiv.org/abs/1208.3943)
24. Ramesh D, Vardhan BV (2013) Data mining techniques and applications to agricultural yield data. *Int J Adv Res Comput Commun Eng* 2(9):3477–3480
25. Patel H, Patel D (2014) A brief survey of data mining techniques applied to agricultural data. *Int J Comput Appl* 95:9
26. Wirth R, Hipp J (2000) CRISP-DM: towards a standard process model for data mining. In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol 1, pp 29–39
27. Altaleb M, Deeken H, Hertzberg J (2022) A data mining process for building recommendation systems for agricultural machines based on big data. 42. *GIL-Jahrestagung, Künstliche Intelligenz der Agrar. Ernährungswirtschaft*
28. Ayele WY (2020) Adapting CRISP-DM for idea mining a data mining process for generating ideas using a textual dataset. *Int J Adv Comput Sci Appl* 11(6):20–32
29. Wirth R (2000) CRISP-DM: towards a standard process model for data mining. *Proc Fourth Int Conf Pract Appl Knowl Discov Data Min* 24959:29–39

Lung Cancer Detection and Classification Model Using Inception V3 Algorithm



Sitaram Meena, Amod Kumar, Meenakshi Sood, and Rajesh Kumar Meena

Abstract Lung cancer is one of the primary cancer-related causes of death worldwide. Early lung cancer detection can significantly increase the likelihood of successful treatment. In this research, we propose the Inception V3 algorithm as a “deep learning”-based method for detecting lung cancer. The performance of the system is measured using measures like accuracy, precision, and recall on a dataset comprised of chest CT pictures during both training and testing. With an accuracy of 94.98% and a precision of 95%, our findings demonstrate that the suggested lung cancer diagnosis method is highly effective. These findings demonstrate the potential for deep learning algorithms to aid radiologists in the early diagnosis of lung cancer.

Keywords Lung cancer nodules · Deep learning · Image processing · Inception V3

1 Introduction

Lung cancer (Fig. 1) is one of the most prevalent and lethal cancers, with a significant mortality rate. Early lung cancer detection is essential for increasing the likelihood of successful treatment. Traditional methods of lung cancer detection, such as visual examination of chest X-rays and CT scans, may be unable to detect small or early-stage tumors accurately.

S. Meena (✉) · R. K. Meena

Department of Electronics, Rajesh Pilot Government Polytechnic College, Dausa, India

e-mail: sitaram065@gmail.com

A. Kumar

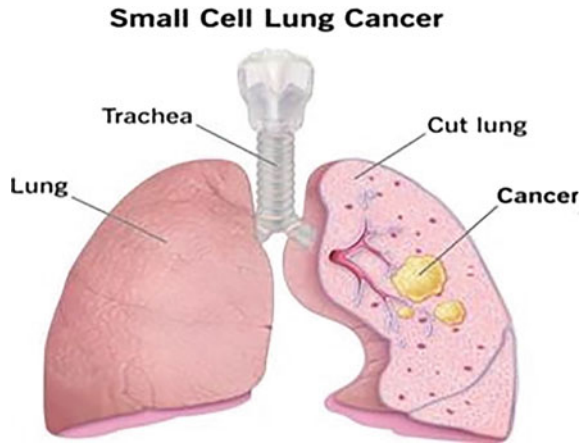
Department of ECE, NITTTR, Chandigarh, India

M. Sood

Department of Curriculum, NITTTR, Chandigarh, India

e-mail: meenakshi@nitttrchd.ac.in

Fig. 1 A small cell lung cancer nodule



1.1 Motivation

The application of deep learning, a type of machine learning, to the analysis of medical images has shown much promise. “Convolutional neural networks” (CNNs) and other deep learning algorithms are well-suited for tasks like tumor diagnosis because of their ability to learn complicated patterns and features in medical images. In this research, we suggest a deep learning-based strategy for identifying lung cancer using the Inception V3 algorithm. The Inception V3 algorithm is a CNN that is pre-trained on a large dataset of images and is effective in various image classification tasks. Algorithm performance is measured by several different measures, including accuracy, precision, f_1 score, and recall, and is tested and trained using a dataset of chest X-ray pictures. Our mission is to show how deep learning algorithms may help radiologists make earlier diagnoses of lung cancer, which in turn improves patient outcomes. It was found that results from systems based on cutting-edge techniques were more accurate than those from methods based on conventional AI techniques.

1.2 Deep Learning

Deep learning is a subpart of machine learning (ML) based on multiple layered artificial neural networks (ANNs). The term “deep” refers to the number of layers in the network, which can range from dozens to hundreds.

Deep learning has been applied to many diverse fields and has proven to be very successful in many of them. It has been utilized to enhance the efficacy of speech and image recognition, natural language processing, and predictive modeling.

1.3 Inception Method in Deep Learning

The Inception method is a technique used in “deep learning” to improve the performance of convolutional neural networks (CNNs). It was first introduced in the Inception architecture, which was developed by Google researchers in 2014. The key idea behind the Inception method is to use a combination of different convolutional filter sizes in the same layer, rather than using a single filter size.

The Inception architecture consists of multiple Inception modules, each of which contains a set of convolutional layers with different filter sizes. The different filter sizes are used to detect different types of features in the input image, such as edges, textures, and shapes. The outputs of the different filters are then combined and passed to the next layer of the network.

2 Literature Survey

Various methods have been utilized to detect lung cancer in its earliest phases. This study compares and contrasts a variety of machine learning-based strategies for detecting lung nodule early on. Most of the detection methods depends upon CT film pictures while few make utilization of X-ray imaging.

Huang et al. [1] proposed a model to detect cellular breakdown in the lungs, and created a breath test merging AI computing with a chemical sensor. Between 2016 and 2018, one alveolar air test was analyzed using carbon nanotube sensor clusters, and a planned report led to the recording of cellular breakdown events in the lungs and non-growth controls.

Md. Sakif Rahmani et al. [2] proposed a cellular breakdown in the lungs identification and expectation technique utilizing the profound brain organization. The proposed procedure is utilized to identify the cellular breakdown in the lungs in its beginning phase and to foresee the cellular breakdown in the lungs. The proposed strategy works in two stages: Firstly, the CT pictures are preprocessed through obscuring and thresholding to work on the quality and essentially the pictures, achieved 95.60% accuracy with 0.387732 log loss.

The cutting-edge Entropic Degradation Method (EDM) was extensively employed by researchers to identify Small Cell Lung Cancer (SCLC) in computed tomography (CT) images.

Wu, Qing and Zhao et al. [3] gave the model for early diagnosis of lung cell damage. The last two outputs were each given five arbitrary sweeps to choose from to evaluate the models. The suggested calculation has a success rate of 77.8%.

Singh and Gupta (2022) have fostered a strategy for ordering and separating CT check-related pictures of cellular breakdown in the lungs into threatening and harmful stages. In the proposed strategy, these pictures are first handled utilizing picture handling procedures, and afterward, the examination calculations are controlled to isolate them. Stable components are separated notwithstanding numerical data,

and different result components are given by the dividers. Also, there are seven distinct classifications utilized. The supporting vector stage, the k -segment of the following stage, the choice tree, the stochastic inclination plummet stage, the multi-facet perceptron (MLP) stage, and the arbitrary timberland stage is instances of numerous inconsequential Bayes stages. Also, 15,750 clinical pictures from 8840 and 6910 malignant growth-related data were utilized to prepare and assess these channels.

Reddy et al. incorporated pre-handling, picture collection, thresholding, binarization, characteristic extraction, division, and acknowledgment of the brain. It was observed that the results from techniques depend upon deep learning methods showed better correctness if weighs with methodologies those had employed using typical machine learning methodologies.

According to a comparable review of the state of the DL technique arts now, the suggested DL method provides a higher level of accuracy in comparison to the current frameworks.

Shakeel et al. [4] utilized deep learning to improve the multidisciplinary approach (IPCT) and utilized the Instantaneously Trained Neural Networks (ITNN) strategy to anticipate cellular breakdown in the lungs in CT pictures. Lung CT checks were first found on the Cancer Imaging Archive (CIA) site, which involves 5043 DICOM pictures isolated by 2043 imaging pictures and 3000 preparation pictures. Picture quality was subsequently improved with PC disentanglement, which supplanted the pixel, utilizing a potential conveyance cycle and assortment. After further developing the picture portrayal, the impacted region was isolated utilizing a pixel-like worth. Assortments are created based on a proportion of the similitude of the result of ghostly related structures. Partition techniques were utilized to prepare and separate highlights, which were found to precisely foresee malignant growth up to 98.42% of the time with a little detachment blunder of 0.038. Utilizing UNet design.

Bhandary et al. (2020) [5] utilized the changed Alex Net (MAN) proposed to analyze lung problems, Chest X-Ray and CTs for the lungs are two sorts of imaging. The proposed MAN is independently tried on these two picture datasets. X-Ray thoracic X is tried as expected during the underlying symptomatic technique, and pneumonia is not entirely settled as displayed in Fig. 7. Contrasted and other DL strategies thinking about this audit, the proposed DL technique gives 96% precision.

3 Proposed Methodology

We conducted a thorough evaluation of researches for the detection of lung nodules employing deep learning and inferred that convolution neural networks utilizing the Inception approach would be more effective and useful to identify lung illness in light of our findings. CNN's Inception model was utilized in the suggested work. In our setup, we trained and tested our model using CT scan images. We employed a fully linked layer, max pooling, and 2D convolution [2].

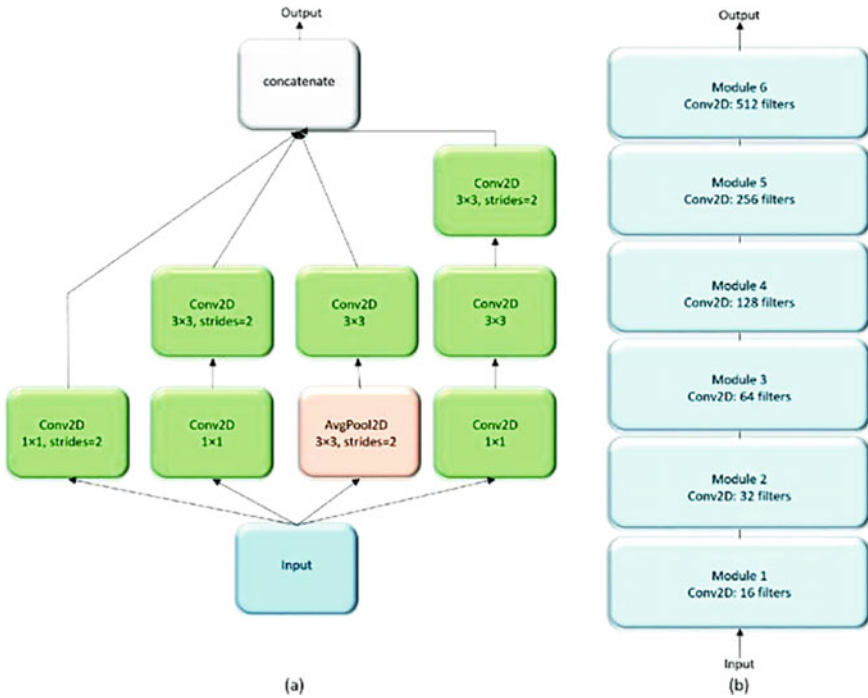


Fig. 2 Architecture of the Inception V3 model [6]

(i) **Model Architecture for Inception V3**

The Inception V3 model, called as naïve model, has 48 layers overall and a minimized false rate than its forerunners. Figure 2 shows the architecture of the Inception V3 model. The layered architecture of the Inception V3 model is shown in Fig. 3.

(ii) **Performance of Inception V3**

The coding was done for the Inception V3 model (Fig. 4). On evaluation, it proved better in accuracy with less computational cost as compared to the naive Inception version.

(iii) **Data Augmentation**

To further enhance the accuracy of the suggested method [7], utilizing transformations such as scaling, rotation, and contrast adjustment, data augmentation is used to fetch meaningful training samples from already available training datasets. By artificially expanding the sample amount, data augmentation makes DNN more resilient and able to avoid overfitting issues, allowing for the generation of many more images with the same output. Each image is flipped horizontally, resized, sheared, and zoomed. To complete the objective, this study makes use of an image data generator. In this case, we apply a rescaling transformation to the image to achieve a more

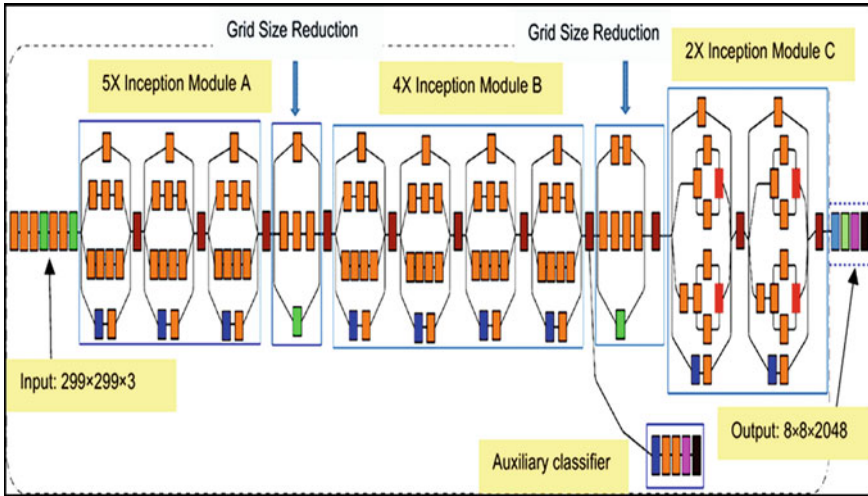


Fig. 3 Layered architecture of Inception V3

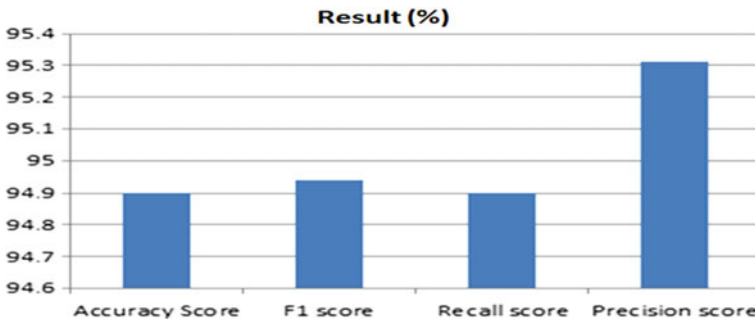


Fig. 4 Result of proposed methodology

uniform appearance, and the scaling factor we use is $1/255$. In the suggested work, a value of 0.2 is chosen for shearing, although this method can be applied with values from 0 to 1 to yield outstanding results. With the zoom set to 0.2, the zoom range was completely arbitrary between 0 and 20%. If you toggle the Horizontal flip on, the image will be randomly rotated 90° to the left or right.

4 Results and Discussion

Python packages like Keras and TensorFlow were used for the experiments, and they were run on a Windows machine equipped with an i5-1135G7 CPU. Kaggle’s dataset of lung CT images was used to acquire data for this analysis, which was then split

Table 1 Performance of proposed work

Performance metrics	Result
Accuracy score	0.9498
F_1 score	0.9495
Recall score	0.9498
Precision score	0.9511

72:28 across training and testing sets. A total of 1382 lung CT scans were used in the training set, whereas 400 scans were used in the validation set. A total of 1382 CT images were divided into training and validation sets using the suggested method. From the 1782 CT scans, 1372 are used to teach the model and 400 are used to test it after each training iteration. In the simulation, we used a total of 613 images of four categories of lung cancer and we applied vertical and horizontal augmentation and set zoom range and shear range equal to 0.2 and fill mode to the nearest value on this data set that yielded a total of 1013 images for training purposes. In Python, “Fill mode” typically refers to the strategy used to handle missing values in a dataset. As shown in Table 1, it is recommended to use the default values for these performance parameters. Additionally, 16-person batches are now the norm. Therefore, the DNN model did very well. Its accuracy was 94.98%. In addition to accuracy, this study also looks at the crucial metrics of recall and F_1 score to gauge overall performance. The summary is shown in Fig. 4. The proposed Inception V3 model has 48 layers that pre-trained the model on millions of images with 1000 object categories, hence this model has proved better for the proposed work.

The accuracy and loss curves for training and validation are shown in Fig. 5. Epoch 30 yields the highest validation accuracy, while Epoch 28 shows the lowest validation loss. As a result, it is understood that the loss lowers and the accuracy improves as the no of epochs grows.

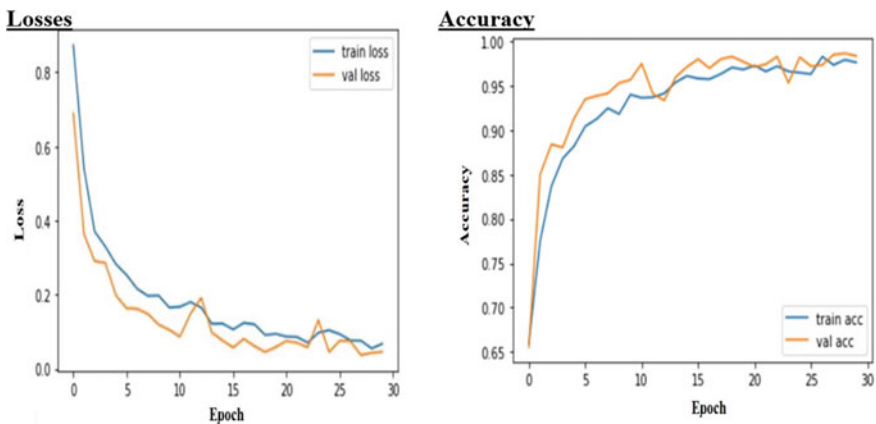


Fig. 5 Training and validation accuracy and losses

During experimentation, the epoch was set to 30 and the step size of 43 hence we got maximum validation accuracy at epoch 30 and minimum validation loss at epoch 25.

Its confusion matrix is shown in Fig. 6. Recall, precision, and F_1 score for the cancer test data are all 0.94, 0.95, and 0.94. Figure 7 displays that 86 of 100 images of adenocarcinoma malignant data will be positive (TP) and 13 will be negative (FN), 97 of 100 images of large cell carcinoma malignant data will be false positive (FP), 100 of the normal data will be TP, and 96 of the squamous cell carcinoma data will be TP, with only 4 being falsely positive.

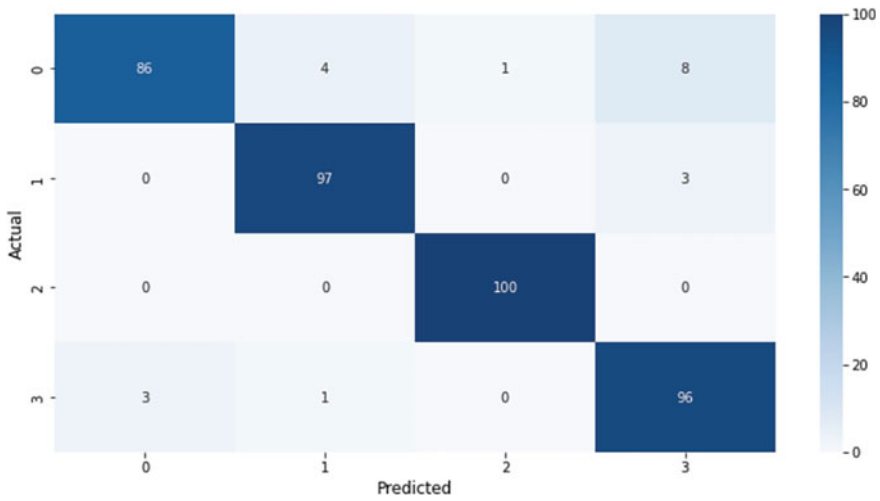


Fig. 6 Confusion matrix for proposed work

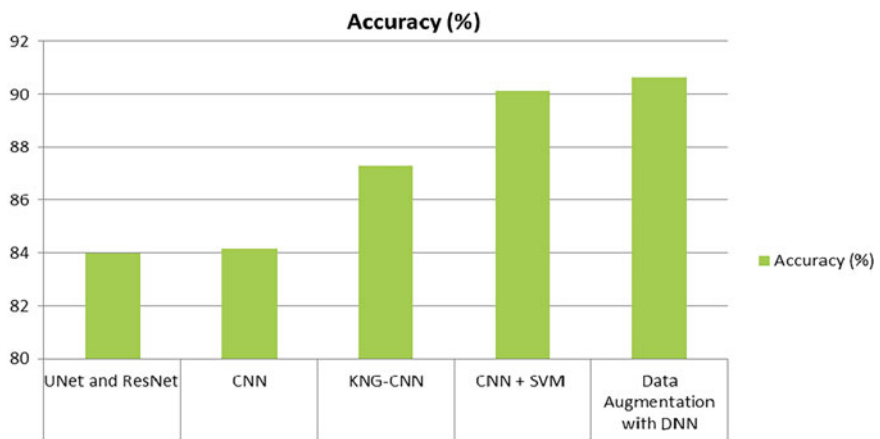


Fig. 7 Performance of the proposed work

Table 2 and Fig. 7 present a comparison of the proposed work’s performance to that of existing works. All of the presently available CAD methods for lung cancer categorization in Table 2 have acceptable accuracies. When determining how to categorize a CAD system, it is crucial to think about both the size of the datasets and the small nodule size.

Figure 7 provides a visual representation of the accuracy metric performance of the proposed methodology.

The performance of the proposed work based on the precision metric is compared with the existing works and the values of the performance are given in Table 3 and Fig. 8.

The precision metric is compared with existing research and shown by the graph in Fig. 8.

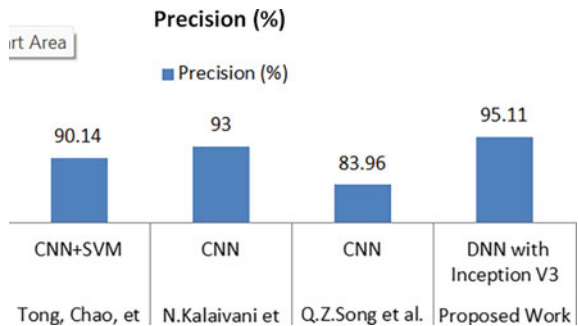
Table 2 Dataset performance utilizing the proposed augmented DNN

Method	Architecture	Accuracy (%)	Year
Bhatia et al. [8]	UNet and ResNet	84	2019
Song et al. [3]	CNN	84.15	2017
Jena et al. [9]	KNG-CNN	87.3	2020
Chao et al. [10]	CNN+SVM	90.14	2021
Proposed work	DNN with Inception V3	94.98	2023

Table 3 Precision metric comparison of proposed work with state of the art

Methods	Architecture	Precision (%)	Year
Chao et al. [10]	CNN+SVM	90.14	2021
Kalaivani et al. [11]	CNN	93	2022
Song et al. [3]	CNN	83.96	2017
Proposed work	DNN with Inception V3	95.11	2023

Fig. 8 Performance comparison of precision metric



Since the proposed work uses a smaller data dimension (224×224 pixels), a larger batch size (16), and a smaller kernel size (1.0 by 1.0) in its experiments, as well as augmentation techniques with fill mode to increase the size of the datasets, it achieves better results than the existing.

5 Conclusion and Future Work

The use of the Inception V3 deep learning model for lung cancer detection has shown promising results in accurately identifying and classifying lung tumors in medical images. However, further research and testing are needed to improve the performance of the model and to ensure its effectiveness in a clinical setting. Some possible directions for future work could include incorporating more diverse and representative data sets, fine-tuning the model's architecture, and experimenting with different pre-processing techniques. Additionally, it may be beneficial to explore the use of other deep learning models in combination with Inception V3 to improve overall performance.

References

1. Huang GZ, Liu L, Maaten VD, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the 30th IEEE conference on computer vision and pattern recognition, CVPR 2017, vol 2017-Jan, pp 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
2. Khan S, Rahmani H, Shah SA, Bennamoun M (2018) A guide to convolutional neural networks for computer vision. Synth. Lect. Comput. Vis. 8(1):1–207. <https://doi.org/10.2200/s00822ed1v01y201712cov015>
3. Song QZ, Zhao L, Luo XK, Dou XC (2017) Using deep learning for classification of lung nodules on computed tomography images. J Healthc Eng 2017. <https://doi.org/10.1155/2017/8314740>
4. Shakeel PM, Burhanuddin MA, Desa MI (2019) Lung cancer detection from CT image using improved profuse clustering and deep learning instantaneously trained neural networks. Meas. J. Int. Meas. Confed. 145:702–712. <https://doi.org/10.1016/j.measurement.2019.05.027>
5. Pham D, Bhandari S, Pinkston C, Oechsli M, Kloecker G (2020) Lung cancer screening registry reveals low-dose CT screening remains heavily underutilized. Clin Lung Cancer 21(3):e206–e211
6. <https://iq.opengenus.org/inception-v3-model-architecture>
7. Alakwaa W, Nassef M, Badr A (2017) Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). Int J Adv Comput Sci Appl 8(8):409–417. <https://doi.org/10.14569/ijacsa.2017.080853>
8. Bhatia S, Sinha Y, Goel L (2019) Lung cancer detection: a deep learning approach. Adv Intell Syst Comput 817:699–705. <https://doi.org/10.1007/978-981-13-1595-4-55>
9. Jena SR, George ST (2020) Morphological feature extraction and KNG-CNN classification of CT images for early lung cancer detection. Int J Imaging Syst Technol 30(4):1324–1336. <https://doi.org/10.1002/ima.22445>
10. Chao T et al (2021) Pulmonary nodule classification based on heterogeneous features learning. IEEE J Sel Areas Commun 39:574–581

11. Kalaivani N, Manimaran N, Sophia, Devi D (2020) Deep learning based lung cancer detection and classification. *IOP Conf Ser Mater Sci Eng* 994:012026
12. Doi K (2007) Computer-aided diagnosis in medical imaging: Historical review, current status, and future potential. *Comput Med Imaging Graph* 31(4–5):198–211. <https://doi.org/10.1016/j.compmimag.2007.02.002>
13. Choi WJ, Choi TS (2014) Automated pulmonary nodule detection based on three-dimensional shape based feature descriptor. *Comput Methods Programs Biomed* 113(1):37–54. <https://doi.org/10.1016/j.cmpb.2013.08.015>
14. Mithila EE, Kumar SS (2017) Automatic detection of solitary pulmonary nodules using swarm intelligence optimized neural networks on CT images. *Eng Sci Technol Int J* 20(3):1192–1202. <https://doi.org/10.1016/j.jestch.2016.12.006>
15. Mahto MK, Bhatia K, Sharma RK (2021) Deep learning based models for offline Gurmukhi handwritten character and numeral recognition. *Electron Lett Comput Vis Image Anal* 20(2):69–82. <https://doi.org/10.5565/rev/elcvia.1282>
16. Kriegsmann M et al (2020) Deep learning for the classification of small-cell and non-small-cell lung cancer. *Cancers (Basel)* 12(6):1–15. <https://doi.org/10.3390/cancers12061604>
17. Dou Q, Chen H, Yu L, Qin J, Heng PA (2017) Multilevel contextual 3-D CNNs for false positive reduction in pulmonary nodule detection. *IEEE Trans Biomed Eng* 64(7):1558–1567. <https://doi.org/10.1109/TBME.2016.2613502>
18. Sori WJ, Feng J, Godana AW, Liu S, Gelmecha DJ (2022) DFD-Net: lung cancer detection from denoised CT scan image using deep learning. *Front Comput Sci* 15(2). <https://doi.org/10.1007/s11704-020-9050-z>; Bushara AR et al (2022) *Electron Lett Comput Vision Image Anal* 21(1):130–142
19. Joshua Neal ES, Bhattacharyya D, Chakkravarthy M, Byun YC (2021) 3D CNN with visual insights for early detection of lung cancer using gradient-weighted class activation. *J Healthc Eng* 2021. <https://doi.org/10.1155/2021/6695518>
20. Kang X, Song B, Sun F (2019) A deep similarity metric method based on incomplete data for traffic anomaly detection in IoT. *Appl Sci* 9(1). <https://doi.org/10.3390/app9010135>
21. Zhu W, Zeng N, Wang N (2010) Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS[®] implementations. In: *Northeast SAS users group 2010. Health care and life sciences*, pp 1–9
22. Saikia T, Hansdah M, Singh KK, Bajpai MK (2022) Classification of lung nodules based on transfer learning with K-nearest neighbor (KNN). In: *IEEE international conference on imaging systems and techniques (IST)*, pp 1–6
23. Pham D, Bhandari S, Pinkston C, Oechsli M, Kloecker G (2020) Lung cancer screening registry reveals low-dose CT screening remains heavily underutilized. *Clin Lung Cancer* 21(3):e206–e211

Real-Time Recognition of Handwritten Characters Using CNN



Ritesh Kumar and Ritik Rao

Abstract The objective of our paper is to create a system that can recognize and classify characters written in three dimensions in the air based on a variety of parameters. Air-writing is a revolutionary approach for producing linguistic signals or words in free space through the use of hand or finger movements. In our paper, we are using the object of a pen's tip to write in the air. This approach allows for greater precision and control in the writing process. This paper could combine computer vision and machine learning for handwriting recognition. The air-writing recognition system uses the computer's digital camera to track the movement of the pen's tip as it writes characters and numbers in the air. Using a convolutional neural network, the letters, numerals, and other symbols are then classified into one of the available classes. While many previous systems rely on complex and costly pursuit configurations, we created a system that can recognize gestures using a significantly simpler and less costly pursue configuration. The proposed model achieved an overall accuracy of 89.62%.

Keywords Handwriting recognition · Gesture recognition · Computer vision · Machine learning · Convolutional neural network · Digital camera · Object tracking

1 Introduction

Hand gestures play a critical role in computer vision and human–computer interface, as researchers seek ways for people to interact more naturally and intuitively with machines. Hand gesture recognition has a wide range of applications beyond sign language interpretation, including video conferencing, menu navigation, and alphanumeric character recognition. In recent years, there has been a surge of interest in the development of algorithms for recognizing hand gestures and motion patterns [1–5].

R. Kumar (✉) · R. Rao
Delhi Technological University, Bawana Road, Rohini, New Delhi 110042, India
e-mail: riteshks1111@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
A. Swaroop et al. (eds.), *Proceedings of Data Analytics and Management*, Lecture Notes in Networks and Systems 787, https://doi.org/10.1007/978-981-99-6550-2_33

435

Hidden Markov Models (HMMs) have gained popularity in handwriting identification and spatiotemporal pattern recognition due to their ability to learn model parameters from observation sequences using training methods like Forward–Backward, Baum–Welch, and Viterbi [6, 7]. In this work, HMMs were used in combination with object (pen’s tip) motion trajectories to recognize alphabetic characters [8].

While publicly available web databases contain recordings of people’s gestures, these recordings typically track the movements of the mass center of the object (pen’s tip) or hand [9]. For this experiment, a new dataset was created, consisting of spatiotemporal patterns of object (pen’s tip) motion trajectories corresponding to writing gestures. Because of their respective differences, the trajectories had been adjusted so that they could be interpreted as a set of directional angles which were then converted to observation symbols indicative of gesture through Time-Independent Technique. Finally, an HMM was built for each letter, based on the observation symbols [8].

2 Related Work

Renata F. P. Neves et al. presented an offline technique for the recognition of handwritten digits that was based on Support Vector Machines (SVMs) in the article [7]. The authors of the study believe that the SVM classifier is superior to the Multilayer Perceptron (MLP) classifier in terms of its overall performance. In their evaluations, they made use of the NIST SD19 benchmark dataset. One of the benefits of the MLP is that it can differentiate between classes in a way that is not possible using linear differentiation. If MLP believes that it has found the best possible spot on the error surface, it may enter a local minimum area, which is an area where training is stopped. Another one of MLP’s many flaws is that it is unable to select the optimal network architecture for the solution of a problem while simultaneously taking into account the total number of layers and perceptron’s present in each hidden layer. As a consequence of these constraints, an MLP-based digit recognizer might not be able to achieve a satisfactory level of accuracy.

In a subsequent study, published in [10], the authors proposed a system that utilizes the color and depth information obtained from a Kinect sensor for detection of hand shape. However, even with the Kinect sensor, gesture recognition remains a difficult challenge. The Kinect sensor’s resolution is only 640×480 , which is sufficient for tracking a large object like the human body, but it is challenging to track small objects like fingers. Therefore, despite the advancements in technology, gesture recognition remains a challenging problem that requires further research.

Furthermore, Moni and Shawkat Ali [11] proposed an enhanced HMM-based method for hand gesture recognition by incorporating prior knowledge about the spatial structure of hand movements. Their approach utilized a graph-based representation of hand postures, which was then integrated into an HMM framework. This method effectively captured the spatial and temporal characteristics of hand

gestures, achieving notable performance gains compared to traditional HMM-based approaches.

3 Methodology

This study presents a technique for classifying object (pen’s tip) and hand motion trajectory-based gestures using Hidden Markov Models (HMMs). The technique consists of three stages: feature extraction, preprocessing, and classification.

3.1 Preprocessing

The preprocessing stage was developed by Guillermo Garcia-Hernando, a Ph.D. candidate at Imperial College’s Computer Vision and Learning Lab, and involves object (pen’s tip) tracking using a GoPro camera and an infrared depth sensor. The object (pen’s tip)’s motion trajectory is captured and saved as pictures and Cartesian system coordinates frame-by-frame. The dataset used in this study consists of 260 recorded frame sequences, with each letter represented by ten recorded frame sequences.

The next stage involves feature extraction, which includes refining the gesture path and quantizing the orientation to identify discrete vectors or observation symbols.

Finally, the gesture is recognized and classified using “discrete vector” and a “Left–Right Banded model”. The proposed approach shows promise in accurately recognizing alphabet characters based on object (pen’s tip) and hand motion trajectories (Fig. 1).

To achieve optimal performance, the gesture path must be described using a comprehensive set of characteristics. The three essential elements in this process are location, orientation, and speed. Among these elements, orientation has been the focus of research [5–7], followed by velocity and position. In this study, the object (pen’s tip) gesture trajectory is described using orientation as a feature. The researchers contend that the spatiotemporal pattern of an object’s placement, such as the tip of a pen, can be effectively captured by a gesture trajectory. By quantizing the orientation of the trajectory, the researchers generate discrete vectors or observation symbols, which are then used in the Hidden Markov Model (HMM) for classification. Overall, this approach shows promise in accurately recognizing alphabet characters based on object (pen’s tip) and hand motion trajectories.

$$L_t = (x_t, y_t), (t = 1, 2, 3, \dots, T). \quad (1)$$

The frame number is represented by the variable t , where T is the last frame and denotes the end of a gesture. The orientation feature is represented by the symbol “ t ”, and it is calculated using subsequent frames along the gesture path (as seen in Fig. 2).

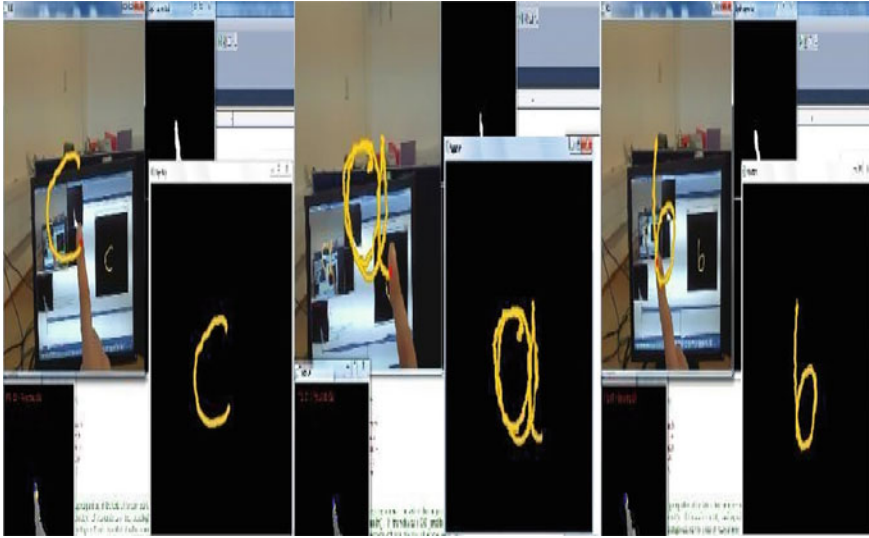


Fig. 1 Trajectory motion of detected tip

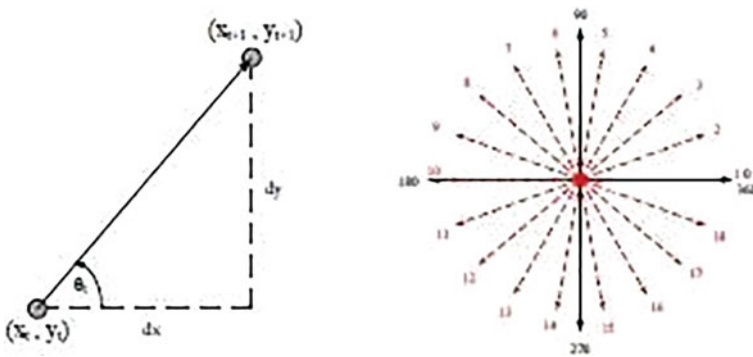


Fig. 2 Its codewords and orientation

Specifically, the orientation at each frame is calculated as the angle between the vector from the object (pen’s tip) to the wrist and the x -axis of the camera coordinate system. By quantizing the orientation feature into discrete values, the researchers generate observation symbols that are used in the Hidden Markov Model (HMM) for classification.

$$\theta_t = \arctan \frac{y_{t+1} - y_t}{x_{t+1} - x_t}. \tag{2}$$

3.2 Feature Extraction

The complexity and shape of each gesture affect the size of the feature vectors obtained from that gesture, which in turn affects processing speed. Therefore, before comparing feature vectors, data alignment is necessary. A data alignment method proposed in [8] guarantees that the length of feature vectors for each gesture is consistent.

3.3 Classification

In the LRB topology, each state can only transition to itself or to a state within a fixed range of its own index, ensuring that the state sequence is strictly increasing. The fixed range is known as the bandwidth, and the model is referred to as a banded model. The banded model reduces the number of possible transitions between states and improves computational efficiency while maintaining good recognition accuracy [12]. The number of states in the model is a critical factor affecting the recognition performance, and the optimal number of states may vary depending on the size and complexity of the dataset. In this work, the number of states was varied between 3 and 10, and the performance was evaluated for each case.

1. The Hidden Markov Model: Hidden Markov Chain Models represent stochastic processes mathematically. An HMM is represented by the following triple:

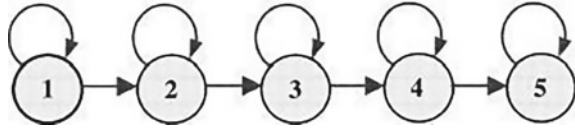
$$\lambda = (A, B, \Pi). \quad (3)$$

- The equation $A = a_{ij}$, defines these variables as an N -by- N transition matrix, where N is the total number of model states. This is a probability density function of a normal distribution with mean μ and standard deviation σ .
- $B = b_{im}$ is an observation matrix with N by M symbols, where the number of observation symbols is M .
- Beginning probability for each state $\Pi_i, i = 1, 2, \dots, N$.

Parameters' Initialization: After segmenting the components of each letter, the researchers initialized the parameters needed to estimate the total number of states. They found that five states were the ideal number to represent each letter, in addition to a predefined number of states. These results were compared to alternative approaches. The researchers referred to the settings required to initialize the Hidden Markov Model with the LRB topology, as described in [5]. The properties of matrix A change with the time (d) associated with each state, as shown in Eq. 4, while Eq. 5 describes the matrix A (Fig. 3).

$$d = \frac{T}{N}, \quad (4)$$

Fig. 3 Left–right banded model with five states



$$A = \begin{pmatrix} a_{ii} & 1 - a_{ii} & 0 & 0 & 0 \\ 0 & a_{ii} & 1 - a_{ii} & 0 & 0 \\ 0 & 0 & a_{ii} & 1 - a_{ii} & 0 \\ 0 & 0 & 0 & a_{ii} & 1 - a_{ii} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

$$a_{ii} = 1 - \frac{1}{d}. \tag{6}$$

Matrix B , which is also important, is the second parameter needed to initialize the Hidden Markov Model after matrix A . Equation 7 is used to calculate the values of matrix B , where each element’s initial value is the same for all possible states.

$$B = \{b_{im}\}, b_{im} = \frac{1}{M}. \tag{7}$$

The initial state distribution vector “ Π ” (Eq. 8) is the third crucial factor in the initialization of a Hidden Markov Model. This vector is in charge of making sure that the system begins in the first state.

$$\Pi = (1 \ 0 \ 0 \ 0 \ 0)^T. \tag{8}$$

2. Model Training and Evaluation

The Baum–Welch algorithm is a popular method for training Hidden Markov Models (HMMs) with discrete observation vectors for each alphabet. This algorithm requires the initial model parameters and a discrete observation vector obtained from the feature extraction stage as inputs. The algorithm then calculates the forward and backward probabilities for each observation, which are used to update the model parameters. This process is repeated iteratively until the parameters converge to an optimal solution. Once trained, the HMM model can provide improved predictions for the model outputs A , B , and. These out puts can be used to accurately recognize and classify gestures.

The Viterbi algorithm is a technique that utilizes dynamic programming to calculate the most probable hidden state path for a given sequence of observations. In the present study, it is employed to determine the most plausible state sequence for each class, which corresponds to a specific alphabet letter. Upon obtaining the most probable state sequence, it is matched with a known path in the database to recognize the letter indicated by the gesture.

It is worth noting that the database used in this study consists of ten recorded frame sequences for each letter, resulting in a total of 260 recorded frame sequences. The reported results show that the HMM-based approach achieved an overall recognition accuracy of 90.76% when 12 codewords were used to represent the orientation feature and an accuracy of 87.08% when 18 codewords were used. The results are comparable to those reported in previous studies using similar approaches.

In summary, this report presents an HMM-based approach for gesture recognition using object (pen's tip) and hand motion trajectory data. The approach involves three stages: feature extraction, preprocessing, and classification. The extracted feature is the orientation of the object (pen's tip) gesture trajectory, which is quantized into discrete vectors or observation symbols using codewords.

A discrete HMM with the Left-Right Banded topology is used for classification, and the models are trained using the Baum-Welch algorithm. The approach achieved high recognition accuracy for recognizing the 26 letters of the alphabet. Figure 4 depicts the architecture of a system designed to detect letters and numbers generated by the movement of any flying object. The system functions as a single unit, comprising four separate components. The first module records and preprocesses the video input stream from the camera. It utilizes techniques such as Gaussian blurring, background subtraction, and thresholding to separate the user's foreground objects and position them accurately in any backdrop environment.

The second module is responsible for motion tracking and preparing the frame with the tracked route. The frame that has been preprocessed is then passed on to the third module, which is a convolutional neural network (CNN). The CNN has been trained to recognize multiple alphabets and numerical symbols using an appropriate and sufficiently sized training set.

The fourth and final modules are responsible for recognizing text. It is carried out by the trained CNN model, which, upon being provided with the preprocessed frames, generates the recognized characters. Through the utilization of cutting-edge computer vision algorithms and a CNN model that has been trained, the system is ultimately able to recognize and classify characters that are written in the air.

Overall, this system's architecture incorporates the latest advancements in computer vision and deep learning to achieve high accuracy in recognizing handwritten characters. It is capable of detecting alphabets and numbers generated by the movement of any flying object, making it a versatile solution with potential applications in diverse areas such as virtual reality, gaming, and gesture recognition.

4 Results and Discussion

The HMM recognition system was implemented using Kevin Murphy's HMM toolbox in MATLAB. The system was tested on a dataset consisting of ten recorded frame sequences for each letter, with nine sequences used for training and one chosen at random for testing. The dataset was aligned to ensure that the paths of each alphabet

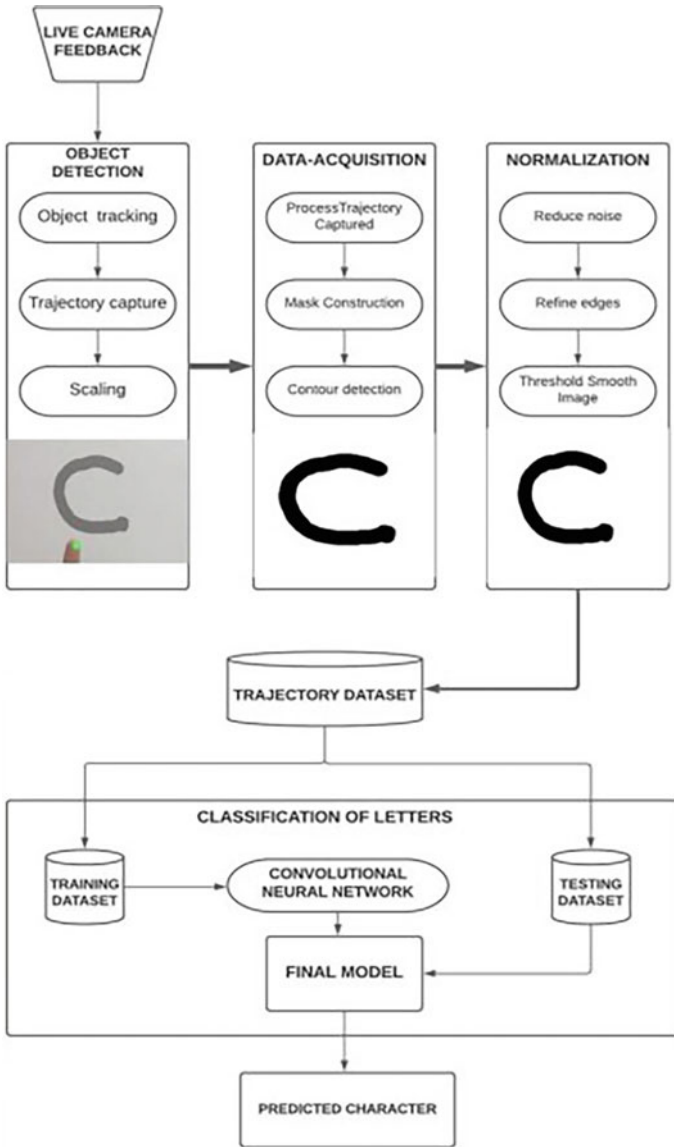


Fig. 4 Overall block diagram of the system, as well as air-writing recognition and handwritten character classification

Table 1 Comparing accuracy with other models

Model	Input 1	Input 2	Input 3
HGR-CNN	74.32	89.23	89.62
HGR-HMM, DTW [9]	96.46	92.00	–
HGR-CNN [13]	90.00	92.00	96.00

were of the same length. The simulation was performed with $M = 8, 12,$ and 18 observation symbols and $3-10$ states and repeated ten times for each number of states. The successful alphabet categorization rate was calculated as the mean of ten simulations for each number of states. The results for a set number of states are presented below, while the results for varying numbers of states per letter are not included as they are trivial in comparison.

The mean value of ten simulations was used to determine the successful categorization rate of the alphabet, where each simulation is corresponded to a varied number of states:

$$\text{Mean } R_c = \left(\frac{\text{Total Correct Alphabets Classification}}{\text{Total Alphabets}} \right). \quad (9)$$

The results of the simulation indicate that the most successful combination of observation symbols and states was achieved when $M = 8$ and 5 states were used, resulting in an overall accuracy rate of 89.62% for alphabet classification. When comparing the recognized state sequences to the known paths from the database, the model with $M = 8$ and 5 states achieved a 74.32% match. However, the model with $M = 18$ observation symbols and 7 states was also quite successful in alphabet classification, scoring 89.23% , but had a lower match rate for state sequences, scoring 65.07% . These results suggest that a simpler model with fewer states and observation symbols may be preferable for this gesture recognition task (Table 1).

4.1 Limitations

Hand gesture recognition using a blue object exhibits certain limitations that should be acknowledged. One such limitation is its potential variability in performance across users, which can be influenced by factors like writing style and hand size. Different individuals may have diverse hand gesture patterns, leading to inconsistencies in recognition accuracy. Environmental conditions also play a role, as inadequate lighting or background noise can impact the system's performance by affecting image quality or interfering with the detection and tracking of the blue object. Occlusion caused by hand or object obstruction can further hinder accurate recognition. Additionally, the complexity of gestures can pose challenges, with more intricate or dynamic gestures being harder to interpret accurately. Training data bias is another consideration, as a lack of diversity in the data used to train the system

may result in reduced accuracy for users outside the demographic represented in the training set. Recognizing these limitations is vital for understanding the practical implications of hand gesture recognition using a blue object, highlighting the need for further research to address these challenges and improve system adaptability, environmental robustness, and accuracy in recognizing complex gestures across diverse user profiles.

5 Conclusion

A method for recognizing alphabet based on object (pen's tip) motion trajectory has been presented. The method consists of three stages: feature extraction, preprocessing, and classification. During the preprocessing stage, the trajectory is recorded, while the feature extraction stage quantifies the direction. In the classification stage, each object (pen's tip) gesture is classified into one of 26 alphabets, and the appropriate alphabet from the database is compared. The results show that the best gesture categorization is achieved with $M = 8$ observation symbols and 5 state models, with an overall classification accuracy rate of 89.62% and a path recognition rate of 74.32. While the proposed system achieved high accuracy and real-time performance, there is still room for improvement. Future work could investigate techniques to improve the accuracy and speed of the system, such as data augmentation, transfer learning, or model compression. The paper focused on recognizing handwritten characters in English. However, there are many other languages and scripts that use different character sets and writing styles. Future work could extend the system to recognize characters in other languages and scripts, such as Chinese, Arabic, or Devanagari.

References

1. Soontranan N, Aramvith S, Chalidabhongse TH (2005) Improved face and tracking for sign language recognition, ITCC 2005, vol 2, pp 141–146, Apr 2005
2. Askar S, Kondratyuk Y, Elazouzi K, Kauff P, Schreer O (2004) Vision-based skin-colour segmentation of moving hands for real-time application, 1st European CVMP, pp 79–85
3. Zhu X, Yang J, Waibel A (2000) Segmenting hands of arbitrary color. In: Proceedings of the fourth IEEE international conference on automatic face and gesture recognition, pp 446–453, Mar 2000
4. Liu N, Lovell BC, Kootsookos PJ, Davis RIA (2004) Model structure selection training algorithms for a HMM gesture recognition system. In: Ninth international workshop on frontiers in handwriting recognition, IWFHR-9 2004
5. Elmezain M, Al-Hamadi A, Krell G, El-Etriby S, Michaelis B (2007) Gesture recognition for alphabets from hand motion trajectory using hidden Markov models. In: 2007 IEEE international symposium on signal processing and information technology, pp 1192–1197
6. Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12(2011):2825–2830

7. Gaurav K, Bhatia PK (2013) Analytical review of preprocessing techniques for offline handwritten character recognition. In: 2nd International conference on emerging trends in engineering management, ICETEM, 2013
8. Elmezain M, Al-Hamadi A, Appenrodt J, Michaelis B (2008) A hidden Markov model-based continuous gesture recognition system for hand motion trajectory. In: 19th International conference on pattern recognition, ICPR 2008
9. Carmona JM, Climent J (2012) A performance evaluation of HMM and DTW for gesture recognition. In: Progress in pattern recognition, image analysis, computer vision, and applications. Lecture notes in computer science, vol 7441, pp 236–243
10. Sudderth EB, Mandel MI, Freeman WT, Willsky AS (2004) Visual hand tracking using nonparametric belief propagation. MIT Laboratory for Information Decision Systems Technical Report P-2603. Presented at IEEE CVPR workshop on generative model-based vision, pp 1–9
11. Moni M, Shawkat Ali ABM (2009) HMM based hand gesture recognition: a review on techniques and approaches. In: 2nd IEEE international conference on computer science and information technology, ICCSIT 2009, Jan 2009
12. Hameed MZ. Initial project report hand gesture recognition for egocentric depth video. Project supervisor: Dr. Tae-Kyun Kim. Department of Electrical Electronic Engineering Imperial College of Science, Technology Medicine
13. Yang Z, Li Y, Chen W, Zheng Y (2012) Dynamic hand gesture recognition using hidden Markov models. In: 2012 7th International conference on computer science education (ICCSE). IEEE, pp 360–365

Grid Search-Optimized Artificial Neural Network for Heterogeneous Cross-Project Defect Prediction



Ruchika Malhotra and Shweta Meena 

Abstract In software engineering, the defect prediction is a crucial aspect. Every software development process aims for the removal of defects in the beginning. Cross-project Defect Prediction (CPDP) is an important research area in software engineering. CPDP is based on transfer learning. Transfer learning is based on transfer of knowledge from one project to another project with similarity in source and target domain to some extent. CPDP aims to identify defects in the target project based on the knowledge transferred from the source project. Due to the limited amount of data, CPDP domain grows rapidly in the upcoming years. However, the selection of suitable target projects is a challenging task in the research. Artificial Neural Network (ANN) is used to design a CPDP model using grid search. Furthermore, the performance of grid search ANN is compared with traditional ANN with default parameter settings. The grid search ANN outperformed the traditional neural network for CPDP. Grid search ANN optimizes hyperparameters through multiple iterations. ANN in combination with optimization algorithms results in an efficient CPDP model. The performance of prediction models is evaluated using AUC metric.

Keywords Artificial Neural Network · Grid search · Cross-project defect prediction · Hyperparameter

1 Introduction

Nowadays, software defect prediction is a key domain for efficient utilization of resources in a reasonable amount of time. The identification of defective modules resulted in the efficient utilization of resources for academics and industrial workplaces. In machine learning (ML), the prediction model holds significant importance. In the past years, the defect prediction model is trained using empirical dataset for the identification of defects in the test data [1].

R. Malhotra · S. Meena (✉)

Department of Software Engineering, Delhi Technological University, Delhi 110042, India
e-mail: shwetameena@dtu.ac.in

The defect prediction models can be developed considering two situations, one is Within-Project Defect Prediction (WPDP), and another is CPDP. Types of defect prediction models differ based on the projects used for training and testing [2]. In the context of WPDP, the prediction model is trained and tested using the same project. However, WPDP can be further used for the identification of defects in the upcoming versions of the same project and helps in enhancing the functionality of upcoming projects. Nowadays, data is exhausted for experiments in the upcoming era. Due to this, researchers are facing issues with the availability of data. The constraint of the limited amount of data availability can be removed by considering different projects for the development of prediction models. CPDP utilizes different projects for model development and testing.

The prediction model designed for CPDP is based on the concept of transfer learning. Transfer learning is based on the transfer of knowledge learned from one task to another task. The knowledge is transferred using different mechanisms on the basis of two different projects characteristics [3]. The source and target domains consider the same project [4] for WPDP. However, CPDP considers different projects for source and target data. Transfer learning is one of the methodologies to conduct CPDP. Further, transfer learning helps in improving the software quality for future projects with similar feature distribution. The source and target projects consist of different features that are termed as heterogeneous transfer learning.

The study is conducted to analyze the effectiveness of ANN for CPDP and grid search optimization with ANN for CPDP. ANN is efficient for learning a compact representation of different behaviors rapidly. They are flexible in managing various behaviors sequentially. One of the advantages of using ANN is to use it without relearning the synaptic weights or strengths. In comparison to other algorithms in machine learning which require relearning, ANN has various advantages for defect identification in various projects.

In the research domain, academics, and industry domain, CPDP plays an important role due to the limited amount of data. The CPDP considered two scenarios such as Homogeneous CPDP (HomCPDP) and Heterogeneous CPDP (HetCPDP). Both scenarios vary depending on the feature types in the training and testing projects. In HomCPDP, the training and target projects consist of similar features. In HetCPDP, the training and testing dataset contains different features with some similarities. Moreover, there must be some relationship between the training and testing datasets when few features consist of different to establish the same ground for knowledge transfer [5]. In case of HetCPDP, the idea of transfer learning has emerged.

The main aim of integrating transfer learning for CPDP is to enhance the quality of software. The software quality depends on various parameters depending on the types of features considered for experimentation. The software quality attributes are categorized into two categories based on functional and non-functional requirements. Further, the quality attributes consider various measures for quality estimation using software metrics. In existing research, the researchers considered traditional ML algorithms for analyzing the efficiency of CPDP models.

The experiment is conducted using AEEEM and NASA datasets in this study. To perform heterogeneous transfer learning, two datasets of AEEEM are used as a

source dataset and eight projects of the NASA repository are used as testing datasets. In this study, ANN is used for CPDP grid-based model development. This study has also identified the grid-based ANN model efficiency with optimization algorithms for CPDP. The grid-based ANN model performed better in comparison to the traditional ANN model for CPDP. The Research Questions (RQs) addressed in this study are: *RQ1*: What is the predictive capability of ANN for CPDP considering AEEEM and NASA projects? *RQ2*: What is the predictive capability of the grid search ANN model for the AEEEM and NASA dataset?

Paper Organization: Sect. 2 has summarized the related work in existing studies. Section 3 has discussed the research methodology used in this study. The results are discussed in detail in Sect. 4. Lastly, the findings are concluded in Sect. 6 with future work.

2 Related Work

In the existing literature, researchers have discussed applications of CPDP. Researchers faced challenges in resolving the issue of selecting features from different projects, that is heterogeneous transfer learning. Cross-version defect prediction performance is also analyzed in comparison to CPDP [6]. The researchers showed that the Peter-15 approach is best for CPDP and conducted experiments with 23 CPDP approaches.

The performance of deep learning neural networks is analyzed for CPDP [7]. Further, the academicians also analyzed the efficiency of two ensemble models (bagging and boosting). In the existing studies [8], CPDP is accomplished using Multi-Kernel Transfer Convolutional Network (MK-TCNN). A novel MK-TCNN approach was designed for CPDP, and it extracts semantic and structural information from the program ASTs [9]. In the existing study, authors also analyzed the characteristics of the defect dataset collected from the NASA repository in terms of its significance, and meaningful for defect prediction. The authors experimented developed 622 CPDP models to resolve the issue of domain distribution among source and target projects [10]. Out of 622 models, only 3.4% of models actually worked considering the feature distribution issue.

The CPDP model is developed using an Adversarial Discriminative Convolutional Neural Network (ADCNN) [11]. The main aim of ADCNN is to establish the common ground for source and target projects. In the existing study, DAECNN-JDP is a novel just-in-time technique proposed by the authors [12]. Autoencoders and CNN are denoised for developing DAECNN-JDP. In the existing study, Bi-LSTM neural network is used to construct CPDP models using AST [13]. The authors experimented with Bi-LSTM and compared with five state-of-the-art CPDP techniques using AUC metric for performance analysis. The authors analyzed the importance of features and instances for CPDP using NASA and PROMISE repository dataset [14]. However,

based on the results obtained, authors concluded that feature is more important than instance for CPDP.

The CPDP model was developed using a ranking-oriented approach termed as ROCPDP and performance was analyzed using Spearman, Kendall, and accuracy [15]. Ensemble learning is also widely used for the development of defect prediction models [16]. The study analyzed that software defect prediction depends on various parameters such as dataset characteristics, classification, clustering, association, and estimation. The most common issue of establishing a relationship between source and target features' distribution is addressed using the optimal transport approach for HetCPDP [5]. The authors developed the EGW+ transport algorithm for HetCPDP. Software quality is improved by developing optimized ACO-based SVM for CPDP [17]. The performance of scientific programming-based ACO-based SVM is better in comparison with KNN. In this study, the ANN is used for developing HetCPDP.

The authors [18] used kernel twin SVM (KTSVM) for resolving the problem of domain adaptation (DA) in various settings. However, authors have used KTSVM with DA for CPDP. Thus, KTSVM parameters have a significant impact on predictive performance. Hence, the parameter optimization has been done by using an improved version of Particle Swarm Optimization (PSO). Moreover, DA-KTSVM outperformed CPDP for defect prediction.

In the existing study [19], the authors proposed a novel approach for CPDP. The CPDP approach proposed by the authors is based on a multi-objective logistic regression model using a genetic algorithm. The multi-objective approach performed better than the single-objective approach used for CPDP. The main aim of a single defect prediction model is achieved using a comparison between precision and recall. The researchers conducted a study [20] to analyze the usefulness of CPDP using three experiments. The experiment was conducted using 34 datasets of ten different open-source projects. It is concluded that CPDP also provides better prediction results and training data must be selected carefully. Moreover, the training data of the same projects does not outperform the experiment conducted using training data from different projects.

The researchers [21] have analyzed the effect of combining transfer learning and machine learning classifier with hyperparameter settings resulting in significant improvement in CPDP performance. In BiLO-CPDP, the CPDP process is automated with comparison to 21 state-of-the-art techniques. However, in the existing studies the importance of automated parameter optimization is analyzed in combination with transfer learning. The authors [22] focused on maintaining a balance between data distributions. The study has been conducted to enhance the reliability and feasibility of the prediction model for conducting CPDP. However, the existing studies are not useful for unit testing since a huge amount of training data is required. A novel transfer learning-based technique has been proposed for increasing product reliability.

3 Research Background

The research background concerning the experiment conducted in this study is further discussed in this section. The features of the dataset used (input and output), dataset characteristics, preprocessing techniques used, optimization techniques used, ANN, and ML classifiers used are summarized in further sections.

3.1 Input and Output Variable

In this study, NASA and AEEEM repository projects are used. The input and output variables of the dataset used in this experiment differ based on the types of metrics. The output variable depicts the presence and absence of defects in a module of specific projects. If the output variable is labeled as clean, it indicates that the module is non-defective. If the output variable is labeled as buggy, it indicates that the module is defective. The AEEEM repository dataset consists of a total of 61 input variables, and the NASA repository consists of varying input variables with a maximum of 39 input variables. NASA [23] dataset consists of Halstead, McCabe, and object-oriented metrics. Table 1 discusses the five groups of the AEEEM dataset [24].

3.2 Dataset Description

The detailed description [15] of AEEEM and NASA [25] dataset project metrics is presented in this section. Table 2 summarizes the AEEEM and NASA datasets used in this study considering the type of project, the programming language of the projects, the total number of files, no. of features, and % of buggy instances. AEEEM and NASA datasets are selected for experimentation in this study.

3.3 Data Preprocessing Techniques

The dataset used in this study is preprocessed using various data preprocessing values. Thus, the techniques used in this study are handling missing values, outliers' removal, null values' handling, and normalization of a dataset is performed.

Table 1 Dataset statistics of the data collected from the AEEEM repository

Repository	Type	Description	# of metrics grouped into categories	
AEEEM	Group 1	ck_oo_cbo, ck_oo_numberOfLinesOfCode	Coupling-based metrics, coupling-based metrics consider a line of code	17
	Group 2	NumberOfBugsFoundUntil, numberOfCriticalBugsFoundUntil	Defect metrics, defect metrics consider based on the security level	05
	Group 3	CvsEntropy, CvsLogEntropy	Change-based entropy metrics, change-based logarithmically entropy metrics	05
	Group 4	LDHH_cbo, LDHH_numberOfLinesOfCode	Decayed entropy (linearly) of CBO and LOC metric	17
	Group 5	WCHU_cbo, WCHU_numberOfLinesOfCode	Coupling metrics (CBO metric and churn of LOC metric)	17

3.4 *Imbalanced Data*

The output class label is majorly corresponding to one output class category in the imbalanced dataset. Model development using an imbalanced dataset results in a biased model toward the majority class. However, the imbalance dataset is handled using SMOTE in this study. SMOTE is a technique, that oversamples the instances of minority class. Furthermore, new instances of minority class are synthesized from existing data only.

3.5 *Optimization Algorithm*

In this study, three optimization algorithms are used such as Greedy Search (GS), Particle Swarm Optimization (PSO), and Evolutionary Search (EVS). GS for ANN architecture starts with a small network and iteratively leads with the selection of neurons and hidden layers till final output. PSO is used as an optimization technique for training ANN. The basic idea is to use PSO to search for the optimal set of weights and biases for the ANN that minimizes the error between the predicted outputs and

Table 2 Dataset statistics of the data collected from AEEEM and NASA repository

Repository	Project	Type	Programming language	Number of files	% of buggy files	# of metrics
AEEEM	Equinox_EQ	OSGi framework	Java	325	36.69	71
	Apache Lucene	Search engine library	Java	691	9.26	71
NASA	CM_1	A NASA spacecraft instrument	C	344	12.21	38
	KC_1	A storage management system for handling ground data	C++	2095	15.51	22
	KC_3	Data program defect dataset	C++	200	18	40
	MC_2	One of the NASA metrics data program defect datasets	C	125	35.2	40
	MW_1	Zero-gravity experiment	C	263	10.27	38
	PC_1	Earth-orbiting satellite (flight software)	C	735	8.3	38
	PC_2	NASA for earth object	C	1493	1.07	37
	PC_3	NASA orbiting project	C	1099	12.56	38

the actual outputs. EVS is used to find out the optimal set of weights and biases to minimize the error between predicted and actual outputs.

3.6 Machine Learning Techniques

In this study, traditional ANN with default parameter settings and ANN with optimization algorithms such as GS, PSO, and EV under varying settings of ANN parameters were used for the development of prediction models.

4 Results

In this section, the answers to research questions are discussed in detail. The experiment conducted in this experiment used two projects of AEEEM and eight projects of NASA. The projects used for experimentation consist of different features such as Halstead metrics and object-oriented metrics. To perform HetCPDP using transfer learning, the setting of selected projects has been fixed. AEEEM projects are used for training of prediction model and NASA projects are used for testing a developed prediction model. Further, 16 pairs are formed for experimentation.

In this study, 112 models were designed considering possible pairs of source and target datasets. The performance of each pair is analyzed according to the selected classifier with default and grid search parameter settings. However, the performance of the prediction models was analyzed using AUC metric. AUC provides more generalized and unbiased results in the case of an imbalanced dataset. This section interprets the results and answers to the RQs discussed in Sect. 1.

RQ1: What is the predictive capability of ANN for CPDP considering AEEEM and NASA projects?

In this study, CPDP models are developed using various projects with different features. CPDP is developed using ANN with default parameter settings. In default parameter settings, `batch_size = 100`, `learning_rate (lr) = 0.3`, `momentum (mome) = 0.2`, `epochs (#) = 500`. The traditional ANN performance does perform well with default parameter settings. During experimentation, we have to see the features of both the training and testing datasets which must have the same vector. If the feature vector space is not matching, then the transfer learning concept is not applicable. Since both source and target projects have similar features considering object-oriented metrics. Thus, the challenging task was to extract relevant features from both projects. Moreover, training and testing projects consist of different features. The results obtained using ANN for CPDP are summarized in Table 3. Based on covariance among the features of both projects, specified feature pairs were selected for designing the prediction model. Further, the prediction model is used for defect prediction in upcoming projects with similar characteristics. ANN-based prediction model can be improved using grid search optimization.

RQ2: What is the predictive capability of the grid search ANN model for AEEEM and NASA dataset?

To answer this RQ, the default parameters of ANN were changed using a grid search approach. In the grid search approach, the model has been trained multiple times for varying sets of parameter values. In the default parameter setting, the learning rate was 0.3, which is used for training large and complex datasets. However, the dataset is not much larger in this study. Thus, the experiment has been conducted for varying sets of learning values, which provides the best performance of the ANN model at a learning rate of 0.01. In optimized ANN, the no. of hidden layers is updated with number of neurons in each hidden layer. In ANN, we have used the logistic

Table 3 AUC values for ANN and grid search ANN model for CPDP

Source dataset → target dataset	ANN	ANN + PSO	ANN + EVS	ANN + GS
EQ → CM1	0.5	0.79	0.74	0.79
EQ → KC1	0.57	0.74	0.7	0.73
EQ → KC3	0.65	0.64	0.65	0.64
EQ → MC2	0.65	0.78	0.76	0.79
EQ → MW1	0.47	0.72	0.7	0.69
EQ → PC1	0.54	0.69	0.65	0.66
EQ → PC2	0.53	0.65	0.63	0.67
EQ → PC3	0.59	0.73	0.72	0.75
Lucene → CM1	0.49	0.72	0.7	0.78
Lucene → KC1	0.53	0.68	0.72	0.7
Lucene → KC3	0.57	0.7	0.67	0.7
Lucene → MC2	0.69	0.65	0.69	0.67
Lucene → MW1	0.5	0.69	0.67	0.69
Lucene → PC1	0.2	0.68	0.67	0.64
Lucene → PC2	0.3	0.78	0.72	0.79
Lucene → PC3	0.5	0.71	0.7	0.74

activation function for experimentation. The performance of the grid search ANN model for CPDP was 0.79. In some of the pairs, it was 0.3 also depending on the size of the training and testing datasets. ANN ensures that the size of the training and testing dataset is the same. Furthermore, statistical analysis has been done to analyze the performance of techniques statistically. Thus, Friedman test is used for validation of results. The null hypothesis states that there is no significant difference in the performance of the four techniques. The alternate hypothesis says that there is a significant difference in the performance of the four techniques. Thus, based on the Friedman test result at a 0.05 significance level, the computed Chi-square value is 16.016. The mean rank of four techniques is represented in Table 4. Hence, an alternate hypothesis is accepted such that there is statistical difference among the performance of the four techniques. The next step is to analyze the performance of each technique individually using post-ad hoc analysis such as Wilcoxon signed-rank test. Furthermore, six combinational pairs are formed for Wilcoxon signed-rank test.

Table 4 Mean rank of CPDP models

Model	Mean rank
ANN	1.38
ANN_PSO	2.92
ANN_EVS	2.46
ANN_GS	3.23

The hypothesis for the further comparison among six pairs is as follows: Null Hypothesis (H_{o1}): There is no significant difference among the performances of ANN_PSO and ANN. Alternate Hypothesis (H_{a1}): There is a significant difference among the performances of ANN_PSO and ANN. Thus, the Asymp. Sig. (two-tailed) is 0.003 at significance level 0.05. Hence, H_{a1} is accepted. H_{o2} : There is no significant difference among the performances of ANN_EVS and ANN. H_{a2} : There is a significant difference among the performances of ANN_EVS and ANN. Thus, the Asymp. Sig. (two-tailed) is 0.003 at significance level 0.05. Hence, H_{a2} is accepted. H_{o3} : There is no significant difference among the performances of ANN_GS and ANN. H_{a3} : There is a significant difference among the performances of ANN_GS and ANN. Thus, the Asymp. Sig. (two-tailed) is 0.003 at significance level 0.05. Hence, H_{a3} is accepted. H_{o4} : There is no significant difference among the performances of ANN_EVS and ANN_PSO. H_{a4} : There is a significant difference among the performances of ANN_EVS and ANN_PSO. Thus, the Asymp. Sig. (two-tailed) is 0.105 at significance level 0.05. Hence, H_{o4} is accepted. H_{o5} : There is no significant difference among the performances of ANN_GS and ANN_PSO. H_{a5} : There is a significant difference among the performances of ANN_GS and ANN_PSO. Thus, the Asymp. Sig. (two-tailed) is 0.472 at significance level 0.05. Hence, H_{o5} is accepted. H_{o6} : There is no significant difference among the performances of ANN_GS and ANN_EVS. H_{a6} : There is a significant difference among the performances of ANN_GS and ANN_EVS. Thus, the Asymp. Sig. (two-tailed) is 0.027 at significance level 0.05. Hence, H_{o6} is accepted. However, it has been observed that ANN + GS outperformed.

5 Limitations

In this section, the limitations of the experiment in this study are discussed. Threats to construct validity are based on the equation between theory and experiment. The AUC performance metric used in this paper is widely used in the existing studies and literature. Threats to construct validity are based on used metrics and defective datasets. In this study, the experimentation has been conducted using five AEEEM group datasets and a NASA dataset. Threats to conclusion validity are based on the treatment and outcome variable. The AUC performance metric is used for performance analysis. In this study, we have validated the result statistically using Friedman and Wilcoxon signed-rank test. Threats to internal validity refer to the factors that could influence our results. Thus, GA is repeated several times. However, in this study, we have used a limited dataset for experimentation, but this study can be replicated for various projects with different metrics as independent variables.

6 Conclusion

The main aim of conducting experiment is to analyze the efficiency and evaluate the performance of ANN with the ANN grid search model for CPDP. HetCPDP is developed in this study considering projects from different repositories. In the source dataset, the AEEM dataset was specified, while in the target dataset, NASA datasets were specified. The number of features differs in both, and the CPDP model is designed considering similarity among features to some extent. The experiment performed in this study showed that traditional ANN performance was not much significant. Grid-based ANN performed better than traditional ANN considering the parameters' setting using a grid-based approach. Grid search ANN model worked better by considering more number of neurons and hidden layers for CPDP. Grid search ANN model outperformed ANN based on # of neurons in the hidden layer, # of hidden layers, and lr value. Grid search provides an effective combination of hyperparameters. Moreover, grid search is expensive, especially for large networks with many hyperparameters. In future work, random search or Bayesian optimization may be used as an alternative to grid search with deep learning. The grid search ANN mode can be tested further with more optimization algorithms for HetCPDP and HomCPDP.

References

1. Tong H, Wei L, Weiwei X, Wang S (2023) ARRAY: adaptive triple feature-weighted transfer Naive Bayes for cross-project defect prediction. *J Syst Softw* 202:111721. <https://doi.org/10.1016/j.jss.2023.111721>
2. How far does the predictive decision impact the software project? The cost, service time, and failure analysis from a cross-project defect prediction model. *J Syst Softw* 195:111522. <https://doi.org/10.1016/j.jss.2022.111522>
3. Seyedrebar H, Burak T, Gunarathna D (2017) A systematic literature review and meta-analysis on cross project defect prediction. *IEEE Trans Softw Eng* 45:111–147
4. Liu Y, Khoshgoftaar TM, Seliya N (2010) Evolutionary optimization of software quality modeling with multiple repositories. *IEEE Trans Softw Eng* 36:852–864. <https://doi.org/10.1109/TSE.2010.51>
5. Zong X, Li G, Zheng S, Zou H, Yu H, Gao S (2023) Heterogeneous cross-project defect prediction via optimal transport. *IEEE Access* 11:12015–12030. <https://doi.org/10.1109/ACCESS.2023.3241924>
6. Amasaki S (2020) Cross-version defect prediction: use historical data, cross-project data, or both? *Empir Softw Eng* 25:1573–1595. <https://doi.org/10.1007/s10664-019-09777-8>
7. Elish MO, Elish K (2021) An empirical comparison of resampling ensemble methods of deep learning neural networks for cross-project software defect prediction. *Int J Intell Eng Syst* 14:201–209. <https://doi.org/10.22266/ijies2021.0630.18>
8. Deng J, Lu L, Qiu S, Ou Y (2020) A suitable AST node granularity and multi-kernel transfer convolutional neural network for cross-project defect prediction. *IEEE Access*. 8:66647–66661. <https://doi.org/10.1109/ACCESS.2020.2985780>
9. Cao H, Bernard S, Heutte L, Sabourin R (2018) Improve the performance of transfer learning without fine-tuning using dissimilarity-based multi-view learning for breast cancer histology images. In: *Lecture notes in computer science (including subseries Lecture notes in artificial*

- intelligence and Lecture notes in bioinformatics), 10882 LNCS, pp 779–787. https://doi.org/10.1007/978-3-319-93000-8_88
10. Zimmermann T, Nagappan N, Gall H, Giger E, Murphy B (2009) Cross-project defect prediction 91. <https://doi.org/10.1145/1595696.1595713>
 11. Sheng L, Lu L, Lin J (2020) An adversarial discriminative convolutional neural network for cross-project defect prediction. *IEEE Access* 8:55241–55253. <https://doi.org/10.1109/ACCESS.2020.2981869>
 12. Zhu K, Zhang N, Ying S, Zhu D (2020) Within-project and cross-project just-in-time defect prediction based on denoising autoencoder and convolutional neural network. *IET Softw* 14:185–195. <https://doi.org/10.1049/iet-sen.2019.0278>
 13. Li H, Li X, Chen X, Xie X, Mu Y, Feng Z (2019) Cross-project defect prediction via ASTToken2Vec and BLSTM-based neural network. In: *Proceedings of the international joint conference on neural networks*, pp 1–8, July 2019. <https://doi.org/10.1109/IJCNN.2019.8852135>
 14. Yu Q, Jiang S, Qian J (2016) Which is more important for cross-project defect prediction: Instance or feature? In: *Proceedings—2016 International conference on software analysis, testing and evolution, SATE 2016*, pp 90–95. <https://doi.org/10.1109/SATE.2016.22>
 15. You G, Wang F, Ma Y (2016) An empirical study of ranking-oriented cross-project software defect prediction. *Int J Softw Eng Knowl Eng* 26:1511–1538. <https://doi.org/10.1142/S0218194016400155>
 16. Sharma T, Jatain A, Bhaskar S, Pabreja K (2023) Ensemble machine learning paradigms in software defect prediction. *Procedia Comput Sci* 218:199–209. <https://doi.org/10.1016/j.procs.2023.01.002>
 17. Shafiq M, Alghamedy FH, Jamal N, Kamal T, Daradkeh YI, Shabaz M (2023) Scientific programming using optimized machine learning techniques for software fault prediction to improve software quality. *IET Softw*. <https://doi.org/10.1049/sfw2.12091>
 18. Jin C (2021) Cross-project software defect prediction based on domain adaptation learning and optimization. *Expert Syst Appl* 171:114637. <https://doi.org/10.1016/j.eswa.2021.114637>
 19. Canfora G, De Lucia A, Di Penta M, Oliveto R, Panichella A, Panichella S (2013) Multi-objective cross-project defect prediction. *Proceedings—IEEE 6th international conference on software testing, verification and validation, ICST 2013*, pp 252–261. <https://doi.org/10.1109/ICST.2013.38>
 20. He Z, Shu F, Yang Y, Li M, Wang Q (2012) An investigation on the feasibility of cross-project defect prediction. *Autom Softw Eng* 19:167–199. <https://doi.org/10.1007/s10515-011-0090-3>
 21. Li K, Xiang Z, Chen T, Tan KC (2020) BiLO-CPDP: bi-level programming for automated model discovery in cross-project defect prediction. In: *Proceedings—2020 35th IEEE/ACM international conference on automated software engineering, ASE 2020*, pp 573–584. <https://doi.org/10.1145/3324884.3416617>
 22. Ryu D, Baik J (2016) Effective multi-objective naïve Bayes learning for cross-project defect prediction. *Appl Soft Comput J* 49:1062–1077. <https://doi.org/10.1016/j.asoc.2016.04.009>
 23. Croft R, Babar MA, Kholoosi M (2023) Data quality for software vulnerability datasets
 24. D'Ambros M, Lanza M, Robbes R (2003) An extensive comparison of bug prediction approaches. In: *Proceedings of the international conference on mining software repositories*, pp 31–41
 25. Giray G, Bennin KE, Köksal Ö, Babur Ö, Tekinerdogan B (2023) On the use of deep learning in software defect prediction. *J Syst Softw* 195:111537. <https://doi.org/10.1016/j.jss.2022.111537>

Design of Smart Weed Detection and Evacuation Robot Using TensorFlow Model Maker



P. Jothilakshmi, C. Gomatheeswari Preethika, and R. Mohanasundaram

Abstract Agriculture was the foundation of civilization, making its importance more apparent. Yet, agriculture is actually a vital industry in every nation on earth. Crops and weeds compete for the same nutrients, water, sunlight, and space. The proposed work intends to eliminate the weeds by sprinkling herbicide in the field while causing no harm to the plant. The automated robot includes a Raspberry Pi for object identification, a USB webcam for object detection, a charging circuit, a sprinkler setup for spraying the herbicide, a movement setup using L298N Motor Driver and DC motors coupled with wheels. The robot is free to travel around the field, while the webcam catches the real environment. The Model Maker algorithm is applied for object detection, which is followed by the detection of the weed and obstacles that exist in the field. The sprinkler arrangement is then instructed to just spray the weed from that point on. The experimental results illustrate that a miniaturized, functionally designed weed control bot is capable of detecting the weeds under various topographical situations and crop growth phases.

Keywords Agriculture · Automation · Objection identification · Object detection

1 Introduction

By 2050, the 7.7 billion people who currently inhabit the planet are predicted to number over 9 billion. The global food supply will need to be boosted by 70–100% to feed this population. Crop production is hampered by a number of biotic and abiotic factors, as well as socioeconomic and crop management-related problems. Weeds are the most significant biotic restrictions to agricultural production in both developing and wealthy countries. In general, diseases (fungi, bacteria, etc.) and

P. Jothilakshmi (✉) · C. Gomatheeswari Preethika · R. Mohanasundaram
Sri Venkateswara College of Engineering, Sripierumbudur, Tamilnadu, India
e-mail: jothi@svce.ac.in

C. Gomatheeswari Preethika
e-mail: cgomathi@svce.ac.in

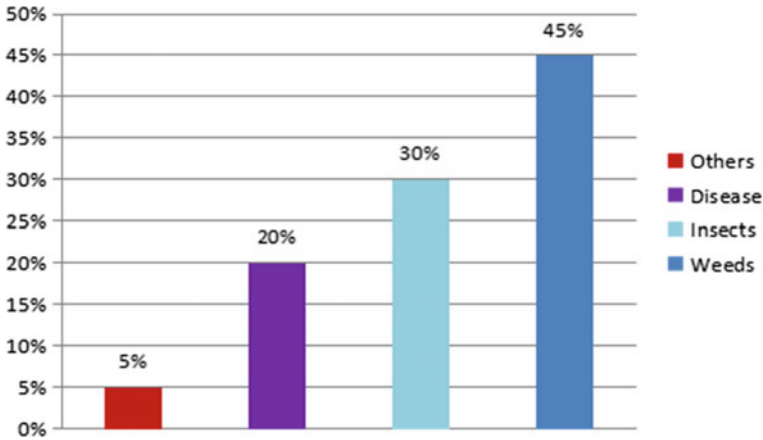


Fig. 1 Annual agricultural losses based on data gathered by Tamil Nadu Agricultural University. Source: http://www.agritech.tnau.ac.in/agriculture/agri_weedmgt_aboutweed.html

animal pests (insects, rodents, nematodes, mites, birds, etc.) are of less concern, and weeds account for the biggest potential loss. A nation's well-being depends so heavily on a healthy agricultural sector. Scientists and farmers have been working to find ways to improve crop output with less negative environmental impact.

Unwanted plants known as weeds proliferate alongside crops and plants being raised. Any farmed field has to deal with these plants as an unwanted guest. Indeed, these weeds prevent the growth of vegetative crops by absorbing the water intended for their growth. According to figures gathered by the Tamil Nadu Agricultural University, weeds are responsible for 45% of losses. A graphic depicts the data on annual agricultural production losses [1]. Therefore, weed removal is essential throughout the entire agricultural process in order to prevent losses and thereby meet the rising demand for food (Fig. 1).

Automation in agriculture is an upcoming challenge throughout the world [2]. In latest time, artificial intelligence has been seeing loads of direct software in farming. The advances in laptop vision, mechatronics (mechanical and electronics), synthetic intelligence, and device studying are allowing the improvement and deployment of far-flung sensing technology to perceive and control plants, weeds, pests, and diseases. Cognitive computing has come to be the maximum disruptive era in agricultural offerings and it can learn, understand, and engage with distinctive environments to maximize productivity. In order to identify weeds, smart weeding machines depend on the effectiveness of the machine vision system [3]. However, environmental uncertainties, like as lighting conditions and color variations in soil or leaves, have an impact on how well the machine vision system performs, putting a cap on the precision of weed management. The main contribution of this proposed work is as follows:

- To get surplus production with decreased human work.

- To optimize the available resources through weed management.
- To automate the process.
- To measure accurately the amount of fertilizers needed for the farms, therefore minimal usage of fertilizers.

The paper is organized into three main sections. Section 2 describes about the literature survey followed by the proposed method in Sect. 3. Section 4 explains about the implementation and results and finally conclusion in Sect. 5.

2 Related Work

Since the advent of artificial intelligence, there have been various attempts in improvising multiple parts of agriculture. A comprehensive review of the same [4] had given an idea to pursue weed control using artificial intelligence. From [5, 6], it has been observed that image processing can be very useful in weed control robots. Further studies on different weeds and weed control [7, 8], weed science and its practices [9, 10] had provided understanding on weed control. But from [6], it has been observed that different weeds require different detection algorithms which make the process vague.

Although weed identification using computer vision techniques has been largely successful, deep learning models like convolution neural networks (CNNs) have recently been the models of choice for computer vision jobs. Weed detection is a unique instance of classifying plant species. Models based on deep learning have the advantage of eliminating the need for feature selection and segmentation since they incorporate the extraction of features and their mapping to output results into the network. In [11], a semantic segmentation model called SegNet that is built on a fully convolutional network segmented weeds and rice seedlings with an accuracy of 92.7%. In order to detect common weeds in cotton and tomato plants with over 99% accuracy, a transfer learning strategy that takes advantage of pre-trained CNN models combined with support vector machines was utilized in [12].

A thorough analysis of a fully convolutional neural network [13] was enhanced by the suggestion of using a single model with distinct layers for various crops, which simplifies and clarifies the identification process. References [11, 14, 15] presented a number of techniques for designing hardware that is appropriate for weed control robots. The advancements in sensor-based mechanical weeding since the 1980s are examined in this paper [16]. It is concentrated on scientific studies that presented data from their research in order to give an overview of the potential uses for sensor-based systems and to demonstrate their effectiveness. Consideration and discussion are given to the practical use for existing and upcoming farms. Additionally, a future prognosis for sensor-based mechanical weeding is provided, along with suggestions for enhancements.

This paper [17] studies the three critical phases of agriculture: cultivation, monitoring, and harvesting, while taking into account the degree of AI involved and the

robots used. This study includes a thorough analysis of more than 150 studies that focus on the use of automation in agriculture that has been done between 1960 and 2021. It draws attention to the unmet research needs for developing intelligent autonomous agricultural systems. The various biotechnological methods to develop the crop plants resistant to herbicides were investigated [18]. The development of weed detection systems and weed control techniques are the main topics of this chapter [1]. The difficulties of robotic weed management, concentrating on perception systems that can distinguish between crop and weed plants and weed control procedures that include both chemical and mechanical weed control, were discussed. An automated weeding system case study was offered.

3 The Proposed System

3.1 Methodology

The suggested approach entails creating the hardware modules needed to capture the image and, if required, applying herbicides. To distinguish between a crop and a weed in a captured image, a software model is to be created. Image processing can be highly helpful for weed control robots. As a result, weed photos are taken and trained. From the literature, it is discovered that convolution neural networks aids in training the dataset in robust manner. So, the plant and weed datasets are pre-trained to develop the software model. The proposed system is made up of the following fundamental structures:

- i. Hardware design.
- ii. Object detection.
- iii. Dataset labeling.

3.2 Hardware Design

The block diagram for the proposed work is shown in Fig. 2. The Raspberry Pi acts as the brain of the architecture, and hence, the size of the bot is to be reduced when compared the mechanical robots used in literature [2, 5, 6]. It is in contact with all of the peripherals that are connected to it. It gathers data from additional peripherals and generates the required outputs. Finding crops and weeds in a field requires the use of a camera. Based on data from the camera, Pi directs a spray servo and relay to spray the weed with herbicide. A detect servo is used to rotate the camera attached on it 180° in order to detect the weed. In order to move the robot, a pair of DC motors and an ultrasonic sensor are used. The DC motor is driven by a motor driver module.

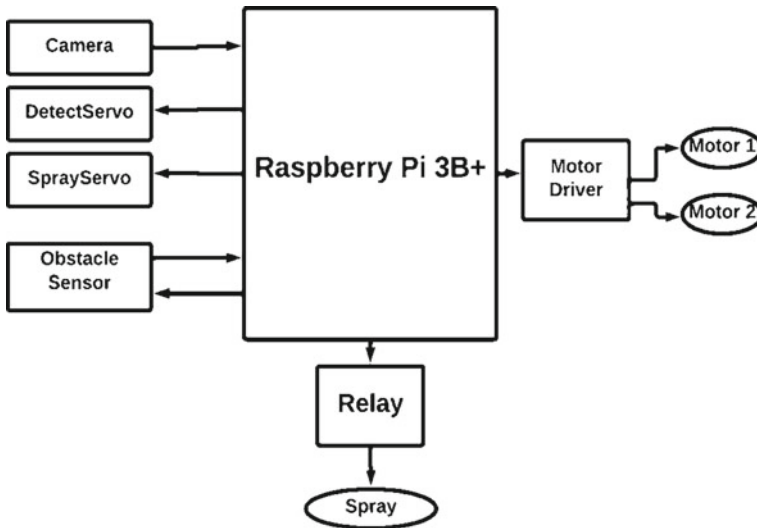


Fig. 2 Block diagram of the proposed system

3.2.1 Automated Structure

The proposed automated system is composed of a motor driver, DC motor, ultrasonic sensor, USB web camera for indication, motor driver, servo motor, power supply unit, ultrasonic sensor, and RF trans-receiver for controlling part, in contrast to mechanical weeding machines that have complex structures [17].

- (a) The Raspberry Pi 3 Model B + includes 1.4 GHz equipped with a quad-core Cortex A53 processor and 1 GB of RAM.
- (b) The image captured using webcam has image quality of RGB 24 or I420 with hardware resolution of 500,000 pixels and frame rate of 30 fps. To rotate the camera, servo motor is used.
- (c) The relay module is a switch which is generally used as an automatic control circuit. In the proposed model, two-channel relay module is used to operate the sprayer.
- (d) An ultrasonic sensor and DC motor make the robot to move freely in field.
- (e) The suggested method uses L298N motor driver module which can control up to four DC motors or two DC motors with direction and speed control. It has a L298N motor driver IC and a voltage regulator (78M05) to output 5 V which is used to power the internal circuitry.
- (f) A 5 V power bank for powering the Raspberry Pi and 12 V LIPO battery is used for charging the other peripherals.
- (g) Raspberry Pi knows the angles where the weed is present from the detection module.

- (h) The Raspberry model detects the weed positions and it operates the servo motor to stop in those positions and switch on the relay module. The relay module switches on the spray mounted on top of the servo motor and sprays the herbicide.

3.3 Object Detection

The term “object detection” describes the process of locating objects using video or picture inputs. Modern state-of-the-art architecture for image classification is convolution neural networks. Three elements make up the foundation of the CNN architectural design: convolution, pooling, and fully connected layers. Together, these elements enable the learning of a high-density feature representation of the input. One-level and two-level object recognition models are the two different categories of object recognition models. The EfficientDet model, which is categorized as a one-level object detection, is part of this research. This model has good scalability and efficiency and is robust and complex. Each model’s architecture consists of parts for the head, neck, and spine. Stochastic Gradient Descent algorithm has been used to train all iterations of EfficientDet. It has been done using a momentum of 0.9 and a weight decay of $4e^{-5}$. In the first training period, the learning rate is linearly increased from 0 to 0.16 and then annealed down using the cosine decay method.

3.3.1 Bounding Boxes

To recognize an object, it must be localized in order to obtain information about the object in the image. By encoding the model’s output into a bounding box structure, this task is accomplished. The bounding box refers to the imaginary rectangle that encloses the object. The coordinates of the image are used to specify the precise implementation. Using the box’s center point along with its width and height is one method of defining a bounding box. It can also specify the x and y coordinates’ minimum and maximum values. An effective method for localization is to encode it in the bounding box, which may be represented and showed as a regression problem.

3.3.2 Dataset

The dataset used for the proposed work is sesame dataset obtained from Kaggle. It consists of 546 images. As the images are not sufficient in number, data augmentation technique is used to increase the size of the dataset to 1300. Each image is a 512×512 color image. The sample crop and weed images are shown in Figs. 3 and 4.



Fig. 3 Sample crop (sesame) images



Fig. 4 Sample weed images in sesame field

3.3.3 Test–Train Data

The process of making a deep learning model begins with the test–train split. It is a method used for evaluating the performance of the deep learning model. The steps involve splitting the dataset into two subsets, namely the train set and the test set.

a. Train data

The learning in the training set gains the experience that the algorithm uses to learn. For each input variable, there exists an observed output variable in supervised learning. It is used to fit the deep learning model.

b. Test data

This subset is used to test the model like input data which is provided to the model and the predictions are made against the true value. It only decides the performance of the model.



Fig. 5 Labeling images in bounding box using LabelImg

3.4 Dataset Labeling

To label the images in the dataset, labelImg, an open-source graphical image annotation tool is used. Annotations are saved as an XML file in PASCAL VOC format. The tool also supports the YOLO and CreateML formats. Figure 5 shows the labeling of a plant in field by creating a bounding box around it.

4 Implementation and Results

The implementation of hardware and software is discussed in this section. The working methodology involving the hardware modules of the proposed work is given in the form of flowchart in Fig. 6.

- i. Detect servo (which mounts the camera and the ultrasonic sensor) rotates from 0° to 180° to detect if there is a weed in each angle and returns back to the middle focus (90° position).
- ii. The spray servo which mounts the herbicide spray goes to the angles at which the weed is detected and sprays the herbicide.
- iii. After this process, the ultrasonic sensor checks the distance of the obstacle.

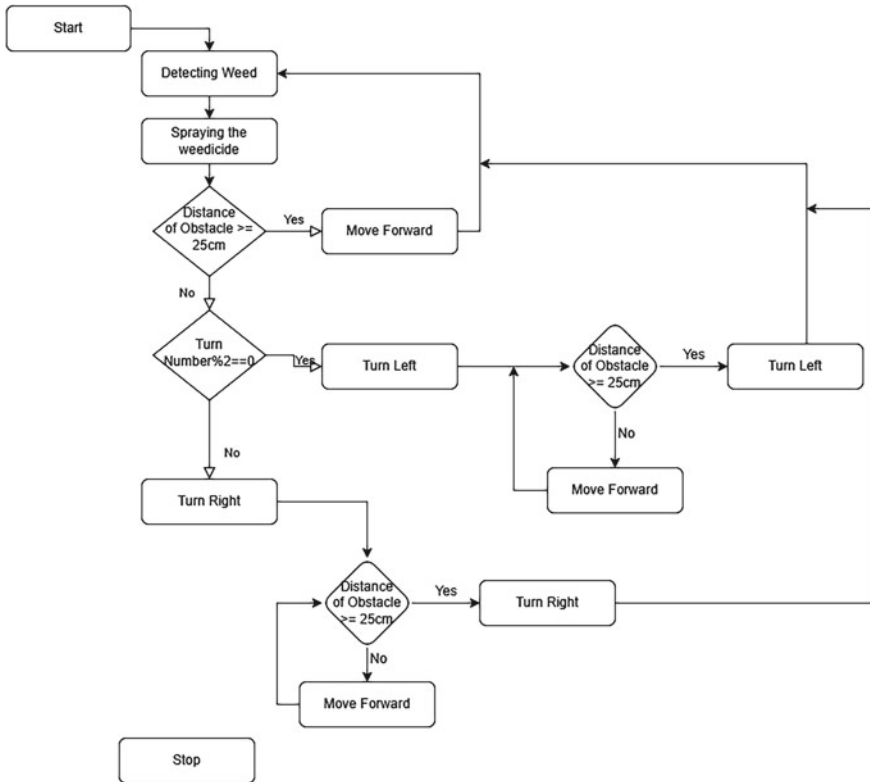


Fig. 6 Flowchart describing working of robot

- iv. If the distance is greater than 25 cm, the robot moves forward for a certain time, stops and repeats the above steps.
- v. Else the robot keeps track of the turn number and turns right for every odd number instance of turn and left for every even number instance of turn (hereafter noted as the X direction).
- vi. After turning, it checks if there is an obstacle in the X direction.
- vii. If the distance of the obstacle in the X direction is < 25 cm, then move forward for a certain time and repeat the above steps, otherwise turn to the X direction.

The hardware working model (top and front view) is shown in Fig. 7. The software’s used to develop the robot are Colab notebooks, LabelImg for labeling an image and object detection with TFLite Model Maker. In machine learning, Colab is widely used for creating and refining neural networks. The TensorFlow Lite (TFLite) Model Maker Library is a high-level library that makes it easier to train TensorFlow Lite models with unique datasets. Transfer learning techniques shorten training periods and use less training data. This work makes use of the TFLite Model Maker.

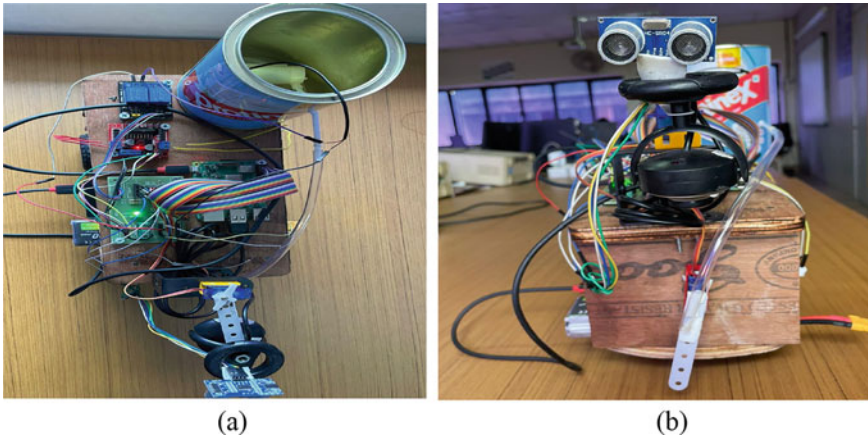


Fig. 7 Working model. **a** Top view, **b** Front view

The TEfficientDet-Lite0 model architecture is applied to detect the weeds and crops in the proposed model. Pascal VOC format is used to label the dataset. The `object_detector.DataLoader.from_pascal_voc` is used to load the data. After loading the data, the TensorFlow model can be trained using the `object_detector.create` method. The create method is the driver function that the Model Maker library uses to create models. Figure 8 shows the screenshot of result obtained in TFLite Model Maker indicating the accuracy as 84% for crop and 81% for weeds. The challenges involved in the proposed system are to identify the image database which is to distinguish the weed and crop and also the farmers may lack knowledge about the state-of-the-art facilities. Also, the herbicides used for weed management are crop-specific and its identification is highly important. Table 1 illustrates the comparative study of accuracy obtained in different fields using convolution neural networks.

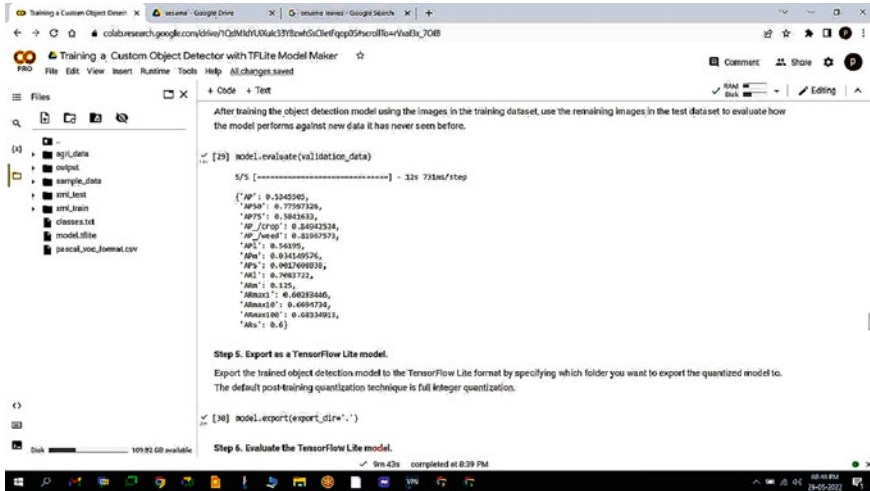


Fig. 8 Accuracy of crop and weed detection using TFLite Model Maker

Table 1 Comparison of proposed work with literature

References	Crop dataset	Accuracy (%)	Software used
Dyrmann et al. [13]	Wheat	46	DetectNet by Nvidia
Hussain et al. [19]	Okra, bitter gourd, and sponge gourd	88	VigoEngraverL
Proposed work	Sesame	84	TFLite Model Maker

5 Conclusion

The robot was developed and designed effectively. Each module has been tested and operates as intended. These modules were fully automated and integrated. Additionally, the final model was examined for functionality under all circumstances. The results obtained an accuracy of 84% for crop and 81% for weeds in sesame crop but can also be employed to different agricultural crops. It is implied that a robot can be created using a Raspberry Pi that can detect weeds and spray herbicide, assisting farmers in performing their agricultural tasks efficiently. Because the bot designed is modest in size, even a fleet of bots can be employed to control the weeds. This model makes use of less hardware while providing more accuracy in terms of detecting the weed and crop.

References

1. Steward B, Gai J, Tang L (2019) The use of agricultural robots in weed management and control. In: Billingsley J (ed) *Robotics and automation for improving agriculture*. Volume 44 of Burleigh Dodds Series in Agricultural Science Series. Burleigh Dodds Science Publishing, Cambridge
2. Perez-Ruiz M, Upadhyaya SK (2012) GNSS in precision agricultural operations. In: *Intech Open Book New Approach of Indoor and Outdoor Localization Systems*
3. Lamm RD (2000) *Robotic weed control for cotton*. Ph.D. dissertation, Department of Biological and Agricultural Engineering, University of California, Davis, p 116
4. Jha K, Doshi A, Patel P, Shah M (2019) A comprehensive review on automation in agriculture using artificial intelligence. *Artif Intell Agric* 2:1–12
5. Wu X, Aravecchia S, Lottes P, Stachniss C, Pradalier C (2020) Robotic weed control using automated weed and crop classification. *J Field Robot* 37(2):322–340
6. Wang A, Zhang W, Wei X (2019) A review on weed detection using ground-based machine vision and image processing techniques. *Comput Electron Agric* 158:226–240
7. Schweizer EE, May MJ (1993) *Weeds and weed control. The sugar beet crop*. Springer, Dordrecht, pp 485–519
8. Ampong-Nyarko K, De Datta SK (1991) *A handbook for weed control in rice*. International Rich Research Institute, Philippines
9. Das TK (2019) *Weed science basics and applications*. Jain Brothers, New Delhi
10. Monaco TJ, Weller SC, Ashton FM (2002) *Weed science: principles and practices*. Wiley-Blackwell, 4th Edition
11. Espejo Garcia B, Mylonas N, Athanasakos L, Fountas S, Vasilakoglou I (2020) Toward weeds identification assistance through transfer learning. *Comput Electron Agric* 171:105306
12. Bak T, Jakobsen H (2004) Agricultural robotic platform with four wheel steering for weed detection. *Biosystem Eng* 87(2):125–136
13. Dyrmann M, Jorgensen RN, Midtiby Robo HS (2017) Weed support—detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. Cambridge University Press, Cambridge
14. Slaughter DC, Giles DK, Downey D (2008) Autonomous robotic weed control systems: a review. *Comput Electron Agric* 61(1):63–78
15. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(12):24812495. <https://doi.org/10.1109/TPAMI.2016.2644615>
16. Yaduraju NT (2006) Herbicide resistant crops in weed management. In: *The extended summaries, golden jubilee national symposium on conservation agriculture and environment*. Banaras Hindu University, Varanasi, pp 297–298
17. Machleb J, Peteinatos GG, Kollenda BL, Andújar D, Gerhards R (2020) Sensor-based mechanical weed control: present state and prospects. *Comput Electron Agric* 176:105638
18. Wakchaure M, Patle BK, Mahindrakar AK (2023) Application of AI techniques and robotics in agriculture: a review. *Artif Intell Life Sci* 2023:100057
19. Hussain A, Fatima HS, Zia SM, Hasan S, Khurram M, Stricker D, Afzal MZ (2023) Development of cost-effective and easily replicable robust weeding machine—premiering precision agriculture in Pakistan. *Machines* 11(2):287

Document Store Schema Design Alternatives and Their Impact



Monika Shah and Amit Kothari

Abstract Smart apps in the twenty-first century can use any data from real-world objects properties, behaviours, and events. Thus, the amount and variety of data, from structured to unstructured, are growing. The integration of data from many sources produces a wide range of variability. NoSQL database popularity has increased due to its flexible schema, easy query interface, and scalability to handle such heterogeneous data. Document oriented NoSQLs are more popular to handle such heterogeneous data with flexible schema and a semi-structured data model. A flexible schema facilitates multiple ways to represent different types of variability that exist in the data collection. However, the impact of choosing different data representations on storage, energy, and performance is unknown. On the other hand, analysing the scope of optimising storage and energy efficiency is recommended for sustainable development. This paper discusses key components of the document store schema, its alternate implementations, and empirically analyse its impact on performance, storage space, and energy requirements. The aim of this paper is to assist developers in choosing an appropriate representation of each schema component for their application in the document stores. Considering its high popularity rating, MongoDB is chosen here for empirical analysis.

Keywords Schema · NoSQL · Document store · MongoDB · Energy consumption · Performance · Optionality · Type variability · Multivalued · Index

1 Introduction

These days, most modern applications use data collected from the same source at different times or from various sources, which may produce collections of data in differ-

M. Shah (✉)

Computer Science and Engineering Department, Nirma University, Ahmedabad, India
e-mail: monika.shah@nirmauni.ac.in

A. Kothari

Gujarat Technological University, Ahmedabad, India
e-mail: amitdkothari@gmail.com

ent formats, types, and attributes. These data formats can be structured, unstructured, or semi-structured. Product reviews, product galleries, comments, likes, surveillance data, and geospatial data are well-known examples of unstructured data. Unstructured data may occasionally be labelled as semi-structured data because it possesses one or more classifying attributes. Integration of structured data formats at different times or from different sources may also generate semi-structured data like products with distinct features, IoT sensor data, user profiles, etc. Therefore, our main focus is on semi-structured data formats. Traditional relational database software struggles with the continuous growth of such heterogeneous data. NoSQL's flexible schema accommodates unstructured and semi-structured data with ease, along with scalability, performance, and availability features. E-commerce, the Internet of Things (IOT), social networking, big data, and games are well-known examples of NoSQL-based applications. Key-value, graph-store, document store, and column store are four main NoSQL categories. The document store is more popular because of its semi-structured data model and most flexible schema.

Every data has some organisation, and a database schema describes the data organisation. So, a flexible schema means no schema is untrue. A flexible schema allows data records to have different schemas. Document stores are the most flexible NoSQL. However, schema design decisions affect NoSQL databases' scalability, consistency, and performance [1]. Gómez et al. [2] and Shah et al. [3] shows how document store schema affect query performance, storage, and energy use. It shows schema design matters even in NoSQL. However, document store schema design is usually ad-hoc, resulting in complex query design, schema modification to optimise query performance. There are publications on document store logical schema, but there are many data structure implementations for each component of logical schema. This implementation affects storage, energy, and query performance. These document store implementations rarely describe their effects.

The primary goal of this paper is to investigate and evaluate the effects of various alternative implementations of logical schema components. This article's primary contributions are as follows: (i) identifying the key components of a document store's logical schema; (ii) investigating various possible implementations of these components; (iii) analysing the effect of these various implementations on storage, energy consumption, and performance; (iv) analysing cost of adopting flexibility in document store schema; and (v) identifying a possible trade-off scenario.

The article continues like this. Section 2 reviews related literature. Section 3 describes key components of document store schema and their alternative implementations in document store. Section 4 describes the experimental evaluation. Section 5 discusses experiment results. Section 6 concludes experimental findings with research opportunities and future work ideas.

2 Literature Review

To avoid impedance mismatch, developers choose semi-structured data formats. Semi-structured data storage and querying issues are discussed in [4].

Flexible schemas attract NoSQL and schema design requires proper modelling [5, 6]. Chebotko et al. [7] and Mior et al. [8] suggests physical model rules for column families. The nesting depth level increases the storage and query execution time [9]. Gómez et al. [2] and Shah et al. [3] conclude that no schema is uniformly good or bad for all query pattern. Some articles [6, 9–15] guide on document store schema design for different access pattern. Specifically, [12, 16] are examples of articles guiding relational to document store migration. Imam et al. [17] recommends document storage cardinalities. Document conceptual design is also proposed [6]. Data modelling for analytical processing needs anticipatory schema building and evaluation (soransso2018DataModelingAnalytical). 11 metrics utilising document width, depth, and redundancy can help to select suitable schema [18].

Thakkar et al. [19] shows motivation for power conservations through hardware, power management, database level, etc. [20] recommends optimising NoSQL energy consumption by reducing waiting energy. Shah et al. [21] reviews NoSQL energy efficiency related work. According to [22, 23], query optimisation in NoSQL reduces energy consumption without compromising performance. According to [24], MongoDB uses more energy than relational databases, and modelling affects application performance and database capacity. Schema design may optimise energy consumption [3]. NoSQL energy consumption still lacks energy optimisation.

NoSQL requires design, but physical design research is scarce, according to [25]. Hewasinghage et al. [9] compare some implementations of semi-structured data in relational databases and document stores for basic data access patterns. But, perspective of relational and NoSQL database design is much different. Document store schema design research emphasises good schema design.

3 Document Store Schema Elements and Their Implementations

A database schema is an effective explanation of how data is arranged in a database. The document store's schema is best described by a flexible, semi-structured, and hierarchical data model. Flexible schema allows all documents in a collection to have a varied schema (fields, format, and order). Semi-structured data has some structure but is more flexible. Semi-structured data models describe document schemas with key-value pairs for each data attribute. Unlike structured data models, semi-structured data supports a hierarchical data structure that contains nested information.

The logical schema in the document store describes the interrelationships of documents using an embedding or referencing data model, attributes, and their types. Attributes of a document can be simple, composite, optional, or multivalued. Physical

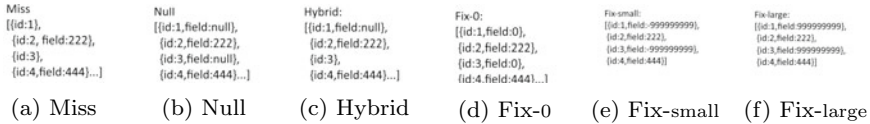


Fig. 1 Alternative physical schema design for optional attribute

schema show NoSQL-specific implementation of logical schema component. The document store physical schema includes data structure representation for attributes, index fields, index type specification, sharding, and replication configuration.

Although the document store is the most flexible NoSQL, it is said that adopting diverse schemas may add programming complexity afterwards. NoSQL data organisation methods may vary in performance, scalability, and integrity [26]. It is also reported that proper schema can help to reduce computation, I/O, and user contention [27]. It inspires research on the performance and energy impact of different document store schema designs. But it is difficult to discuss the consequences of all the important aspects of the physical schema of a document store in a single article. This article analyses the impact of alternative implementations of optional attributes, multivalued attributes, type variability, and index type selection.

3.1 Attribute Optionality

Time constraints, technical challenges, the fact that not all real-world items share the same attributes, and the fact that not all contacted sources offer all desired information all increase the likelihood of data gaps. As a result, the logical schema of this collection may contain optional attributes. Here, three different strategies to implement the optional attribute are studied. (i) *Miss*: document misses optional attribute with null data; (ii) *Null*: null keyword represents null data value; (iii) *Fix*: a fix value represents null data. Selecting a value for null data in *Fix* is another issue. The experiment analyses *Fix-0*, *Fix-large*, and *Fix-small*, where 0, a large number (999999999), and a small number (-999999999) represent null values, respectively. Figure 1 shows optional attribute representations. The last implementation *Hybrid* is designed to test the schema flexibility for optional attributes, where some documents may contain null values while others skip attributes for null data.

3.2 Type Variability

Data for the same attribute from various sources may come in a variety of formats. A product’s pricing or star rating may be presented as text by one seller and as numbers by another. The integration process may convert all price data to numeric

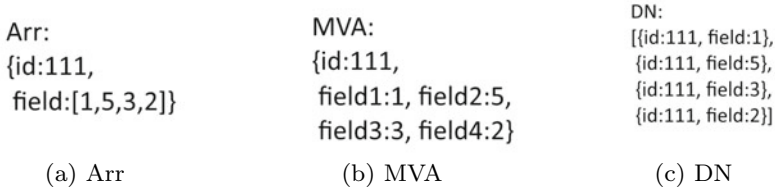


Fig. 2 Alternative physical schema design for a multivalued attribute

or text format or keep it in its original format, which is acceptable in the document store as per its type variability feature. To know the impact of different data types or type flexibility, the same data is tested in 3 formats: `Miss_Num`, `Miss_Txt`, and `Hybrid_Type`.

3.3 Multivalued Attribute

Real-world applications often use data objects with multivalued attributes such as addresses, sizes, colours, ratings. Relational database modelling recommends a separate table for multivalued attributes and entity ids. Document storage accommodates multivalued attribute in one document to avoid expensive `Join` operation. This article compares array (`Arr`), multiple attributes (`MVA`), and denormalized (`DN`) data structures for multivalued attributes. `DN` creates a new document for each multivalued attribute value and copies all other attributes (Fig. 2).

3.4 Index

Applications run faster when indexed data is used. Modern document storage typically offers many indexing options for different kinds of data. The content of arrays can be indexed by multikey indexes, text can be processed by text indexes, and geographical queries can be efficiently executed by using geospatial indexes. Although contrasting B-tree and hash indexes can be useful for future sharding type selection, many NoSQLs provide the `Sparse` and `Partial` for indexing not-null data and data of interest, respectively.

4 Experiment Setup

The study compares impact of alternative implementations of document store schema components presented in Sect. 3. MongoDB, a most popular document store [28],

is used in the experiment. MongoDB 5.0.14 runs on 64-bit Ubuntu 18.04.6 LTS Bionic, an Intel Xeon(R) CPU E5-2630 v3 @ 2.40 GHz x 16, 16 GB of memory, and a 441.8 GB disc partition. This study examines storage size, performance, and energy consumption (EC) of alternate physical schema design of document store.

In-memory databases are popular due to cheaper memory and greater performance. With increasing computation load, low power and high performance memory is still a research demand [29]. `stat().storageSize` and `stat().Size` of the collection in MongoDB represent compressed data on disc and uncompressed data on WiredTiger cache respectively. Cache memory is important for data processing, while Disc memory is crucial for massive data sets. Size and storageSize are in Megabytes (MB). Response time in milliseconds (ms) indicates query performance. The dataset and related querysets are generated as per description given in subsections generated as per des. PowerAPI is a popular software-defined power metre for real-time software power estimation. A Docker-based PowerAPI is installed on MongoDB's server. It records energy in Joules every second. Energy consumption of a query is computed using product of t (response time in second) and P (average power consumption during query execution). Every query is executed 3 times. Average performance and energy consumption are taken into account.

4.1 Data and Queryset for Attribute Optionality

Ten million documents with an ID and an optional field are generated and transformed to the six formats described in Sect. 3.1. Attribute optionality query sets include `CountNull`, `CountNotNull`, `CountRange`, `FindRange`, and `Insert` to count documents with null, not-null, count and search documents with given data range in optional attributes, and insert documents, respectively. Insert operations are executed in bulk. These query sets are tested with a B-tree index, a hash index, a sparse index, and a partial index. The default B-tree index addresses all documents. With the help of a partial index, a true sparse index is generated that addresses only non-null data. The query set applies different indexes to optional fields for different percentages of null data.

4.2 Data and Query Set for Type Variability

Ten million documents have an ID, and an optional field with numeric data is generated for `Miss_Num`. It is transformed to `Miss_Txt`, where the numeric data of the optional field is represented in string format such as '123'. In `Hybrid_Type` collection, a random set of documents have an optional field value in text format, and the rest have a numeric optional field. As standard practise, this field is indexed. The query set of type variability includes `InsertMany`, `CountRange` and `FindRange` queries.

4.3 Data and Query Set for Multivalued Attribute

Multivalued attributes, such as sizes, addresses, and colours, are short lists of distinct values that can be inserted, updated, or deleted and searched for any value from the list. Multivalued attributes, such as a product's ratings, are inserted or aggregated. The experiment generates two types of datasets: (i) a multivalued field with a short list of distinct values (5, 10, and 20 values), and (ii) a multivalued field with a long list (5, 10, 50, 100, and 300 values) with the possibility of repeated values. The query set for the first type of dataset includes `Insert`, `Delete`, and `FindIN` operations; and the second type of dataset include `Insert`, `FindRange`, and `Aggregate` queries. An aggregate query applies an aggregate operation to the multivalued attribute values of a data object. MVA is unsuitable for multivalued attributes with a long list. Therefore, MVA is tested for datasets having few values. The B-tree index is applied to the multivalued field(s) at `Arr`, and MVA.

5 Result Analysis

This section summarises the results of the experiment described in Sect. 4 using a radar chart, which ranks alternative design choices for schema components. Closer to the centre is considered a better choice.

5.1 Analysis of Attribute Optionality and Index

Figure 3 shows that storage of compressed data on disc and uncompressed data on cache for `Miss` is both optimal and optionally proportionate. Figure 3a shows that every null value needs some space but less than any data. Optionality proportionate storage size is not observed for disc data of `Null` and `Fix`, which shows the research scope of optimising compression algorithms. 0, 999999999, and -999999999 all belong to the double datatype. As shown in Fig. 3c, picking any of these settings does not influence the size on the cache, but affects the size on the disc. Figure 4a, b indicate that storage requirement by B-tree index on `Miss`, `Null`, and `Hybrid` formats are proportional to optionality but less than Hash Index. Despite varying optionality, the B-tree index on `Fix` uses the same space. Figure 4d shows that indexing different parts separately needs a little more space than a single index on the entire dataset. Figure 5b, d shows that count queries execute much faster and use less energy than `Find` and `Insert` queries. `Miss` and `Hybrid` with a sparse index are efficient in both energy and performance for `Insert` and `Find` queries. Ranking of count queries are magnified in Fig. 5a, c. `Fix` with partial index is ranked high for optimal energy and performance by `CountNull`. It indicates comparing null values needs more power than comparing zeros.

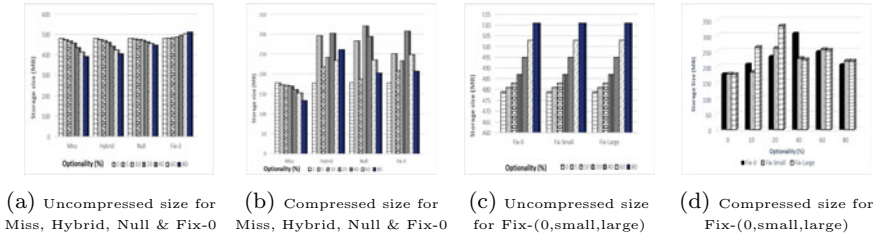


Fig. 3 Impact of optional attribute representations on storage size

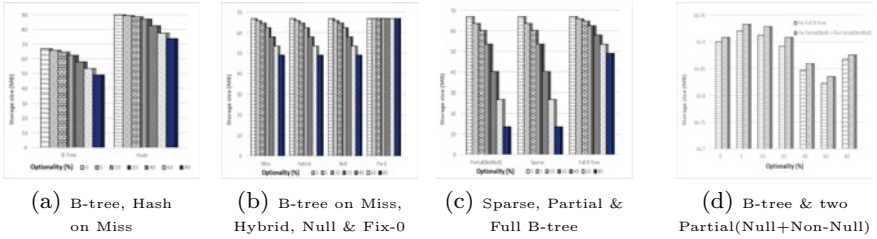


Fig. 4 Storage requirement for indexes on optional attribute

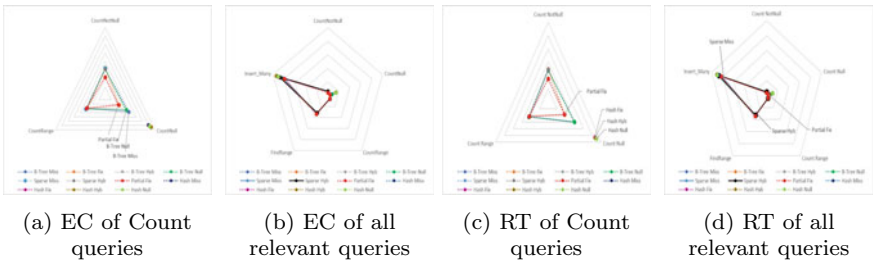


Fig. 5 Impact of optional attribute representations on EC and performance

5.2 Analysis of Incorporating Type Variability

Graphs in Fig. 6 show that text format uses more storage space than numeric type. Figure 7a, b rank alternative type representations for relevant query set. Numeric data representation is energy-efficient and speedy for all query types. Flexible schema permits any data type, such as Hybrid_Type, which scans numeric and text index keys separately. It may require a little more processing power. It may be a reason of more energy requirement by Hybrid_Type for CountRange and InsertMany query patterns. Document scanning depends on storage size and Hybrid_Type needs less storage space than Miss_Txt. It may be a reason of less energy consumption by Hybrid_Type in compare to Miss_Txt.

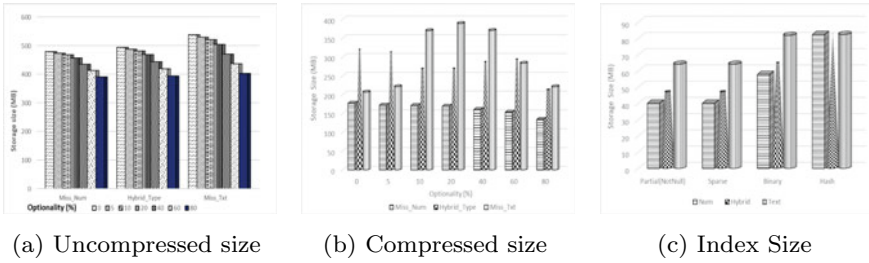


Fig. 6 Impact of type variability on storage size

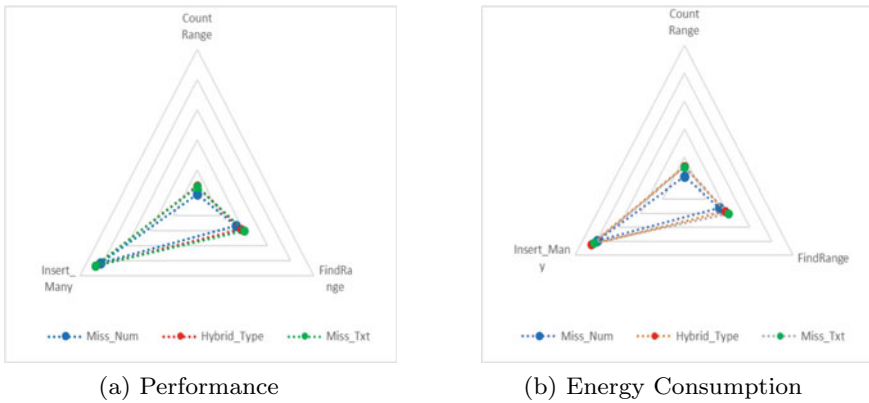


Fig. 7 Impact of type variability on EC and performance

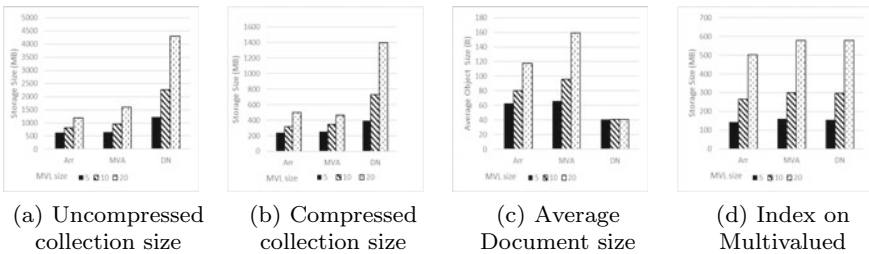


Fig. 8 Impact of multivalued attribute implementations on storage size

5.3 Analysis of Multivalued Attribute Implementations

Figure 8a, b, d illustrated that *Arr* has optimal storage requirement for entire data collection as well as index of multivalued attribute, while Fig. 8c shows that *DN* is optimal as well as consistent for single document size.

Figure 9a, c rank multivalued attribute implementations using average tuple cost with respect to energy consumption and performance. The ranking of read queries

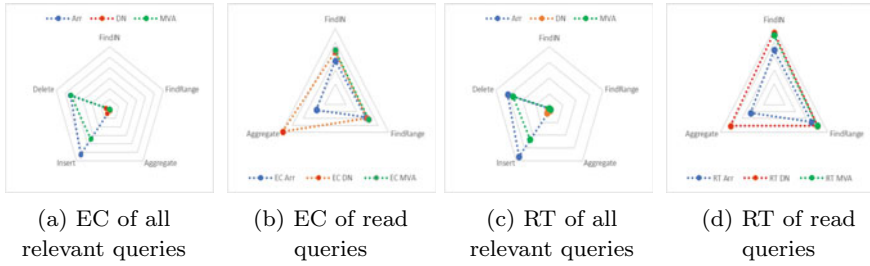


Fig. 9 Impact of multivalued attribute implementations on EC and RT

is not clearly visible because their EC and RT are much lower than write operations like Insert and Delete. Figure 9b, d rank MVA implementations for read operations. No implementation of multivalued attribute fits all data access patterns. Arr is storage-, energy- and performance-efficient for read operations. DN results in performance and energy efficiency for write operations. It may be because of smaller document size. A little trade-off observed. SPSVERBc13 executes FindIN and FindRange queries faster but utilises more energy than DN.

5.4 Discussion

This empirical study found some interesting findings. More storage increases execution time, which dominates energy. Therefore, storage optimization for data and indexes is recommended. Miss, Arr, and Miss_Num are storage-efficient implementations for attribute optionality, multivalued attributes, and numeric data, respectively. A smaller document size like DN is energy-efficient for inserting or processing a single multivalued attribute. Sparse and partial indexes reduce the storage requirement for indexes.

The investigation found index scanning for zero uses less power than null values. Fix and Arr are energy-efficient for compute intensive operations on optional attributes and multivalued attributes, respectively.

Power and processing time are prime components of energy consumption. Power optimization scale is usually much smaller than performance optimization. For example, DN uses 8–10% less power, but Arr needs 15–35% less response time for FindRange. It may be a reason that most researchers focus on performance optimization.

Flexible schema simplify data integration but incur costs. For example, numeric data representation in numeric and string forms would increase query complexity, demand separate index scans for number and string data, and increase energy consumption.

No document schema component implementation is efficient for all data access patterns. On optional attributes, Miss is energy-efficient for FindRange and

Insert queries, while Fix with partial index is energy-efficient for count queries. On the multivalued attribute, DN and Arr are efficient for write and read operations, respectively. Optimization level of each operation helps to select the schema design for the entire application. For example, if an application has Insert and CountNull operations on optional attributes, then efficient physical schema design for Insert will be considered.

6 Conclusion and Future Work

This article discusses document store schema design alternatives for optional attributes, multivalued attributes, and type variability. Additionally, it shows how these implementations, index type choices, and flexible schema affect storage space requirements, energy consumption by the document store, and application performance. This article concludes that no optimal schema design exists that fits all data access patterns, and flexibility has a cost. The highlighted trade-offs may help developers select an energy-efficient physical schema design for optionality, multivalued attributes, and type variability. The impact of choosing a shard key, alternate implementations of sharding, and alternate schema designs for given data interrelationships may be investigated in future works.

References

1. de Lima C, dos Santos Mello R (2015) A workload-driven logical design approach for NOSQL document databases. In: Proceedings of the 17th international conference on information integration and web-based applications & services, pp 1–10
2. Gómez P, Casallas R, Roncancio C (2016) Data schema does matter, even in NoSQL systems! In: 2016 IEEE tenth international conference on research challenges in information science (RCIS), pp 1–6
3. Shah M, Kothari A, Patel S (2021) Influence of schema design in NoSQL document stores. In: International conference on mobile computing and sustainable informatics. Springer, Berlin, pp 435–452
4. Abiteboul S (1997) Querying semi-structured data. In: Database theory ICDT'97: 6th international conference Delphi, proceedings. Springer, Berlin, pp 1–18
5. Atzeni P (2016) Data modelling in the NoSQL world: a contradiction? In: Proceedings of the 17th international conference on computer systems and technologies 2016. ACM, pp 1–4
6. Varga V, Jánosi-Rancz KT, Kálmán B (2016) Conceptual design of document NoSQL database with formal concept analysis. Acta Polytech 13(2):229–248
7. Chebotko A, Kashlev A, Lu S (2015) A big data modeling methodology for Apache Cassandra. In: 2015 IEEE international congress on big data, pp 238–245
8. Mior MJ, Salem K, Aboulnaga A, Liu R (2017) NoSE: schema design for NoSQL applications. IEEE Trans Knowl Data Eng 29(10):2275–2289
9. Hewasinghage M, Nadal S, Abelló A (2020) On the performance impact of using JSON, beyond impedance mismatch. In: New trends in databases and information systems: ADBIS 2020 short papers, proceedings, vol 24. Springer, Berlin, pp 73–83

10. Chen L, Davoudian A, Liu M (2022) A workload-driven method for designing aggregate-oriented NoSQL databases. *Data Knowl Eng* 142
11. Imam AA, Basri S, Ahmad R, Watada J, Gonzalez-Aparicio MT (2018) Automatic schema suggestion model for NoSQL document-stores databases. *J Big Data* 5(1):1–17
12. Jia T, Zhao X, Wang Z, Gong D, Ding G (2016) Model transformation and data migration from relational database to MongoDB. In: 2016 IEEE international congress on big data (BigData congress), pp 60–67
13. Razoqi SA (2021) Data modeling and design implementation for CouchDB database. *AL-Rafidain J Comput Sci Math* 15(1):39–55
14. Rossel G, Manna A (2020) A big data modeling methodology for NoSQL document databases. *Database Syst* 37
15. Roy-Hubara N, Sturm A, Shoval P (2021) Designing document databases: a comprehensive requirements perspective. In: *Advances in conceptual modeling: workshops CoMoNoS, EmpER, proceedings, Canada, vol 40*. Springer, Berlin, pp 15–25
16. Stanescu L, Brezovan M, Burdescu DD (2016) Automatic mapping of MySQL databases to NoSQL MongoDB. In: *Federated conference on computer science and information systems*. IEEE, pp 837–840
17. Imam AA, Basri S, Ahmad R, Aziz N, Gonzalez-Aparicio MT (2017) New cardinality notations and styles for modeling NoSQL document-store databases. In: *TENCON 2017—2017 IEEE region 10 conference*, pp 2765–2770
18. Gómez P, Roncancio C, Casallas R (2018) Towards quality analysis for document oriented bases. In: *International conference on conceptual modeling*. Springer, Berlin, pp 200–216
19. Thakkar A, Chaudhari K, Shah M (2020) A comprehensive survey on energy-efficient power management techniques. *Procedia Comput Sci* 167:1189–1199
20. Li T, Yu G, Liu X, Song J (2014) Analyzing the waiting energy consumption of NoSQL databases. In: *Proceedings of 12th international conference on dependable, autonomic and secure computing, DASC 2014*. IEEE, pp 277–282
21. Shah M, Kothari A, Patel S (2022) A comprehensive survey on energy consumption analysis for NoSQL. *Scalable Comput Pract Exp* 23(1):35–50
22. Mahajan D, Blakeney C, Zong Z (2019) Improving the energy efficiency of relational and NoSQL databases via query optimizations. *Sustain Comput Inf Syst*
23. Mahajan D, Zong Z (2017) Energy efficiency analysis of query optimizations on MongoDB and Cassandra. In: *2017 Eighth international green and sustainable computing conference (IGSC)*, Orlando, FL. IEEE
24. Bani B, Khomh F, Guéhéneuc YG (2016) A study of the energy consumption of databases and cloud patterns. In: *Service-oriented computing: 14th international conference, proceedings, vol 14*, Canada. Springer, Berlin, pp 606–614
25. Antonio B, Daniel L (2011) A call to arms: revisiting database design. *ACM SIGMOD Record*
26. Atzeni P, Bugiotti F, Cabibbo L, Torlone R (2016) Data modeling in the NoSQL world. *Comput Stan Interfaces*
27. Scherzinger S, Sidortschuck S (2020) An empirical study on the design and evolution of NoSQL database schemas. In: *Conceptual modeling: 39th international conference, Vienna, Austria*. Springer, Berlin, pp 441–455
28. Db-engines ranking (2023). <https://db-engines.com/en/system/MongoDB>. Last accessed on 14 Apr 2023
29. Badgujar J, Kale V, Shah M, Parekh R (2019) Design and simulation of single electron transistor based SRAM and its memory controller at room temperature. *Int J Integr Eng* 11(6):186–195

On Significance of Subword Tokenization for Low-Resource and Efficient Named Entity Recognition: A Case Study in Marathi



Harsh Chaudhari, Anuja Patil, Dhanashree Lavekar, Pranav Khairnar, Raviraj Joshi, and Sachin Pande

Abstract Named entity recognition (NER) systems play a vital role in NLP applications such as machine translation, summarization, and question-answering. These systems identify named entities, which encompass real-world concepts like locations, persons, and organizations. Despite extensive research on NER systems for the English language, they have not received adequate attention in the context of low-resource languages. In this work, we focus on NER for low-resource language and present our case study in the context of the Indian language Marathi. The advancement of NLP research revolves around the utilization of pre-trained transformer models such as BERT for the development of NER models. However, we focus on improving the performance of shallow models based on CNN and LSTM by combining the best of both worlds. In the era of transformers, these traditional deep learning models are still relevant because of their high computational efficiency. We propose a hybrid approach for efficient NER by integrating a BERT-based subword tokenizer into vanilla CNN/LSTM models. We show that this simple approach of replacing a traditional word-based tokenizer with a BERT-tokenizer brings the accuracy of vanilla single-layer models closer to that of deep pre-trained models like BERT. We show the importance of using subword tokenization for NER and present our study toward building efficient NLP systems. The evaluation is performed on L3Cube-MahaNER dataset using tokenizers from MahaBERT, MahaGPT, IndicBERT, and mBERT.

H. Chaudhari · A. Patil · D. Lavekar · P. Khairnar · S. Pande
Pune Institute of Computer Technology, Pune, Maharashtra, India
e-mail: sspande@pict.edu

R. Joshi (✉)
Indian Institute of Technology Madras, Chennai, Tamilnadu, India
e-mail: ravirajoshi@gmail.com

H. Chaudhari · A. Patil · D. Lavekar · P. Khairnar · R. Joshi
L3Cube, Pune, Maharashtra, India

Keywords Low-resource languages · Named entity recognition · Deep learning · Natural language processing · Convolutional Neural Network · Bidirectional Long Short-Term Memory · Long Short-Term Memory · Marathi NER · Efficient NLP

1 Introduction

Named entity recognition (NER) plays an very important role in natural language processing (NLP) by identifying and classifying named entities in text [1, 2]. While NER has been extensively studied in various languages, there has been limited attention given to the significance of subword tokenization for NER in low-resource languages [3]. Marathi, as one of the prominent regional languages, holds great importance due to its status as a mother tongue for millions of native speakers [4]. Furthermore, Marathi is a morphologically rich language, making it challenging to build accurate NER systems. The existing state-of-the-art models often come with high computational costs, rendering them less practical for real-world applications.

The motivation behind this research paper is to bridge the gap between deeper transformer models with high computation costs and shallow deep learning models that cannot capture the intricacies and nuances of the Marathi language. The deeper pre-trained models provide state-of-the-art performance while non-pre-trained shallow deep learning models are computationally efficient but do not achieve top performance. We aim to address this gap by combining the best of both worlds to enhance the performance of shallow models based on Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) Networks, and BiLSTM. Through modifications in the tokenization process, we have achieved significant improvements in accuracy, making the performance of these shallow CNN/LSTM models comparable to that of state-of-the-art BERT models.

Subword tokenization has emerged as a promising technique in NLP, splitting words into subword units to capture the morphological structure of a language more effectively [5–8]. In this paper, we highlight the importance of subword tokenization for the NER task in low-resource Marathi language. Moreover, the concept can be extended to other low-resource, morphologically rich languages. Our research presents generic approaches to facilitate the development of more accurate and efficient NER systems for low-resource languages.

Our solution aims to tackle the problem of the low-resource named entity recognition (NER) by proposing an extremely shallow model with just one layer, achieving high efficiency without compromising performance. We propose a hybrid solution to integrate subword tokenization from BERT into our shallow model as depicted in Fig. 1. This integration allows us to leverage the linguistic coverage of BERT’s sub-tokenizer while maintaining the efficiency of our shallow architecture. Additionally, we specifically focus on the Marathi language, which is known for its rich morphology and is susceptible to out-of-vocabulary (OOV) tokens. We make use

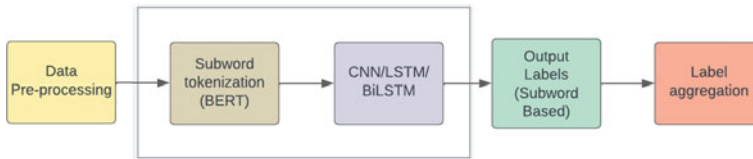


Fig. 1 Our hybrid approach

of subword tokenizers from monolingual Marathi transformer models like L3Cube-MahaBERT¹ [4], MahaGPT,² and multilingual BERT models like mBERT and Indic BERT. We show that MahaBERT tokenizer + vanilla CNN works the best on the L3Cube-MahaNER [3] dataset.

The main contributions of this work are as follows: We present a hybrid approach for low-resource NER by combining a vanilla single-layer CNN/LSTM model with BERT-based subword tokenizer. We present a comparative analysis of different monolingual and multilingual tokenizers for the Marathi language and show that the MahaBERT tokenizer + CNN model works the best.

2 Related Work

In the early 1990s, the term ‘Named Entity Recognition’ was introduced in the field of natural language processing (NLP) [9]. Over time, various statistical, machine learning, and deep learning techniques have been developed for NER [1].

Regarding specific studies, Chopra et al. [10] employed a Hidden Markov model for NER in the Hindi Language, utilizing 12 tags and achieving 97.14% accuracy. Li et al. [1] proposed novel CNN and LSTM architectures for NER, working with five English and one Chinese datasets. Additionally, researchers have demonstrated that combining deep learning with active learning can enhance the performance of NLP models. Another notable work by [11] introduced the first open neural NLP model for German NER tasks.

Evaluation of state-of-the-art machine learning approaches for named entity recognition on the Semantic Web was conducted by [12]. The Multilayer Perceptron performed best in terms of *f*-score, with 0.04% higher recall than Random Forest. However, poorer results were observed in the entity-based evaluation, where MLP ranked second to Functional Trees.

Two novel neural architectures, namely LSTMs and StackLSTM, have been developed to improve NER performance across multiple languages, including English. The LSTM-CRF model outperformed other systems, even those utilizing externally labeled data such as gazetteers. Similarly, the StackLSTM model demonstrated

¹ <https://huggingface.co/l3cube-pune/marathi-bert-v2>.

² <https://huggingface.co/l3cube-pune/marathi-gpt>.

superior performance compared to previous models that lacked external features, as reported in [13].

Challenges in performing NER in Indian languages using a Hidden Markov Model (HMM) were discussed by [14]. The researchers handled a total of seven named entity tags and achieved accuracies of 86% for Hindi, 76% for Marathi, and 65% for Urdu. They experimented with Conditional Random Fields and Maximum Entropy models, altering the feature set to identify the most effective feature set for each model [15].

In [16], the authors investigated three neural network-based models for Indonesian named entity recognition. The BiLSTM-CNNs + pre-trained word2vec embedding model exhibited strong performance with an F1-score of 71.37 [17] focused on NER in Twitter, constructing a new dataset called Tweet-NER7, which contained annotated entities of seven types across 11,382 tweets. In the analysis, three crucial temporal factors were taken into account: the deterioration of NER models in the short term, approaches for adapting a language model across varying timeframes, and the potential use of self-labeling as a substitute in scenarios where recently labeled data is scarce.

Lastly, Litake et al. [3] introduced L3Cube-MahaNER, a significant gold Marathi NER dataset with eight target labels. The dataset was benchmarked on various CNN, LSTM, and transformer-based models such as MahaBERT, IndicBERT, mBERT, and XLM-RoBERTa. Similarly, in [18], the authors focus on NER in low-resource Indian languages like Marathi and Hindi. The study investigates different transformer-based models, such as base-BERT, AIBERT, and RoBERTa, and assesses their effectiveness on publicly accessible NER datasets in Hindi and Marathi. The data reveal that the MahaRoBERTa model obtains greater performance in Marathi NER, whereas the XLM-RoBERTa model outperforms others in Hindi NER.

3 Dataset Details

3.1 Dataset Introduction

The dataset used in this work is the L3Cube-MahaNER [3], which is the first major gold standard Marathi NER corpus. Consisting of 25,000 sentences, the dataset is in Marathi Language. L3Cube-MahaCorpus [4] is used as the base for these sentences which are monolingual and extracted from news domains.

The dataset has predefined train, test, and validation splits. The training dataset consists of 21,500 sentence counts along with a 26,502 tag count, the test dataset consists of 2000 sentences and a 2424 tag count, and the validation dataset consists of 1500 sentences and 1800 tag count.

4 Proposed Methodology

4.1 Models Architectures

Deep learning has revolutionized NLP and become the go-to approach for many tasks. Deep learning models are capable of automatically learning complicated patterns and representations from massive volumes of data, making them particularly successful for applications such as named entity recognition. Depending on the task and the type of data being used, deep learning models can have a broad range of architectures.

CNN: The utilization of Convolutional Neural Network (CNN) models for named entity recognition (NER) is advantageous due to their ability to extract local context, handle word order invariance, perform feature composition, enable effective parameter sharing, and deliver strong performance on local context tasks.

In CNN-based NER models, the input text is typically converted into a sequence of word embeddings, representing each word in a high-dimensional space. In this particular model, a single 1D convolutional layer is employed, with the word embeddings having a dimension of 300. These embeddings are typically trained as part of the NER model training procedure. The 1D convolutional layer, which utilizes the ‘relu’ activation function and has 512 filters with a kernel size of 3, receives the embeddings after that. The output of the Conv1D layer is then fed into a dense layer that has the same size as the output layer and uses the ‘softmax’ activation function. As there are eight classes, the model generates eight output labels. The ‘rmsprop’ optimizer is employed in the training process.

LSTM: The utilization of Long Short-Term Memory (LSTM) models for named entity recognition (NER) is advantageous due to their ability to process sequential information, comprehend context, handle variable-length sequences, and address the issue of vanishing gradients.

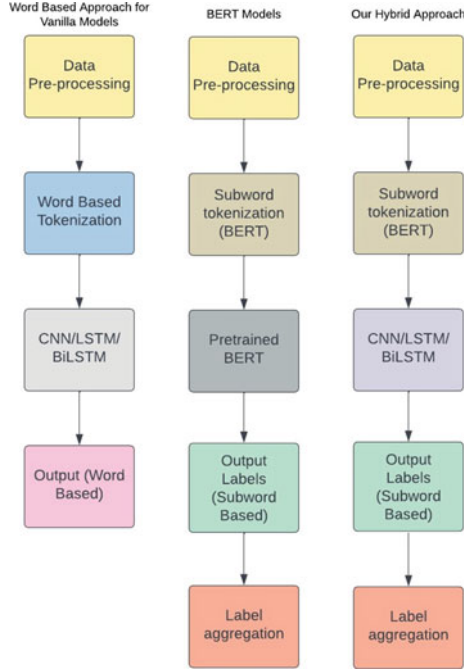
Similar to the CNN model, the LSTM model also follows a similar architecture. In this case, the model comprises a single LSTM layer with word embeddings of 300 dimensions. Following this layer, a dense layer with properties resembling those of the CNN model is present and contains 512 filters.

BiLSTM: The usage of Bidirectional Long Short-Term Memory (BiLSTM) models for named entity recognition (NER) is advantageous due to their ability to provide contextual understanding, improve representation learning, handle ambiguity robustly, fuse features, and handle out-of-order entities.

BiLSTM’s model architecture is comparable to that of CNNs, with a BiLSTM layer in place of the single 1D convolutional layer. This particular model uses a 512-hidden-unit BiLSTM layer and an embedding vector with 300 dimensions. There are 16 batches total.

While the CNN, LSTM, and BiLSTM architectures are typically considered pre-existing options, our research distinguishes itself primarily through the approach taken in the tokenization operation.

Fig. 2 There are three approaches: the first column consists of Word Based Approach for Vanilla Models [3], the second column consists of BERT Models [3], and the third column consists of our hybrid approach which additionally includes the allocation of labels to the sub-tokens before training and clubbing of labels generated by the models for the sub-tokens to the root tokens



4.2 Tokenization

Tokenization involves breaking sentences or paragraphs into tokens, which are then used for model training. Different types of tokenization exist, such as word-based, subword-based, and character tokenization. While word tokenizers were traditionally used with CNN/LSTM models, our research employs subword-based tokenizers in conjunction with these models. The proposed flow is shown in Fig. 2.

Subword-based Tokenizers: Subword tokenization enhances NER by effectively handling unfamiliar words, capturing morphological variations, improving generalization, and enabling cross-lingual transfer learning. Breaking words into smaller sub-tokens enables the model to identify and infer from hidden word components. This detailed approach is particularly beneficial for entities with morphological alterations, as it captures prefixes, suffixes, and stem variations.

Moreover, it enables generalization across similar terms and facilitates knowledge transfer across languages. Overall, subword tokenization broadens the scope of NER, accommodates morphological variations, and enhances performance specifically in the Marathi language. The core methodology described in this work is to integrate subword tokenizers with vanilla CNN/LSTM models. We specifically focus on BERT-based subword tokenizers.

The different types of Marathi subword-based tokenizers based on BERT are as follows:

Tokenizer	Text
Original Text	त्याआधी सिनेटने अमेरिकन सरकारच्या कर्मचाऱ्यांनी टिकटॉक वापरू नये
MahaBert-Scratch	[CLS] तय ##ा ##आधी सिन ##टन अमर ##िकन सरकारच ##या
MahaBERT	[CLS] त्याआधी सिने ##टने अमेरिकन सरकारच्या कर्म ##चा ##प्यांनी टिकट
mBERT	[CLS] त्या ##आ ##धी स ##िने ##टन ##े अमेरिकन सरकार
Indic-BERT	[CLS] त्याआधी सिने टने अमेरिकन सरकारच्या कर्मचाऱ्यांनी टिकट ॉक वापरू
Maha GPT	[CLS] त्याआधी सिनेट ने अमेरिकन सरकारच्या कर्मचाऱ्यांनी टिकटॉक वापरू नये

Fig. 3 Tokens generated by various tokenizers

- MahaBERT-Scratch Tokenizer³ [4, 19].
- MahaBERT Tokenizer⁴ [4, 19].
- mBERT Tokenizer⁵ [20].
- IndicBERT Tokenizer⁶ [21].
- Marathi-GPT Tokenizer⁷ [4].

Eventually, these BERT-based tokenizers were used separately for each model and the metrics were calculated. A sample sentence split for each model is shown in Fig. 3.

There were two major requirements encountered during the sub-tokenization before training and after testing the models:

Requirement before training the model: The root word may be split into multiple subword tokens. So, the entity label of the root token needs to be passed to its corresponding subword tokens.

Requirement after testing the model: The labels generated by the model for the multiple sub-tokens need to be passed to the single root token.

³ <https://huggingface.co/l3cube-pune/marathi-bert-scratch>.

⁴ <https://huggingface.co/l3cube-pune/marathi-bert-v2>.

⁵ <https://huggingface.co/bert-base-multilingual-cased>.

⁶ <https://huggingface.co/ai4bharat/indic-bert>.

⁷ <https://huggingface.co/l3cube-pune/marathi-gpt>.

5 Results

In this research, we conducted experiments on Marathi, a low-resource language, using shallow vanilla models such as CNN, LSTM, and BiLSTM. Our goal was to conduct named entity recognition on the MahaNER dataset. To bridge the gap between deep pre-trained models like BERT transformers and non-pre-trained shallow deep learning models, we augmented these low-level models with subword integration. We employed BERT-based tokenizers including MahaBERT-Scratch, MahaBERT, mBERT, MuRIL, and indicBERT tokenizers. In this section, we present the achieved recall, precision, F1-scores, and accuracy (in percentage) after training the models. We also provide the results obtained using word-based tokenizers from previous research conducted on the same dataset. The performance of different tokenizer types is shown in Table 1 and Fig. 4. The baseline comparison with pre-trained MahaBERT model and word-based vanilla CNN model is shown in Table 2.

Based on the evaluation, we observe that the MahaBERT tokenizer produces the highest F1-scores among all benchmarked sub-tokenizers and the word-based tokenizer: 82.1% for CNN, 76.0% for LSTM, and 82% for BiLSTM. We also consider the F1-scores of the word-based tokenizer as the baseline, 79.5% for CNN, 74.9% for LSTM, and 80.4% for BiLSTM. Thus, the sub-tokenizers contribute to a significant improvement in the F1-score of the single-layer vanilla models. Subword tokenization expands the application of NER, accommodates morphological variations, and enhances performance in the Marathi language. As a result, our hybrid approach distinguishes itself from the traditional word-based approach and achieves results comparable to those obtained with BERT models.

6 Limitations

- The application of subword tokenization increases the length of the input sequence, resulting in longer training times.
- Although we have reduced the difference between vanilla models and BERT-based models to half, the scope for further improvements is still possible.

7 Conclusion

In this work, our focus was on performing named entity recognition (NER) in Marathi, a low-resource language. We addressed the challenge of bridging the gap between high-level models like BERT transformers and low-level models such as CNN, LSTM, and BiLSTM. We propose a hybrid approach to integrate BERT-based subword tokenizers into the vanilla CNN and LSTM models. We compare different tokenizers from monolingual models like MahaBERT, MahaGPT,

Table 1 Comparison table of F1-score, precision, recall, and accuracy of word-based tokenizer and subword-based tokenizers

Tokenizer/model	F1-score			Precision			Recall			Accuracy		
	CNN	LSTM	BiLSTM	CNN	LSTM	BiLSTM	CNN	LSTM	BiLSTM	CNN	LSTM	BiLSTM
Word-based	79.5	74.9	80.4	82.1	84.1	83.3	77.4	68.5	77.6	97.28	94.89	94.99
MahaBert-Scratch	76.4	65.2	75.2	81.3	76.9	83.9	72.6	58.8	69.9	95.0	93.0	95.0
MahaBERT	82.1	76.0	82.0	84.9	79.9	83.9	79.9	73.0	80.6	96.0	95.0	96.0
mBERT	66.4	50.0	75.4	78.1	72.3	78.0	59.4	41.7	73.2	93.0	91.0	94.0
Indic-Bert	81.8	75.6	81.2	83.8	82.1	80.4	79.9	70.8	82.7	96.0	95.0	95.0
Marathi GPT	80.4	58.2	74.5	81.4	80.9	84.0	79.6	56.4	68.9	95.0	94.0	95.0

Model name in the first columns represents the tokenizer type. The best scores are highlighted in bold

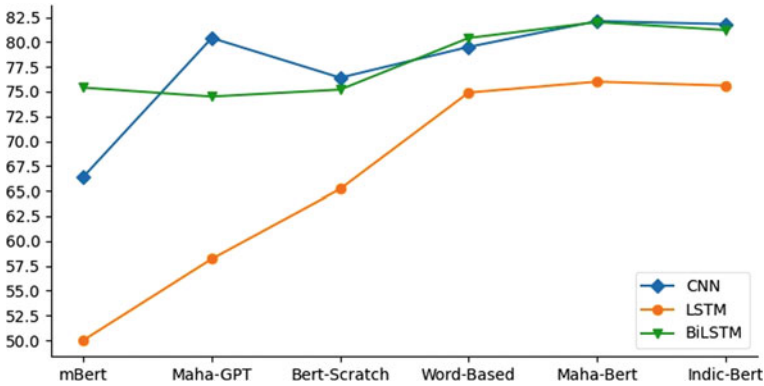


Fig. 4 F1-score for all sub-tokenizers

Table 2 F1-scores of CNN model with word-based tokenizer, MahaBERT model, and CNN model with MahaBERT tokenizer

Tokenizer/model	F1-score
CNN + word tokenizer	79.5
MahaBert	86.8
CNN + MahaBert tokenizer	82.1

MahaBERT-Scratch and multilingual models like IndicBERT, mBERT. Among the different models, the CNN model achieved the highest F1-score using the MahaBERT tokenizer. The increased accuracy and effectiveness of these low-level models can lead to reduced computation costs for NLP applications where NER serves as a fundamental operation.

Acknowledgements We would like to express our sincere gratitude toward the L3Cube mentorship program and our mentor for their continual support and guidance. We are grateful to Pune Institute of Computer Technology for encouraging and supporting us throughout the research period. The issue statement and ideas provided in this work are from L3Cube and its mentors and are a part of the L3Cube-MahaNLP project [22].

References

1. Li J, Sun A, Han J, Li C (2020) A survey on deep learning for named entity recognition. *IEEE Trans Knowl Data Eng* 34(1):50–70
2. Shen Y, Yun H, Lipton ZC, Kronrod Y, Anandkumar A (2017) Deep active learning for named entity recognition. *CoRR* abs/1707.05928. Preprint at <http://arxiv.org/abs/1707.05928>
3. Litake O, Sabane MR, Patil PS, Ranade AA, Joshi R (2022) L3cube-mahaner: a Marathi named entity recognition dataset and Bert models. In: *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pp 29–34

4. Joshi R (2022) L3cube-mahacorporus and mahabert: Marathi monolingual corpus, Marathi Bert language models, and resources. In: Proceedings of the WILDRE—6 Workshop within the 13th Language Resources and Evaluation Conference, pp 97–101
5. Joshi R, Joshi R (2022) Evaluating input representation for language identification in hindi-english code mixed text. In: ICDSMLA 2020. Springer Singapore, Singapore, pp 795–802
6. Kudo T (2018) Subword regularization: improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 66–75
7. Kudo T, Richardson J (2018) Sentencepiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. Preprint at <https://arxiv.org/abs/1808.06226>
8. Senrich R, Haddow B, Birch A (2016) Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1715–1725 (2016)
9. Patil P, Ranade A, Sabane M, Litake O, Joshi R (2022) L3cube-mahaner: a Marathi named entity recognition dataset and Bert models. Preprint at <https://arxiv.org/abs/2204.06029>
10. Chopra D, Joshi N, Mathur I (2016) Named entity recognition in Hindi using hidden Markov model. In: 2016 Second International Conference on Computational Intelligence and Communication Technology (CICIT), pp 581–586
11. Frei J, Kramer F (2021) GERNERMED—an open German medical NER model. CoRR abs/2109.12104. Preprint at <https://arxiv.org/abs/2109.12104>
12. Speck R, Ngonga Ngomo AC (2014) Ensemble learning for named entity recognition. In: Mika P, Tudorache T, Bernstein A, Welty C, Knoblock C, Vrandečić D, Groth P, Noy N, Janowicz K, Goble C (eds) The Semantic Web—ISWC 2014. Springer International Publishing, Cham, pp 519–534
13. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. Preprint at <https://arxiv.org/abs/1603.01360>
14. Singh J, Joshi N, Mathur I (2013) Part of speech tagging of Marathi text using trigram method. Preprint at <https://arxiv.org/abs/1307.4299>
15. Manamini S, Ahamed A, Rajapakshe R, Reemal G, Jayasena S, Dias G, Ranathunga S (2016) Ananya-a named-entity-recognition (NER) system for Sinhala language. In: 2016 Moratuwa Engineering Research Conference (MERCon). IEEE, pp 30–35
16. Sukardi S, Susanty M, Irawan A, Putra RF (2020) Low complexity named-entity recognition for Indonesian language using BILSTM-CNNs. In: 2020 3rd International Conference on Information and Communications Technology (ICOIACT). IEEE, pp 137–142
17. Ushio A, Neves L, Silva V, Barbieri F, Camacho-Collados J (2022) Named entity recognition in twitter: a dataset and analysis on short-term temporal shifts. Preprint at <https://arxiv.org/abs/2210.03797>
18. Litake O, Sabane M, Patil P, Ranade A, Joshi R (2023) Mono versus multilingual Bert: a case study in Hindi and Marathi named entity recognition. In: Proceedings of 3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2022. Springer, pp 607–618
19. Joshi R (2022) L3cube-hindbert and devbert: pre-trained Bert transformer models for Devanagari based Hindi and Marathi languages. Preprint at <https://arxiv.org/abs/2211.11418>
20. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186. <https://aclanthology.org/N19-1423>

21. Kakwani D, Kunchukuttan A, Golla S, Gokul N, Bhattacharyya A, Khapra MM, Kumar P (2020) IndicNLPsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp 4948–4961
22. Joshi R (2022) L3cube-mahanlp: Marathi natural language processing datasets, models, and library. Preprint at <https://arxiv.org/abs/2205.14728>

Analysis and Study of Bug Classification Quintessence and Techniques for Forecasting Software Faults



Shallu Juneja, Gurjit Singh Bhathal, and Brahmaleen K. Sidhu

Abstract Developers frequently save past faults in storage to better understand issues affecting software systems and these flaws can be divided into families. A flaw may occur at any step of the software coding cycle. The most popular names include bug, defect, flaw, fault, or error. In this paper, we examined various defect classification schemes and techniques for forecasting software faults. We also reviewed the severity of software bug, as well as the fault-type classification schema along with software system bugs. The IBM Orthogonal Defect Classification is a popular form of categorization (ODC). Based on a lot of knowledge about the faults, including their symptoms and semantics, their root causes, and a host of other factors, ODC proposes multiple orthogonal classifications of defects. Another method for categorizing defects is the IEEE 1044-2009 Standard. It has six families with severe defects. The research's objective is to learn about and investigate the major types of defects. A thorough grasp of the traits of software defects is necessary to create efficient models for the detection of software errors and their recovery. In this paper, we have also focused on various fault forecasting techniques, used datasets, fault percentage, performance measures, study objectives, and outcomes along with future scope.

Keywords Orthogonal Defect Classification · IEEE 1044-2009 Standard · Severity · Software fault

1 Introduction

In order to manage maintenance tasks and maintain bug information, bug reports are typically processed and tracked with the aid of Programming Tool [1]. After finding the problems, the classification of software bugs is necessary to comprehend the

S. Juneja (✉) · G. S. Bhathal · B. K. Sidhu
Computer Science Engineering Department, Punjabi University, Patiala, India
e-mail: shallujuneja9@gmail.com

S. Juneja
Computer Science Engineering Department, Maharaja Agrasen Institute of Technology,
New Delhi, Delhi, India

underlying reasons and offer the suitable remediation actions. By assigning the bug to the developer who can fix it in the desired amount of time, bug classification primarily serves to shorten the time needed to fix bugs. Several studies have concentrated on applying machine learning and other techniques to automate bug classification tasks in order to solve the bug classification problem.

A number of machine learning-based approaches have been put forth in the literature to aid developers in categorizing bugs. Debugging is a costly and time-consuming operation. To effectively recreate, localize, and solve issues, developers must choose the right tools, techniques, and strategies. These decisions are based on the developers' evaluation of the fault category for a specific bug report. Software developers frequently refer to the emergence of a flaw in a software system as a "software bug". So, a mistake that deviates from the software's standards is referred to as a defect. Today, users of software systems are encouraged to report any flaws they find using bug tracking tools like Jira, etc.

The objectives of this study are as follows:

- Conducting an analytic review of existing literature on
 - Software defect or fault classification schemes.
 - Models and techniques used for forecasting software faults.

The rest of this paper is divided into the following sections:

- Section 2 presents a review of the literature on defect classification schemes—ODC-based defect classification scheme, IEEE-based defect classification scheme along with severity of software bug, and types of faults are also explained. Schema for classifying software faults are also explained. Taxonomy of faults is also included.
- Section 3 covers techniques for forecasting software faults along with datasets, performance measures, etc.
- Section 4 provides conclusion of the study.

2 Defect Classification Scheme and Others

2.1 Defect-Type Family IBM-ODC

Orthogonal Defect Classification, a defect classification scheme created by IBM, has been used to classify defects in a variety of software systems across numerous sectors. Based on defect types, impact, and a variety of other factors, ODC contains numerous orthogonal categories (Table 1). ODC is divided into a number of categories, each of which groups flaws according to a certain criterion. It is suggested to use a variety of criteria, such as defect types, riggers, impact, targets, defect removal activities, ages, qualifiers, sources, [2].

Table 1 Defect-type family IBM-ODC [2–6]

Defect quintessence family IBM-ODC		Elucidation
Structural	Function or class object	“As it affects important functionality, end-user interfaces, product interfaces, interfaces with hardware architecture, or global data structure(s), the issue should need a formal design change”
	Interface or O–O messages	“Issues with inter-module, inter-component, inter-device driver, object, or function communication”
	Relationship in IBM-ODC	“Issues with relationships between objects, data structures and procedures. These connections might be conditional”
	National Lang. Support	“Problems encountered while implementing product functionality in languages other than English...”
Non-code	Documentation	“The problem is with the textual descriptions in user guides, installation manuals, online support and user messages”
	Build/merge package/ (configurations)	“Issues in the library systems, handling of change, or version control,” as well as “The challenges in files containing settings or parameters”
Control and data flow	Algorithm/method	“Efficiency or accuracy issues that have an impact on the work and can be resolved without seeking a design change by (re)implementing an algorithm or local data structure”
	Assignment/initialization	“Wrongly or not at all assigned value(s)”
	Checking	“Errors caused by inaccurate data validation in statement of conditions”
	Timing or serialization	“The required serialisation of a shared asset is not performed, incorrect resource or asset is serialised or the incorrect serialisation approach was used”

2.2 IEEE-Based Defect Classification Scheme

This standard offers a consistent method for classifying software anomalies, regardless of their origin or when they are discovered within the life cycle of a project, a product, or a system. Defect classification is one use for which classification data can be employed (Table 2). This standard aims to develop a common set of qualities that support industry approaches for analyzing software defect and failure data as well as to define a common data that allows diverse people and organizations to communicate effectively about software anomalies [7].

Table 2 Defect-type family IEEE [3, 7]

Defect quintessence family IEEE 1044		Elucidation
Logic and data	Logic data	Flaw in computational method, as detected in Natural Language specifications or implementation language, involving decision logic, branching, sequencing, or other aspects. Missing an else clause, for instance; incorrect operation sequencing; improper operand or operator in the expression lacking the necessary logic to check for or handle an error condition (such as a return code, the end of a file, a null value); input value not inside acceptable range; system response missing from the sequence diagram; business rule’s definition in the specification is ambiguous
		A flaw in a model, specification, or implementation that affects how data is defined, initialized, mapped, accessed, or used. Examples include a variable with no initial value or a flag that is not set. Incorrect column size or data type; the wrong variable name is used; Validity range is unclear; the data model’s incorrect relationship cardinality; incorrect or missing value in the choose list
Interface	Interface	A defect in the interface’s design or implementation
		A defect in a model, specification, or implementation that affects how data is defined, initialized, mapped, accessed, or used
Description	Elucidation	“Defect in programme elucidation or in the use installation or operation of the software”
Syntax	Syntax	“Deviation from a language’s defined rules...”
Standards Build-Config-Install	Various standards	Issues that arise during the building process, “Issues with configuration parameters or files,” or “Issues with the installation process”

Table 3 Severity of software faults [8]

Severity	Explanation
Major	It is related to majorly loss of a function
Minor	It is related to minorly loss of a function
Trivial	Cosmetic issues such as mistyped words
Critical	Crashes, loss of data, severe memory leak
Blocker	Blocks development, testing work
Normal	Not specified
Enhancement	Enhancing request

2.3 Severity of Software Faults [8]

Reporters can categorize bugs according to their severity using bug tracking systems. Developers will likely find this information useful because they would not have to sift through every report to attend to the ones with the highest severity first and the ones with the lowest severity second (Table 3). The severity function is intended to be used by developers to prioritize the bugs that need to be fixed first and classify issues according to their relevance. The priority parameter would also be used by developers to indicate the sequence in which issues should be fixed [8].

2.4 Major Software Faults

The main causes of software unsuccessful rate are referred to as fault types [9]. Definitions which are standard for ISO/IEC/IEEE 24765 are also included [10] (Fig. 1).

1. Requirement Faults: These faults include missing, modified, or incorrect requirements.
2. Coding Faults: Logical flaws, erroneous algorithms, missing code, and other defects in the source code are examples of coding faults.

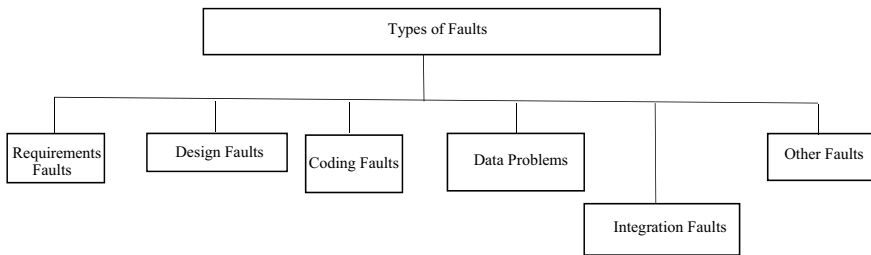


Fig. 1 Types of faults [9, 10]

Table 4 Fault-type classification schema [11]

Fault type	Explanation
Concurrency	Improper synchronization as well as erroneous atomicity assumptions
Memory and resources	Memory and resource management that is inefficient or wrong
Other	Bugs that cannot be fixed by modifying the Java code
Semantic	Requirements that are inconsistent, programmers' intentions, and actual implementations that do not fit into the categories listed above

3. Integration Faults: Faults related to the integration of components, subcomponents, or subsystems are categorized as integration faults.
4. Design Faults: Design flaws are caused by human errors during the system's design.
5. Data Faults: Data issues result in a failure in reaction to a certain pattern of data.
6. Other Faults: Other defects are sorts of faults that do not fit into any of the previous categories, e.g., Simulation Issues.

2.5 Fault-Type Classification Schema

See Table 4.

2.6 A Fault Taxonomy for Component-Based Software

A component-based system is made up of interconnected parts that interact to provide the desired behavior as a whole. Testing might be done on a single component, a group of connected components, or on the entire system (Fig. 2, Table 5).

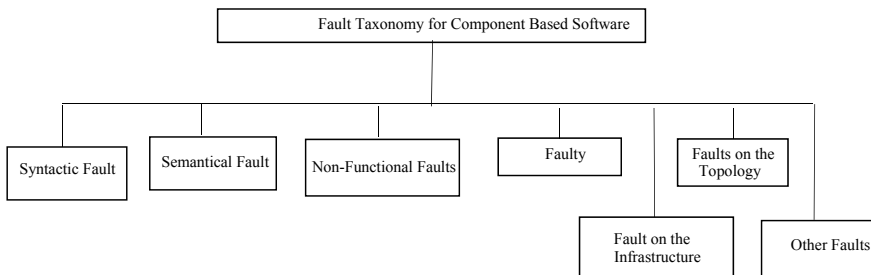


Fig. 2 Fault/error taxonomy for component-based software [12]

Table 5 Summary of the fault/error taxonomy presented in paper [12]

Main category	Sub-categories
Syntactic error	Violation of the interface
Semantic error	Misunderstood on the behavior, parameters, events, and interaction protocol
Non-functional error	Performances, quality of service
Connectors error	Disagreement on the protocol, quality of service
Infrastructure error	Underlying services and system
Topology error	Includes callback, re-entrance, and recursion

2.7 Characteristics of Bug in Software (Open Source)

See Table 6.

Table 6 Memory bug quintessence and semantic bug quintessence [13]

Set	Subset	Elucidation
Memory bug	Memory leak	Unused memory is not released
	Uninitialized memory read	Read data from memory before it is initialized
	Dangling pointer	Pointers continue to preserve freed memory addresses
	NULL pointer dereference	Null pointer dereference
	Overflow	Illegal access outside a buffer border
	Double Free	One memory region is freed twice
	Other	Other memory flaws/bugs
Semantic bug	Missing features	Although not implemented, a feature was intended to be
	Missing cases	A situation where a functionality is not used
	Corner cases	Some boundary cases are erroneously seen or disregarded
	Wrong control flow	The control flow is implemented wrongly
	Exception handling	Proper exception handling is missing
	Processing	Processing is incorrect
	Typo	Typographical errors
Other wrong functionality implementation	Any further semantic error that does not adhere to the specifications	

3 Techniques for Forecasting Software Faults

See Tables 7 and 8.

Table 7 Literature survey for forecasting software faults

S. No	Paper	Algorithm/ technique used	Used fault dataset/ repository	Fault percentage (%)	Used performance measures
1	2010 [14]	Decision tree and fuzzy rules	PROMISE, NASA	20.49	Accuracy
2	2011 [15]	SVM and PNNs	PROMISE, NASA	17.66	Accuracy
3	2012 [16]	Naïve Bayes, Bayes Network, and Nearest Neighbor (KNN)	Eclipse 2.0	20.76	ROC and AUC
4	2013 [17]	Bayesian Network Classifiers	PROMISE Eclipse Foundation	10.25	ROC, AUC H-measure
5	2013 [18]	SD and CN2-SD	PROMISE, NASA Turkish white-goods manufacturer Bug Prediction Dataset	14.41	Positive predictive value or precision
6	2014 [19]	GA, LR, ANN, SVM, DT, Cascade correlation network GMDH polynomial network	PROMISE, Softlab	10.81	Sensitivity Specificity Area Under Curve (AUC)
7	2014 [20]	Defect Prediction using Relational Association Rules (DPRAR)	PROMISE	18.44	Classification accuracy Probability of detection Specificity of the classifier Classification precision ROC, AUC
8	2015 [21]	Adaptive Neuro Fuzzy Inference System	PROMISE	17.62	ROC, AUC
9	2017 [22]	Fuzzy Rule-Based	PROMISE, NASA, MDP	12.84	Average test error

(continued)

Table 7 (continued)

S. No	Paper	Algorithm/ technique used	Used fault dataset/ repository	Fault percentage (%)	Used performance measures
10	2018 [23]	SSDFC Clustering	NASA datasets Eclipse dataset PROMISE	11.40	Probability of detection F-measure AUC
11	2019 [24]	Fuzzy-filtered neuro-fuzzy framework	PROMISE repository and software defect prediction dataset	14.93	Accuracy Mean absolute error Root mean square error G-mean AUC
12	2020 [25]	Naive Bayes Logistic regression J48 (Decision Tree) Dagging Decorate Grading MultiBoostAB RealAdaBoost Rotation Forest Ensemble Selection	PROMISE repository and software defect prediction (SDP) dataset	15.72	Precision/recall AUC Specificity G means
13	2021[26]	SVM DT LDA kNN Whale Optimization Algorithm (Variants)	PROMISE repository	32.30	AUC P-values obtained from Wilcoxon test Running time
14	2022 [27]	Salp Swarm Algorithm and Backpropagation neural network	PROMISE repository, OpenML repository, NASA	9.62	Confusion matrix, AUC, sensitivity, Accu., ER
15	2022 [28]	Bagging Decision Tree Multilayer perceptron Random tree Support vector machine Naïve Bayes AdaBoositM1	PROMISE, NASA BUG repository Jira repository	23.78	Accuracy, AUC (ROC) Root mean square error and F-Score

(continued)

Table 7 (continued)

S. No	Paper	Algorithm/ technique used	Used fault dataset/ repository	Fault percentage (%)	Used performance measures
16	2023 [29]	Naïve Bayes SVM K-STAR RF K-nearest Bagging Boosting Random forest Rotation forest Convolutional neural network Recurrent neural networks	Eclipse Foundation, JIRA, SFP XP-TDD	10.04	Accuracy Receiver operating characteristic Area Under Curve Cohen's Kappa metrics Probability of detection Probability of false Precision True-negative rate F-measure

4 Conclusion of the Study

In the present work, it was found that defect classification schemes have been proposed in the industry to help people comprehend flaws better. These defect categories can provide information about the frequency of various types of problems in a software system.

To perform fault quintessence for a specific software, a professional may enhance and customize the information based on the domain of the software and the technologies employed in its development.

Table 8 Study objectives, outcomes, and future scopes

S. No	Paper	Study objectives	Study outcomes	Future scope
1	2010 [14]	A novel model is developed for predicting fault-prone software modules prior to testing. The model turns the information into fuzzy rules by utilizing a Decision Tree built with the ID3 method	The model developed in the study demonstrated the ability to predict with accuracy of 87.37%	Future research directions may involve conducting an analysis of testing efforts which can be further saved by the new model
2	2011 [15]	The goal of the study is to assess high-performance PNN and SVM-based fault predictions	The PNN offers the highest accurate prediction performance for the majority of datasets, according to the results	To gain a deeper knowledge of how various defect prediction models function, the study can be expanded to include other classes of neural networks as well as other datasets
3	2012 [16]	The minority class, which consists of faulty modules, was oversampled by the researchers using the available fault content to examine more important concerns than the majority class, which consists of faultless modules. Using CK metrics, they created three different classifiers	The study found that using the fault content method improved classifiers' performance without appending new data or information to original datasets. The method outperformed SMOTE, a well-known oversampling technique, and all classifiers performed poorly on the original data	Future studies will investigate more classifiers and preprocessing techniques to enhance imbalanced data predictions, such as bagging, boosting, cost-sensitive learners, and others
4	2013 [17]	The usefulness of the Markov blanket approach for feature selection was investigated, and 15 different BN models were evaluated against other well-known predictive modeling techniques	The study concludes that BN classifiers rather than the NB classifier can be used to construct networks with fewer nodes	The study suggests that future research could explore these information sources using Bayesian network learners to gain important insights

(continued)

Table 8 (continued)

S. No	Paper	Study objectives	Study outcomes	Future scope
5	2013 [18]	In order to generate guidelines for identifying defective or fault-prone modules, the study proposes a descriptive method to defect prediction utilizing the subgroup identification algorithms. This approach offers simple rules that practitioners can easily apply and eliminates the need for preprocessing techniques	The suggested method creates guidelines for testing efforts to raise the standards of software development projects. Metrics of defective modules are identified by the rules, along with their threshold values and correlations between them. They give a dataset specification, which makes them simple to understand and apply	The study suggests that there is a need for further investigation of SD algorithms and related factors such as datasets, quality measures, and imbalanced data issues in feature selection and evaluation methods for defect prediction datasets
6	2014 [19]	The examination and comparison of six (ML) methods for forecasting software problems are done in this paper. To ascertain the connection between static code metrics and error proneness, empirical validation is used	The study demonstrates how well machine learning techniques can foresee software flaws. Results show that the DT method's AUC for the model is 0.865 for the AR1 dataset and 0.948 for the AR6 dataset, respectively. Compared to logistic regression and other machine learning techniques, the Decision Tree approach performs better	To produce results that are generalizable, replicated experiments using sophisticated software should be carried out. The application of evolutionary or hybrid evolutionary algorithms to fault prediction may be explored in future studies
7	2014 [20]	The proposed model for defect or error prediction aims to identify faulty software modules and improve software quality by using relational association rules' mining to predict module defects. This involves discovering relational association rules, which extend ordinal association rules by describing numerical orderings between attributes	When compared to other defect detecting models, the DPRAR approach performs well. A promising method for defect discovery is relational association rule mining and the DPRAR method can detect defective software entities if the right representation is given	Future research will investigate different software metric relationships using relational association rules, along with analyzing the impact of rule length and confidence on classification task accuracy

(continued)

Table 8 (continued)

S. No	Paper	Study objectives	Study outcomes	Future scope
8	2015 [21]	The article describes how to use an ANFI system to forecast software errors. The effectiveness of ANFIS, artificial neural networks, and support vector machine models is also evaluated	The study reports an achieved performance of 0.8573 by ANFIS with three parameters, indicating its suitability for software fault prediction. ANFIS provides a direct approach for handling data vagueness compared to machine learning methods, making domain expert knowledge valuable in the learning process	To obtain generalized results, it is recommended that further research can be conducted on large-sized and diverse software projects
9	2017 [22]	This study created a framework for automatically extracting fuzzy rules that are understandable by humans to identify and categorize software flaws. The framework simultaneously recognizes pertinent fault attributes and builds fuzzy rules using them	In terms of performance, studies show that the fuzzy rule-based classification method suggested in the article beats the C4.5, random forest learner, and Naive Bayes classifier models. This method is also favored over other opaque models since it offers more clarity into the reasons influencing software errors	Examining the effectiveness of the fuzzy rule-based system for cross-company fault prediction and repeating the study for fault prediction with open-source projects are two interesting directions for future research
10	2018 [23]	DFCM is presented in this paper. The approach combines feature compression techniques with semi-supervised DFCM clustering	The model evaluation suggests that the combination of deep multi-clusters and techniques such as feature compression yields superior performance, as it allows for the identification and consolidation of critical information in data features	In future work, the approach will be extended by investigating the interrelations between various feature techniques, studying the influence of characteristics of multiple clustering evaluation on classification accuracy, and conducting empirical studies with other classification methods to validate the approach

(continued)

Table 8 (continued)

S. No	Paper	Study objectives	Study outcomes	Future scope
11	2019 [24]	The study offers a paradigm for forecasting software flaws in internal and external software initiatives. With new projects serving as testing sets and older projects or project versions serving as training sets, the framework is utilized to find faults through inter-version and inter-project evaluation	The results of the investigation showed that the framework had much higher inter-project and inter-version evaluation AUC and GM values and a lower error rate when compared to the other classifiers	In order to improve the feature selection stage and performance of inter-version and inter-project fault prediction, future research may use an optimization method to generalize the feature selection process regardless of the dataset specification. To obtain threshold boundaries dynamically, a measure could be employed. Using optimization techniques, it is feasible to create both linear and nonlinear rules
12	2020 [25]	Three different classification models Naive Bayes, logistic regression, and Decision Tree are used in the studies with the help of benchmark fault datasets	Precision is 0.995. The AUC value for decorator was the highest and that is 0.986	In order to generalize the study's findings, additional research will assess ensemble approaches for fault datasets collected from other software systems
13	2021 [26]	The study introduced an improvement to the WOA by integrating a single-point crossover method. The proposed approach aims to enhance WOA's exploration process to prevent it from getting stuck in local optima. The experiment utilized five selection methods	The proposed approach in the study was found to outperform the Whale Optimization Algorithm along with other state-of-the-art methods, leading to an added overall performance of the ML classifier	Future studies can evaluate the performance of the proposed method utilizing a range of optimization issues from practical applications, such as the travel salesman problem, the scheduling problem in education, and the prediction issue for student performance
14	2022 [27]	The study proposes a novel approach for predicting software faults by combining the SSA along with a BNN. The approach aims to enhance the prediction accuracy of BPNN parameters by utilizing an SSA optimizer, which is referred to as SSA-BPNN	The results of the study indicate that used algorithms perform better than the conventional BPNN for all datasets. Furthermore, the proposed approach exhibits superior performance compared to several state-of-the-art methods	One limitation of the proposed method, as noted by the authors, is its high computational cost over the most of the datasets. Hence, new strategy can be developed

(continued)

Table 8 (continued)

S. No	Paper	Study objectives	Study outcomes	Future scope
15	2022 [28]	The goal of the project was to create and assess fault prediction models. They suggested a three-phased methodological framework: metrics identification, testing with ML classifiers and ensemble construction, and evaluation of performance along with cost sensitivity	According to the study, ensemble-based advanced fault prediction models have improved cost sensitivity and predictive ability, with a median <i>F</i> -score that ranges from 76.50 to 87.34% and ROC (AUC) that ranges from 77.09 to 84.05%. The authors also used non-parametric tests to check the classifiers' statistical significance	The study's future work includes duplicating studies utilizing more open-source and cross-project datasets in order to strengthen the generalization of results and bolster the study's case. A potential direction for future research is to investigate various defect prediction algorithms that employ diverse metric data from different projects
16	2023 [29]	In this study, predictions of software faults based on machine learning and deep learning were made using the SFP XP-TDD dataset. Three separate software projects were employed to build this dataset. On the basis of this dataset, five commonly employed classifiers from the literature were trained and put to the test	The study employed three separate software failure prediction datasets to statistically demonstrate that DL algorithms outperform ML techniques in datasets with large sample sizes. The experimental findings from a layer of RNNs-based model were also provided	In order to incorporate transfer learning into the creative work produced using a new hybrid technology, future study will leverage Java Doc papers to develop a new dataset

References

1. Juneja S, Singh GB, Sidhu BK (2023) Programming tool for assembling software bugs through open source repository. *J Data Acquis Process* 37(5):1522
2. Thung F, Lo D, Jiang L (2012) Automatic defect categorization. https://ink.library.smu.edu.sg/sis_research
3. Patil S, Ravindran B (2020) Predicting software defect type using concept-based classification. *Empir Softw Eng* 25(2):1341–1378. <https://doi.org/10.1007/s10664-019-09779-6>
4. ODC Chapter on orthogonal defect classification scheme
5. ODC (2013) Orthogonal Defect Classification v. 5.2 Extensions for Defects in GUI, User Documentation, Build and National Language Support (NLS). To be used in conjunction with Orthogonal Defect Classification v. 5.2 for Design and Code
6. Thung F, Le XBD, Lo D (2015) Active semi-supervised defect categorization. In: IEEE International Conference on Program Comprehension. IEEE Computer Society, pp 60–70. <https://doi.org/10.1109/ICPC.2015.15>
7. Engineering Standards Committee of the IEEE Computer Society (2009) IEEE Std 1044-2009 (Revision of IEEE Std 1044-1993), IEEE Standard Classification for Software Anomalies
8. Herraiz I et al (2008) Towards a simplification of the bug report form in eclipse. In: Proceedings of the 2008 international working conference on mining software repositories
9. Hamill M, Goseva-Popstojanova K (2015) Exploring fault types, detection activities, and failure severity in an evolving safety-critical software system. *Software Qual J* 23:229–265
10. Engineering Standards Committee of the IEEE Computer Society (2010) ISO/IEC/IEEE 24765-2010(E), Systems and software engineering—Vocabulary. www.iso.org
11. Hirsch T, Hofer B (2022) Using textual bug reports to predict the fault category of software bugs. *Array* 15:100189. <https://doi.org/10.1016/j.array.2022.100189>
12. Mariani L (2003) A fault taxonomy for component-based software. *Electron Notes Theoret Comput Sci* 82(6):55–65
13. Tan L, Liu C, Li Z, Wang X, Zhou Y, Zhai C (2014) Bug characteristics in open source software. *Empir Softw Eng* 19(6):1665–1705. <https://doi.org/10.1007/s10664-013-9258-8>
14. Kumar Pandey A, Kumar Goyal N (2012) Predicting fault-prone software module using data mining technique and fuzzy logic. *Int J Comput Commun Technol* 3(1):56–63. <https://doi.org/10.47893/IJCCT.2012.1105>
15. Al-Jamimi HA, Ghouti L (2011) Efficient prediction of software fault proneness modules using support vector machines and probabilistic neural networks. In: 2011 5th Malaysian Conference in Software Engineering, MySEC 2011, pp 251–256. <https://doi.org/10.1109/MYSEC.2011.6140679>
16. Shatnawi R (2012) Improving software fault-prediction for imbalanced data. In: 2012 International Conference on Innovations in Information Technology, IIT 2012, pp 54–59. <https://doi.org/10.1109/INNOVATIONS.2012.6207774>
17. Dejaeger K, Verbraken T, Baesens B (2013) Toward comprehensible software fault prediction models using bayesian network classifiers. *IEEE Trans Softw Eng* 39(2):237–257. <https://doi.org/10.1109/TSE.2012.20>
18. Rodriguez D, Ruiz R, Riquelme JC, Harrison R (2013) A study of subgroup discovery approaches for defect prediction. *Inf Softw Technol* 55(10):1810–1822. <https://doi.org/10.1016/J.INFSOF.2013.05.002>
19. Malhotra R (2014) Comparative analysis of statistical and machine learning methods for predicting faulty modules. *Appl Soft Comput* 21:286–297. <https://doi.org/10.1016/J.ASOC.2014.03.032>
20. Czibula G, Marian Z, Czibula IG (2014) Software defect prediction using relational association rule mining. *Inf Sci (NY)* 264:260–278. <https://doi.org/10.1016/J.INS.2013.12.031>
21. Erturk E, Sezer EA (2015) A comparison of some soft computing methods for software fault prediction. *Expert Syst Appl* 42(4):1872–1879. <https://doi.org/10.1016/J.ESWA.2014.10.025>

22. Singh P, Pal NR, Verma S, Vyas OP (2017) Fuzzy Rule-based approach for software fault prediction. *IEEE Trans Syst Man Cybern Syst* 47(5):826–837. <https://doi.org/10.1109/TSMC.2016.2521840>
23. Arshad A, Riaz S, Jiao L, Murthy A (2018) Semi-supervised deep fuzzy C-mean clustering for software fault prediction. *IEEE Access* 6:25675–25685. <https://doi.org/10.1109/ACCESS.2018.2835304>
24. Juneja K (2019) A fuzzy-filtered neuro-fuzzy framework for software fault prediction for inter-version and inter-project evaluation. *Appl Soft Comput* 77:696–713. <https://doi.org/10.1016/J.ASOC.2019.02.008>
25. Rathore SS, Kumar S (2021) An empirical study of ensemble techniques for software fault prediction. *Appl Intell* 51(6):3615–3644. <https://doi.org/10.1007/S10489-020-01935-6/TABLES/12>
26. Hassouneh Y, Turabieh H, Thaher T, Tumar I, Chantar H, Too J (2021) Boosted whale optimization algorithm with natural selection operators for software fault prediction. *IEEE Access* 9:14239–14258. <https://doi.org/10.1109/ACCESS.2021.3052149>
27. Kassaymeh S, Abdullah S, Al-Betar MA, Alweshah M (2022) Salp swarm optimizer for modeling the software fault prediction problem. *J King Saud Univ Comput Inform Sci* 34(6):3365–3378. <https://doi.org/10.1016/J.JKSUCI.2021.01.015>
28. Sharma P, Sangal AL (2022) Examining the predictive capability of advanced software fault prediction models—an experimental investigation using combination metrics. *e-Inform Softw Eng J* 16(1):220104. doi: <https://doi.org/10.37190/E-INF220104>.
29. Borandag E (2023) Software fault prediction using an RNN-based deep learning approach and ensemble machine learning techniques. *Appl Sci* 13(3):1639. <https://doi.org/10.3390/APP13031639>

Hybrid Cryptography and Steganography Method to Provide Safe Data Transmission in IoT



Atrayee Majumder Ray, Sabyasachi Pramanik, Biplab Das,
and Ashish Khanna

Abstract IoT, the upcoming phase of the information innovation, refers to the condition in which millions and millions of people and appliances are associated to improve the interchange of enormous volumes of data from various regions, necessitating the resultant requirement for intelligent data aggregation. But along with the many advantages and uses it offers, the Internet of Things also introduces a fresh difficulty element in the form of several intrinsic inconveniences, particularly security issues during data transmission phases, which mainly affect data confidentiality and integrity aspects. A combined technique of insubstantial cryptography and steganography (LSB Replacement) approach is employed for data communication between IoT tool and home server and following integrated technique of cryptography and steganography (suggested MSB–LSB Replacement approach) for data transmission stages between home server and IoT device in this paper’s security scheme, which addresses both the aforementioned issues.

Keywords IoT · Decryption · Steganography · MSB · LSB

1 Introduction

In general, the term IoT denotes the situations in which network connection and computation capacity are extended to people, wirelessly recognizable objects, sensors, sensor embedded-intelligent small devices, and daily objects (not typically regarded as computers), facilitating these to create, interchange, and use data with

A. M. Ray
Netaji Subhash Engineering College, Ranabhatia, India

S. Pramanik (✉) · B. Das
Haldia Institute of Technology, Haldia, India
e-mail: sabyalnt@gmail.com

A. Khanna
Maharaja Agrasen Institute of Technology, New Delhi, India

little to no human interference. Lawful experts choose to see “Things” as an integrated combination of H/W, S/W, data, and services. The Internet of Things assures to expand “everywhere, anyway, every time” computation to include “whatever, anybody, and any assistance,” that reflects the emerging tendency of how individuals and companies are probable to integrate network connectivity and the Internet into their day-to-day existence to benefit from a completely associated “intelligent” world. The underlying fact that the IoT comprises a vast range of ubiquitous appliances with sensing capabilities while being restricted to the Internet illustrates the situation that the interactions between items and humans are intricately entwined. Moreover, according to Huawei Internet Services projection, the number of IoT devices deployed will quadruple (to 58 billion devices) by 2025. IoT applications may be found in a variety of industries, including smart health, smart vehicles, intelligent homes, environmental tracking, precision agriculture, and logistics.

The thorny problem of IoT security is a huge cause of concern from its establishment, denoting IoT insecurity as a top 4 security perils in 2023. Thus, a collective approach to reliability is required to provide tolerable and efficient answers to IoT confidentiality [1] concerns for maximizing its many privileges while using IoT in universal growth.

Considering that IoT device security is not a binary choice between safe and insecure which is crucial given the growth of the IoT. Instead, it would be helpful to think of IoT security as a range of gadgets’ vulnerabilities. IoT shows a diversity of newer possibilities to attack possible security flaws as we link further items to the Internet. A major security analysis with IoT devices is the conventional difficulty of security breach caused by the confession of delicate data. If an individual does not think that his linked appliances, like smart TVs, smart phones, and various home accessories, and their data, are legitimately protected for causing exploitation or harm, the ensuing decrease of assurance would make them reluctant to accept IoT worldwide. The inherent decentralized and diversified quality of IoT objects combined with weekly protected IoT tools can pose a risk in cyber-attacks like blocking, eavesdropping, and moderation of private information, where malignant persons may re-program a gadget or make it to crash one time for getting access to the coordinated IoT tools ensuing protection violation.

We provide a security model in this work that is based on an integrated technique of steganography and cryptography approaches for effectively addressing the problem of data security loss in IoT. Steganography is the practice of hiding sensitive information by enclosing it with a cover picture to prevent hackers from discovering the sent data. The study of creating ciphers to encrypt communications and information in a manner by which solely approved and designated transmitting parties can read and understand is known as cryptography, on the other hand. Here, the sensor devices are called as Internet of Things devices as they are in control of collecting data from the surroundings in which they are decentralized and transferring it to the home server.

The aforementioned secured approach may be separated into two categories listed below:

1. As we are acquainted of the evidence that IoT devices have inherent H/W limitations like restricted memory, battery power limitations, and lower computing capacities, the integration of steganography (LSB Replacement method) and a insubstantial cryptography [2] method (like XOR operation) is suggested in the data communication step between IoT device and the authentication server.
2. Since there are no resource-related restrictions in the data communication step between the home server and the cloud, a combined strategy of steganography (suggested MSB–LSB Replacement approach) and reliable encryption techniques like AES and DES is used.

The aforementioned methodology provides improved two-level securities as a result, solving the problem of data privacy loss in data transmission stages in an intelligent IoT context. According to the two-layer security pattern described before, we recommend adopting an integrated strategy that makes use of both cryptography and steganography [3–5] methods for safe data transmission in smart environments. The reason for this is because if just encryption schemes are used, then sensitive information still exists but is only accessible in ciphers, which might encourage criminals to decrypt it when enough time or processing power is given. Therefore, it is clear that the use of cryptography cannot hide the presence of private information or data from prying eyes or attackers. Since the steganography method entirely conceals the data by hiding it in a host picture, it may be added to the Cryptography approach to improve safe and confidential data transmission in Internet of Things. As a result of using the steganography system in the suggested security approach, assailants would be completely unaware that transmitted information is present in the IoT environment, improving security [6].

2 Related Work

Since IoT sensors are extensively used in a variety of unsafe and susceptible locations, ensuring the secure transfer of data and information between IoT devices must be given top attention, and numerous security procedures have been recommended. The use of cryptography is one such measure. Use of cryptography is advised in IoT because to the built-in limitations of IoT devices, like their limited refining power and memory. Since these algorithms need less code and RAM, lightweight cryptography techniques excel in limited settings where RFID [7] tags, sensors, and medical gadgets are widely employed. Additionally, if it is used in the IoT, lightweight cryptographic techniques may improve end-to-end communications and operations for lower resource devices. The fact that these algorithms work with fewer encryption rounds and shorter key lengths is a major cause for worry. In general, there are two types of lightweight cryptography algorithms: symmetric and asymmetric. In symmetric encryption, both communication events employ a common key that is

used for both encryption and decryption. SEA, XOR, AES [8], and other examples of lightweight symmetric encryption are used in security systems. To guard against passive attacks like eavesdropping, a lightweight cryptographic protocol dependent on XOR operation is suggested for RFID tags and readers.

Since wireless communication is most common among IoT devices, computational intelligence was recommended to produce the Wireless Intrusion Detection System that provides numerous advantages like better latency, its capacity to learn preferences, higher computational speed, and better adaptability in changing altering surroundings. The receiver must preserve a minimal growth in acquired power of 3 dB in order for this approach to work well, which is one of its two major limitations. Another drawback of the aforementioned method is that although the receiver preserves the minimal rise in acquired power, this framework will be ineffective if the transmitter's signal comes at 20 dB from the interceptor's recipient location. Furthermore, a number of researchers have previously successfully employed steganography to securely hide data on low-processing-power appliances like embedded systems and smart phones.

3 The Proposed Scheme

In this paper's Fig. 1, we assume a situation in which an Internet of Things (IoT) sensor node collects data from its deployment environment and transmits it to a server through LAN for data authentication. Following successful authentication, the data is subsequently sent to WAN (for example, to cloud servers) for storage and additional calculations as required by the IoT [9] implementation. The key issue here is the initiation of any cyber assault by malevolent persons, like packet sniffing or eavesdropping, especially in the LAN connection which compromises of the privacy of communicated data. For instance, in Fig. 1, the sensor data includes sensitive data that may be effectively retrieved by an eavesdropping attack, allowing the offender to identify themselves with the authentication server.

Our suggested plan, which ensures the dependability of communicated sensitive data/information utilizing IoT device, is shown in Fig. 2.

Below is a detailed explanation of the steps:

Step 1: Toward sensor's (IoT device's) location:

- (a) Before transferring data straight to the home server, the detected data is encrypted utilizing a quick encryption method (such the XOR process).
- (b) After that, the Message Digest 5 (MD5) method is used to generate the message digest of the detected data.
- (c) Finally, utilizing the steganography approach, the encrypted data and calculated digest are hidden in a haphazard-chosen cover image, resulting in a stego-image that is thereby sent to the home server for validation purposes.

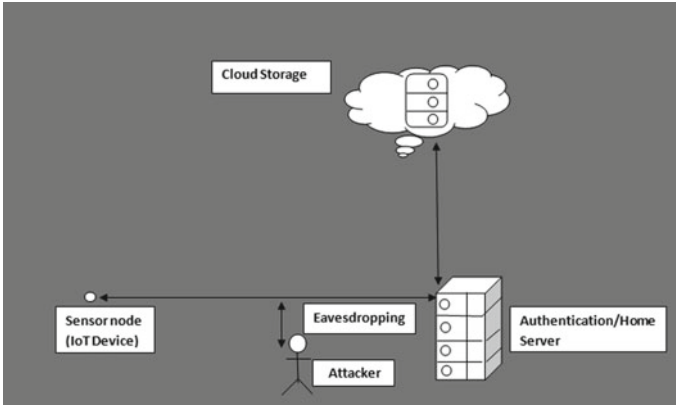


Fig. 1 Current IoT situation

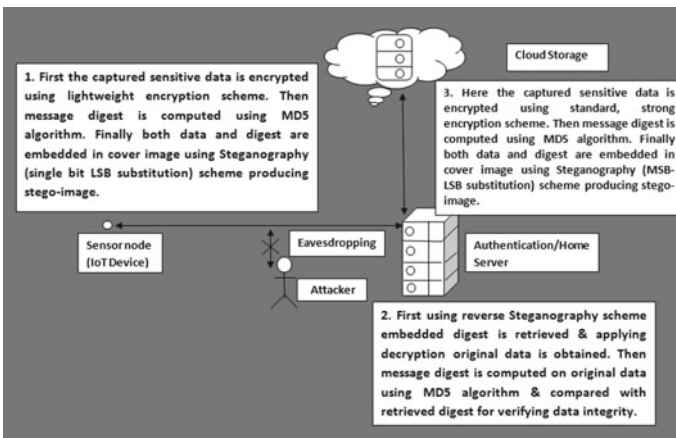


Fig. 2 Proposed IoT security model

The aforementioned steganography technique makes use of the straightforward LSB replacement technique, which is a fundamental steganography approach. It involves replacing the LSB [10] of every cover pixel with the message bit. The straightforward requirement for substitution method is that only replacement takes place when the message bit is 1, the similar LSB of the cover pixel where the message bit for hiding is 0, or when the message bit is 0, and the similar LSB of the cover pixel is 1. The LSB of the cover pixel is otherwise unaltered. Therefore, there is a 50% chance that each cover pixel will flip. This system is well-liked not only because it is extremely straightforward but also because it has two additional benefits, the first of which is that it has a lower computational cost than conventional typical cryptographic algorithms and the second of which is that it can transport data safely.

Step 2: At home or authentication server location:

- (a) By utilizing reverse steganography, the home server initially retrieves the encrypted data version and the embedded message digest.
- (b) In order to recover the original sensed data, it next decrypts the encrypted data version.
- (c) Lastly, after calculating the message digest utilizing the MD5 [10] technique, it compares the freshly calculated digest with the acquired digest; in case a similarity is found, then data integrity is retained and fruitful validation takes place; otherwise, data is deleted by it.

Step 3: Similar to the strategy described above in step 1, the same process is repeated here in the data communication step between the home server and the clouds by directly substituting the suggested lightweight encryption strategy earlier with a robust encryption algorithm like AES/DES and the simplest LSB replacement scheme mentioned earlier with the recently suggested MSB–LSB replacement scheme. Here, the home server serves as a central point of authentication for the LAN network’s Internet-based data transfer to cloud servers for safe storage and any additional calculations necessary to meet the demands of the IoT application being used.

As a result, in a smart IoT environment, we provide a two-level security scheme which addresses the issue of data privacy and furthermore ensures data integrity when data is being transmitted moreover between IoT appliances and the server or within the IoT appliances via Internet connectivity.

4 Proposed MSB–LSB Algorithm

A well-known method called single-bit LSB replacement can only replace a lone bit of data in every carrier pixel (if the carrier is an image), and there do not exist any possibility of inserting additional data bits within the cover media. Therefore, the authors developed an approach that can only embed two data bits and uses MSB and LSB to replace. Thus, the benefit is that the picture quality is barely reduced and that it is equivalent to the basic LSB replacement method (level of replacement is 1). However, the authors may hide up to 2 message bits, while the simple LSB replacement technique can only hide 1.

A. Terminology:

- Cover image pixel’s LSB is L .
- The cover image’s pixel’s MSB is M .
- The first bit of the message is x -1Bit.
- The second bit is x -2Bit.

B. Short forms:

- Message: Me.

C. Hiding Algorithm:

Step 1:

```

if ( $M$  is 0)
then (LSB is 1 and  $x$ -Bit-1 is 0)
replace  $L$  by  $x$ -1Bit
otherwise, if ( $L$  is 0 and  $x$ -1Bit is 1)
substitute  $L$  by  $x$ -1Bit
otherwise, if ( $L$  is  $x$ -1Bit)
keep  $L$ // Embed 1 Me bit.

```

Step 2:

```

otherwise, if ( $M$  is 1 and  $x$ -1Bit is 1)
if ( $L$  is 1 and  $x$ -1Bit is 0)
and if ( $L$  is 1 and  $x$ -2Bit is 0)
keep the  $M$  and substitute the  $L$  with the  $x$ -2Bit//hide 2 Me bits
else, skip the pixel// No Hiding
Otherwise, if ( $L$  is 0 and  $x$ -2Bit is 0)
keep both the  $M$  and  $L$ // Hide two Me bits.
Otherwise, if ( $L$  is 0 and  $x$ -2Bit is 1) keep the  $M$ ;
substitute the  $L$  to  $x$ -2Bit// do not hide

```

Step 3:

```

otherwise, if ( $M$  is 1 and  $x$ -1Bit is 0)
set  $L$  to 1 and keep  $M$ // No Embedding

```

D. Retrieving Method:

```

Step 1: if ( $M$  is 0)// Obtain 1 Me bit.
Step 2: otherwise if ( $M$  is 1 and  $L$  is 0)// Retrieve 2 Me bits
Step 3: Alternately, if ( $M$  is 1 and  $L$  is 1)// No Recovery

```

5 Results

Our experiment was carried out using MATLAB R2013a. Following the phases shown in step 1 of our suggested approach, which is proposed in Sect. 3, the authors initially encrypted the data utilizing a lightweight encryption technique (XOR operation), calculated its digest utilizing the MD5 approach, and thereby inserted both encrypted message and digest in a haphazardly chosen cover image using the simplest Least Significant Bit replacement steganography approach. By first encrypting the data using the DES standard encryption technique, computing its digest utilizing the MD5 approach, and thereby embedding both encrypted data and digest in a second haphazardly chosen cover image utilizing our suggested (MSB–LSB replacement technique) steganography technique, the authors also have strongly completed all

the steps (step 3 of the suggested approach shown in Sect. 3). In most situations, our suggested MSB–LSB replacement technique outperforms its straightforward LSB replacement equivalent.

In this experiment, the authors took into account a dataset with characters varying from 1 to 100. Each dataset was divided into 100 different permutations, and the means of every such dataset were then calculated. One of the dataset’s computed mean may be quantitatively expressed as,

$$\sum_{j=1}^{100} X_j,$$

where X_j = permutations of the individual datasets.

Below is a description of the operations’ consecutive result:

- A. Data transmission operations between an IoT device and a home or authentication server.
 1. Suppose we take the information as IoT data.
 2. The encrypted form of the aforementioned data acquired after utilizing an encryption technique (XOR operation) is: @\$ K(yP) (‘ho
 3. MD5 message digest calculated for the following data: CC0B6E56EEEC9E56207B5FB8241.
 4. Figure 3 shows the original image and the stego-image (which includes the encrypted message, digest, and key).
- B. Data transmission operations between the cloud servers and the household/authentication server.
 1. Suppose we take the information as IoT data.
 2. The encrypted form of the aforementioned data produced after using the standard encryption technique (DES) is: _ û1 -j—caü\$ r qöa.



Fig. 3 Utilizing the simplest LSB replacement **a** the original image and **b** the stego-image



Fig. 4 Host image and the stego-image created utilizing the MSB–LSB replacement

3. The message digest calculated by the MD5 method is: CC0B6E56EEEECC9E56207B5FB8241.
 4. Figure 4 shows the original picture and the stego-image (which includes the encrypted message, digest along with the key) after utilizing the suggested MSB–LSB replacement steganography approach.
- C. Comparison of our suggested MSB–LSB replacement steganography algorithms and straightforward LSB replacement.

The quantity of pixels required embedding the predefined dataset using both of the suggested MSB–LSB replacement technique and a simple LSB replacement technique is shown in Fig. 5. Based on analysis, it may be concluded that the suggested MSB–LSB replacement technique needs significantly fewer pixels to embed in the vast majority of cases (99%). Figures 6 and 7 make it abundantly clear that the suggested MSB–LSB replacement scheme significantly reduces the quantity of pixels affected after hiding when compared to its basic LSB replacement counterpart.

We can easily draw the conclusion that the suggested MSB–LSB replacement technique of steganography achieves superior performance than the simple LSB replacement technique in the majority of cases when looking at both of the earlier mentioned aspects, i.e., the quantity of pixels needed for hiding and the quantity of pixels changed after hiding.

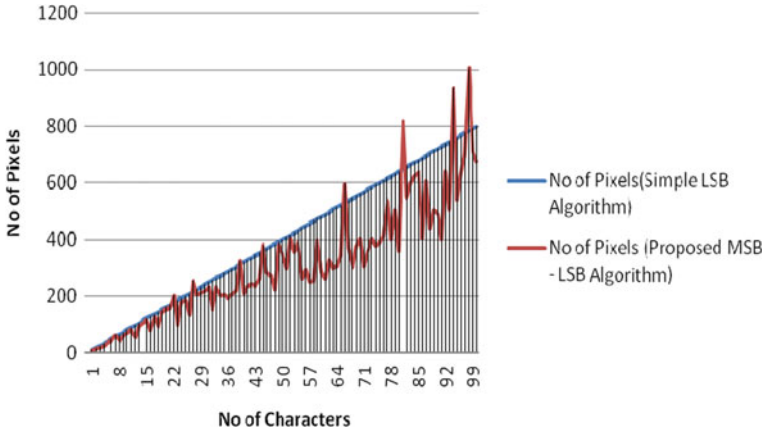


Fig. 5 Required number of pixels for embedding

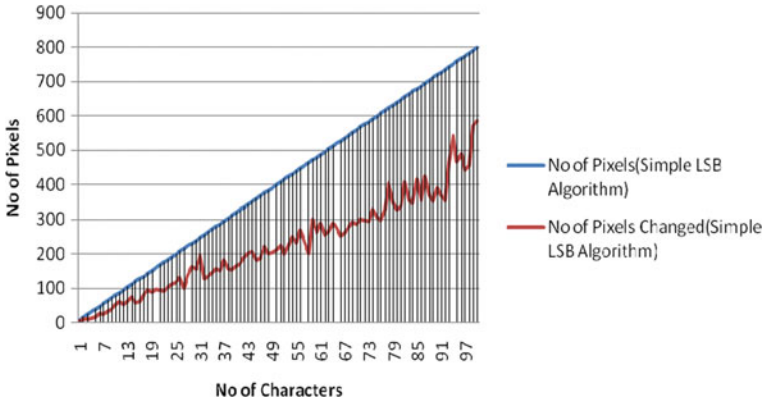


Fig. 6 Total pixels changed by the simple LSB algorithm

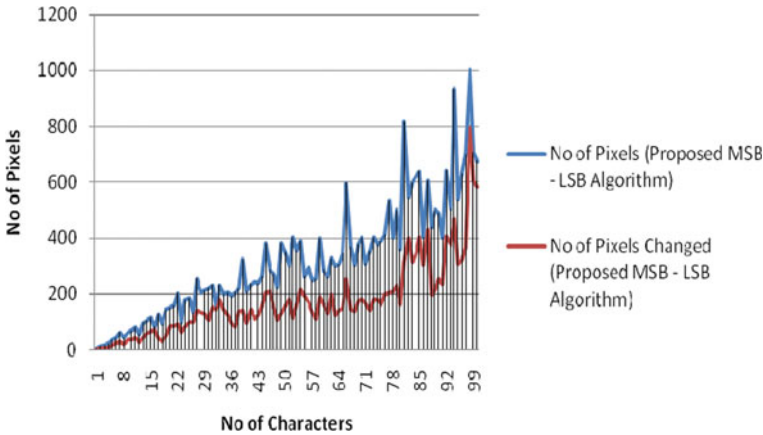


Fig. 7 Total pixels changed by the suggested MSB–LSB method

6 Conclusion

The IoT has been extensively adopted and utilized to solve few important problems in connection with worldwide evolution as a result of the tremendous advancements in the field of WSN and mobile computing technologies. In case the threats and challenges associated to IoT security may be adequately operated and regulated sensibly, especially the fundamental details of data privacy and integrity features when data is actually transmitted from IoT appliances to the servers through Internet, can the commercialized success of IoT implementations be completely perceived on a worldwide extent? For solving the data confidentiality concerns for data transmission steps in IoT appliances, we propose two-layer security architecture in this work depending on a consolidated method of steganography and cryptography technology. Additionally, we conducted a comparison of the performance of elementary LSB Replacement and the suggested newer MSB–LSB Replacement steganography plans, and the results of the focused experiment showed that, in most cases, the suggested MSB–LSB Replacement pattern outperformed its traditional LSB Replacement equivalence.

References

1. Khan LS, Khan M, Hazzazi MM et al (2023) A novel combination of information confidentiality and data hiding mechanism. *Multimed Tools Appl* 82:6917–6941. <https://doi.org/10.1007/s11042-022-13623-3>
2. Hhan M, Morimae T, Yamakawa T (2023) From the hardness of detecting superpositions to cryptography: quantum public key encryption and commitments. In: Hazay C, Stam M (eds) *Advances in Cryptology—EUROCRYPT 2023*. EUROCRYPT 2023. Lecture Notes in

- Computer Science, vol 14004. Springer, Cham. https://doi.org/10.1007/978-3-031-30545-0_22
3. Pramanik S (2023) An adaptive image steganography approach depending on integer wavelet transform and genetic algorithm. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-023-14505-y>
 4. Gupta A, Verma A, Pramanik S (2022) Security aspects in advanced image processing techniques for COVID-19. In: Pramanik S, Sharma A, Bhatia S, Le DN (eds) *An interdisciplinary approach to modern network security*. CRC Press, Boca Raton
 5. Kaushik D, Garg M, Annu, Gupta A, Pramanik S (2022) Application of machine learning and deep learning in cyber security: an innovative approach. In: Ghonge M, Pramanik S, Mangrulkar R, Le DN (eds) *Cybersecurity and digital forensics: challenges and future trends*. Wiley, New York. <https://doi.org/10.1002/9781119795667.ch12>
 6. Bandyopadhyay S, Goyal V, Dutta S, Pramanik S, Sherazi HHR (2021) Unseen to seen by digital steganography. In: Pramanik S, Ghonge MM, Ravi R, Cengiz K (eds) *Multidisciplinary approach to modern digital steganography*. IGI Global, Hershey, pp 1–28. <https://doi.org/10.4018/978-1-7998-7160-6.ch001>
 7. Kumar S, Banka H, Kaushik B (2023) Ultra-lightweight blockchain-enabled RFID authentication protocol for supply chain in the domain of 5G mobile edge computing. *Wireless Netw*. <https://doi.org/10.1007/s11276-023-03234-7>
 8. Pandey BK, Pandey D, Nassa VK, George AS, Pramanik S, Dadheech P (2023) Applications for the text extraction method of complex degraded images, in the impact of thrust technologies on image processing. Nova Publishers, New York
 9. Sinha M, Chacko E, Makhija P, Pramanik S (2021) Energy efficient smart cities with green IoT. In: Chakrabarty C (ed) *Green technological innovation for sustainable smart societies: post pandemic era*. Springer, Berlin. https://doi.org/10.1007/978-3-030-73295-0_16
 10. Wang N, Yu H, Guo Z, Lei H (2022) Firmware security verification method of distance learning terminal based on MD5 algorithm. In: Fu W, Sun G (eds) *e-Learning, e-Education, and Online Training*. eLEOT 2022. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering, vol 454. Springer, Cham. https://doi.org/10.1007/978-3-031-21164-5_34

Assessing the Efficacy of Different BERT Variants for Distinguishing Types of Cyberbullying on Twitter



Ashwin Prajeeth, Binav Gautam, and Garima Chhikara

Abstract BERT is a highly developed language model that has excelled in a wide range of tasks, including question-answering and language comprehension. It now outperforms both the performance of older state-of-the-art computer models and human intelligence. Due to its exceptional capabilities, a large number of businesses, academic institutions, and divisions of Google are actively developing the BERT model architecture through supervised training, with the goal of either enhancing its effectiveness or customizing it for particular tasks by pre-training it with specific contextual representations. In this paper, we classify the category of cyberbullying in each tweet posted on Twitter taken into our dataset. Our research aims to test out different variations of BERT-ALBERT, RoBERTa, and DistilBERT and extrapolate the efficiency of each model concerning the multi-label classification of tweets.

Keywords NLP · BERT · Multi-label classification of tweets

1 Introduction

Social media has become ingrained in our lives. The ubiquity of social media has come about extremely quickly. The technology is based on the Internet, allowing rapid communication between users worldwide through ideas, thoughts, and information through virtual networks and communities. Users can share data from simple text-based messages to media like photos and videos. Any social media platform can

A. Prajeeth (✉) · B. Gautam · G. Chhikara
Department of Computer Science and Engineering, Delhi Technological University,
New Delhi, Delhi 110042, India
e-mail: ashwinkurumkulamprajeeth_2k19co92@dtu.ac.in

B. Gautam
e-mail: binavgautam_2k19co103@dtu.ac.in

G. Chhikara
e-mail: garimachhikara@dtu.ac.in

be accessed through nodes on the Internet like computers, tablets, smartphones, or smartwatches.

According to a study [1] on social media usage, there are estimated to be around 4.5 billion daily active users on various social platforms. Out of this, Asia–Pacific had the most significant social media users, with over 2.4 billion users. This was followed by North America, with over 370 million users, and Europe, with over 700 million users [2]. According to a report, Facebook continues to be the most widely used social media platform worldwide, with more than 2.7 billion active users every month, as of January 2021. The report also highlights other famous platforms like YouTube (with 2.3 billion users), WhatsApp (with 2 billion users), Instagram (with 1.2 billion users), and TikTok (with 689 million users) [2].

Each platform usually aims to fulfill a specific role, such as LinkedIn being used predominantly for job hunts and networking, Twitter is used to get updates on various events worldwide, and Facebook and Instagram are being used to share one’s essential life updates linked to their desired community. Since Twitter is one of the most used social media apps that allows its users to share information with everyone on the platform, it served as the perfect model for this research to find important markers of cyberbullying based on users’ tweets.

Cyberbullying has become one of the most prevalent types of bullying as it allows users to post degrading comments anonymously. About 59% of American teenagers have experienced cyberbullying firsthand or have seen it happen on social media [3]. A wide range of age groups is impacted by cyberbullying. Currently, 37% of adult Internet users have encountered online abuse, and 73% of those incidents occurred on social media [4]. The most popular places for cyberbullying are social media websites like Facebook, Instagram, and Twitter [5].

Serious repercussions of cyberbullying might include melancholy, anxiety, and even suicide. According to one research, those who experience cyberbullying are more than twice as likely to attempt suicide than people who do not [6]. As a result of cyberbullying, victims may also have scholastic challenges, such as declining grades or an unwillingness to attend school. Cyberbullying perpetrators may potentially face unpleasant repercussions. They can feel bad about what they did or embarrassed. Cyberbullying can also result in legal repercussions, such as allegations of harassment or defamation. Cyberbullying may also have an impact on observers. They can feel bad for not stepping in, or they might start to worry that they will be the next victim. Bystanders may potentially encounter detrimental emotional outcomes, such as melancholy or anxiety [7].

In this project, we plan on using a Twitter dataset to use the various BERT transformers to measure the efficiency of the models based on various classification scores. This paper is divided into the following sections, Sect. 2 talks about related works in the area of detecting cyberbullying, Sect. 3 talks about the methodology used for this research, Sect. 4 discusses the results, and Sect. 5 elaborates on concluding the paper and future scope.

2 Related Works

Many research articles have been published that use different machine learning algorithms to detect cyberbullying. We found the following insightful. Dadvar and Eckert [8] used DNN to examine current publications' conclusions in this area and verified their conclusions using the same datasets. They expanded the study even further by using the created approaches on a new dataset. The study demonstrated that the DNN models were transferable and adaptable to the new dataset. Islam et al. [9] combined machine learning and natural language processing to create a powerful method for identifying online bullying and abusive remarks. The accuracy level of four machine learning algorithms is examined using two different features: Bag-of-Words (BoW) and term frequency-inverse text frequency (TF-IDF). Islam et al. [9] discovered that TF-IDF performs more accurately than BoW. SVM surpasses the other machine learning algorithms. Ali and Syed [10] draw attention to earlier studies and suggest a method for identifying sarcasm in cyberbullying. Various ML algorithms like Random Forest, SVM, Naïve Bayes, and Logistic Regression were used to conclude that the SVM classifier outperformed the rest. Giumetti and Kowalski [11] review studies evaluating the connection between well-being and online harassment of children and adults through social media. Giumetti and Kowalski [11] also discuss a number of potential social media predictors of cyberbullying, such as indiscriminate posting, time spent on social media, and personality factors. Neelakandan et al. [12] introduce a novel approach for detecting and classifying cyberbullying on social networks using a feature subset selection technique based on deep learning (DL). The proposed technique, called BCO-FSS, is designed to select the most effective feature combination from preprocessed data, leading to significantly improved classification outcomes. The experimental results demonstrate that the FSSDL-CBDC technique outperforms other existing state-of-the-art methods in terms of classification accuracy. Agrawal and Awekar [13] extensively evaluated the efficacy of machine learning (ML) and deep learning (DL) models in detecting cyberbullying using three different real-world datasets: Formspring (12 k posts), Twitter (16 k posts), and Wikipedia (100 k posts). The results indicated that deep neural network (DNN) models with transfer learning outperformed state-of-the-art approaches for all three datasets. The researchers also suggested that integrating supplementary data, such as user profiles and social network details, could potentially enhance the models' performance even further, based on the findings. Al-Garadi et al. [14] review cyberbullying prediction models in detail and highlight the fundamental problems that arise while building such models in SM. This paper offers details on the general procedure for detecting cyberbullying, and more significantly, it summarizes the technique. Hani et al. [15] suggest using supervised machine learning algorithms like Naïve Bayes classifier, decision tree, and SVM to identify and stop online bullying. Using a cyberbullying dataset, the examination of the suggested technique reveals that neural network works better and obtains an accuracy of 92.8%, while SVM reaches 90.3. Wang et al. [16] aimed to evaluate the feasibility of implementing an automated multi-class cyberbullying detection algorithm capable of identifying instances where cyberbullies target

their victims based on six attributes. The research findings indicated that Dynamic Query Expansion is an effective approach to address the class imbalance and recommended its utilization in diverse settings, including social media data mining and natural language processing, to combat cyberbullying. Using a variety of embedding method + classifier model combinations, we perform thorough experiments to develop fine-grained cyberbullying detection, and the results are comparable to earlier binary cyberbullying classification research. Lastly, the results demonstrate that the performance of the proposed Graph Convolutional Network framework SOSNet, which leverages the natural semantic links between tweets, is on par with or better than that of conventional classifiers in this area. Reynolds et al. [17] trained a computer to identify bullying content using the labeled data and machine learning methods offered by the Weka toolkit. The true positives could be distinguished with an accuracy of 78.5% by both a C4.5 decision tree learner and an instance-based learner.

3 Methodology

3.1 Models Used

1. *BERT*: In this project, a pre-existing deep learning model named Bidirectional Encoder Representations from Transformers (BERT) is utilized. BERT is mainly used for natural language processing (NLP) applications. Since its inception in 2018 by Google researchers, it has become one of the most used NLP models [18]. BERT has been pre-trained on a significant quantity of text data by a process known as masked language modeling. This is done by masking a word in one of two phrases shown to BERT during training, and it is asked to guess the term given the context of the second sentence. Contextualized word embeddings are representations of each word in a phrase that consider the words around it and are produced by the BERT model quite well. These embeddings are very good at capturing the meaning of the text because they capture both the syntactic and semantic links between words in a phrase.
2. *ALBERT*: An acronym for A Lite BERT which was also introduced by Google. It is considered to be a version that is of BERT that is computationally less expensive and faster than BERT. ALBERT maintains its accuracy and performance even though it computes a lot less parameters than BERT [19].
3. *RoBERTa*: Facebook (currently known as Meta) introduced a robustly optimized BERT pre-training approach (RoBERTa) [20]. As a result of optimizations and design changes made to the model architecture and tweaking the hyperparameters while the model is being trained, pre-training takes less time than training itself.

4. *DistilBERT*: Hugging Face and the Google AI Language team created DistilBERT, a more compact and quicker version of the original BERT model. It was created to address the original BERT model’s high computational costs and storage needs while still performing very well on tasks requiring natural language understanding.

3.2 Dataset

The dataset used in this study [16] consists of over 47,000 tweets that are categorized as instances of cyberbullying. The dataset has been divided into two classes, each containing 8000 tweets, to ensure balanced data. It should be noted that the dataset was originally obtained from a previously published paper [16].

3.3 Data Preprocessing

On describing the data, we can see the counts and the unique counts of each data type. There are six categories of cyberbullying under which the data has been ordered. These are namely bullying due to young or old age; due to the race one belongs to; bullying toward males, females or LGBTQ members; against one’s religions and beliefs; and other cyberbullying not classified under any of these categories (Fig. 1).

First, we shuffle the data, as the data is sorted categorically. After we shuffle the dataset, we separate it into train and test sets. We follow standard preprocessing procedures like removal of stop words, multiple spaces, etc. To do this, we use the Neattext library, which provides all the preprocessing steps as convenient functions. The various steps that we perform in preprocessing are the removal of:

- Hashtags (#politics, #India etc.).
- User handles (@JohnDoe, etc.) Multiple Spaces.

cyberbullying_type	tweet_text			freq
	count	unique	top	
age	7992	7992	Here at home. Neighbors pick on my family and ...	1
ethnicity	7961	7959	Racism won't stop as long as u stil select ur ...	2
gender	7973	7948	We proved that we're not just pretty faces. Oh...	2
not_cyberbullying	7945	7937	Strategicscoring should be classed as cheating...	2
other_cyberbullying	7823	7823	@ikralla fyi, it looks like I was caught by it...	1
religion	7998	7997	A Pakistani court has sentenced 86 members of ...	2

Fig. 1 Data description

	tweet_text	cyberbullying_type
37246	cis ppl spread hate & misinfo trans ppl though...	0
12581	like making gay jokes involving rape...not fun...	2
29310	So, save time, BSD family. negative response G...	4
44701	Fuck fucking hate dumb Hawaiians nigger lips!	1
30210	RT evil capitalist sellout cunt	4

Fig. 2 Final data post-preprocessing and encoding

- Stop words (a, at, the, etc.).
- URLs (ww.twitter.com, etc.).

After this, we check to ensure that both our test and train splits have received a balanced set of categories. We then use SciKit-Learn to encode the target variable. We use Label Encoding, the default encoder, and we do this for both train and the test sets (Fig. 2). We do this separately to avoid any potential data leakage that may occur across the two different sets. Since the label encoder also encodes the data by alphabetical order, we get the cyberbullying labels as:

0. Age.
1. Ethnicity.
2. Gender.
3. Not_cyberbullying.
4. Other_cyberbullying.
5. Religion.

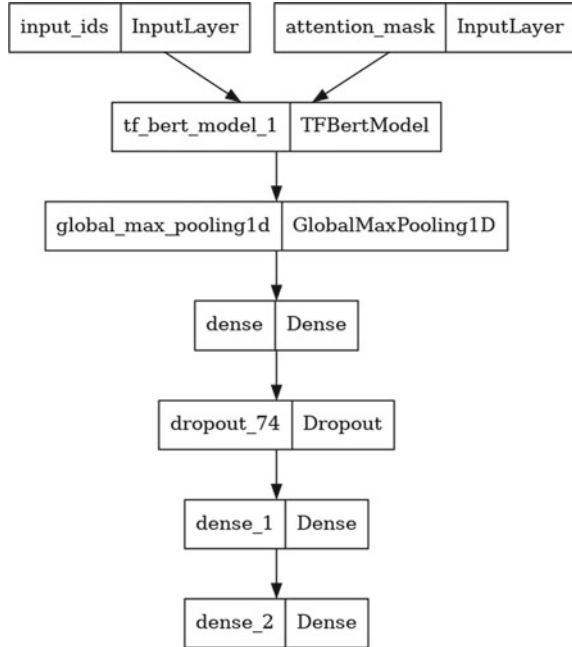
3.4 Tokenization

After performing preprocessing and encoding, we must perform tokenization on the data. Tokenization helps break down the inputs into smaller units based on the needs of the transformer. The tokenizers used are specific to each model. In this case, the various BERT tokenizers use the concept of word piece tokenizer to break words into sub-words, which helps simplify the meaning of the words for the model. It also helps generate input ids and attention masks, which the BERT model requires as inputs.

3.5 Model

In the model architecture used for this, we input the data into the respective transformer after the tokenization process. The base model remains the same, while we

Fig. 3 Model layout



change the transformer as per the BERT variation being used. We use a Global-MaxPooling layer and an alternating dense and dropout layer. Our output layer is a sigmoid activation function (Fig. 3).

4 Results

We use the typical metrics used to evaluate classification issues in general. Furthermore, since our task in this research required a multi-class classification rather than a straightforward binary classification, we used both macro and weighted averaged precision and recall values and micro and macro averaged F1-scores.

- Accuracy: This term is used to describe the ratio of correct predictions to the total number of predictions made by the model. Precision is measured as the ratio of accurate predictions to all accurate predictions, including both accurate and inaccurate predictions.
- Recall: This is the ratio of correctly predicted positive class labels to all positive class labels in the test set (including both correctly predicted positive and falsely predicted negative class labels).
- F1: The F1-score ranges from 0 to 1 and is ideally closer to 1, which is the harmonic mean of the precision and recall values. The macro-averaged F1-score is the arithmetic mean of the individual F1-scores for each class. The F1-score is

Table 1 Classification report of BERT-cased

	Precision	Recall	F1	Support
0	0.98	0.98	0.98	2398
1	0.95	0.97	0.96	2388
2	0.84	0.88	0.86	2384
3	0.68	0.46	0.55	2381
4	0.62	0.71	0.66	2347
5	0.87	0.99	0.93	2399
Accuracy			0.83	14,297
Macro avg.	0.83	0.83	0.82	14,297
Weighted avg.	0.83	0.83	0.82	14,297

```
array([0.9998299, 0.03181883, 0.2119491, 0.5962293, 0.60366315,
       0.11886037], dtype=float32)
```

Fig. 4 Predictions from BERT-cased

averaged using this method with equal consideration given to each label class. For all classes, the micro-averaged F1-score is calculated globally. The true positives, false negatives, and false positives from all the different classes are added together to achieve this.

4.1 BERT-Cased

On assessing the scores, the model has attained, we can see that it performs fairly well on the three specified categories of cyberbullying, i.e., age, ethnicity, and religion (Table 1). The model performs well in the category of other_cyberbullying but poorly in not_cyberbullying. Overall, the macro averages of precision, recall, and F1-scores are good (Fig. 4).

BERT returns the predictions in the form of probabilities, so we can see that for this specific instance, the highest probability falls into class 1, indicating that the tweet is cyberbullying based on the user's age.

4.2 BERT-Uncased

BERT-uncased has performed the second-best overall, giving the highest scores for the three named categories (Table 2). For the other two categories, it has performed more or less similar to BERT-cased. BERT-uncased generally works better than BERT-cased for most text-based information tasks, due to the removal of case

Table 2 Classification report of BERT-uncased

	Precision	Recall	F1	Support
0	0.97	0.98	0.97	2398
1	0.97	0.98	0.98	2388
2	0.86	0.88	0.87	2384
3	0.64	0.54	0.59	2381
4	0.64	0.67	0.65	2347
5	0.93	0.98	0.96	2399
Accuracy			0.84	14,297
Macro avg.	0.83	0.84	0.84	14,297
Weighted avg.	0.84	0.84	0.84	14,297

sensitivity as well as removal of accent markers which makes it simpler for the model.

4.3 *Albert*

ALBERT had performed the lowest out of all the other variants. The model performs well in the named categories of cyberbullying but poorly on other_cyberbullying and not_cyberbullying. This mainly comes down to the fact that ALBERT is a poor model for detecting ambiguous language predominantly used in social media (Table 3). It becomes difficult for a model to decipher that the text written in context may not represent the meaning of that text. Overall, ALBERT exhibited the poorest accuracy scores as well. This proves that it is not a good model for classification, especially from a resource like the Internet, where the language used is inherently ambiguous.

Table 3 Classification report of ALBERT

	Precision	Recall	F1	Support
0	0.97	0.98	0.98	2398
1	0.91	0.98	0.95	2388
2	0.91	0.79	0.84	2384
3	0.60	0.49	0.54	2381
4	0.58	0.65	0.61	2347
5	0.88	0.98	0.93	2399
Accuracy			0.81	14,297
Macro avg.	0.81	0.81	0.81	14,297
Weighted avg.	0.81	0.81	0.81	14,297

Table 4 Classification report of RoBERTa

	Precision	Recall	F1	Support
0	0.98	0.98	0.98	2398
1	0.95	0.98	0.97	2388
2	0.85	0.88	0.87	2384
3	0.71	0.51	0.60	2381
4	0.63	0.76	0.69	2347
5	0.95	0.97	0.96	2399
Accuracy			0.85	14,297
Macro avg.	0.85	0.85	0.84	14,297
Weighted avg.	0.85	0.85	0.84	14,297

4.4 RoBERTa

According to Table 4, RoBERTa performs significantly similarly to BERT-uncased. However, the model is more precise and accurate, gaining more precision and accuracy scores in the three categories of cyberbullying—age, ethnicity, and religion. However, the model is slightly dissimilar to BERT-uncased as it performs well in the category of not_cyberbullying but poorly in other_cyberbullying as opposed to BERT-uncased, which performed better in other_cyberbullying and poor in not_cyberbullying. This shows that RoBERTa is a more robust model for detecting ambiguous text like irony, sarcasm, etc. Overall, the macro averages of precision, recall, and F1-scores are good, which indicates that our model has performed satisfactorily in this task.

4.5 DistilBERT

DistilBERT performs almost similarly to the other models as well (Table 5). Though it has fewer parameters than all other models, it still manages to give an equally good performance in less time.

On assessing the scores, the models have attained, and we can see that RoBERTa has achieved the highest validation-balanced accuracy. DistilBERT has also performed with high accuracy and F1-score. Between the cased and uncased models, we can see that the uncased model has outperformed the cased model by a small margin, though nothing too significant. This difference would be more pronounced in larger bodies of texts. ALBERT had performed the lowest out of all the other variants. The model performs well in the named categories of cyberbullying, but poorly on other_cyberbullying and not_cyberbullying. This is because people may mean something else when they are typing out the tweet, and the model may be unable to pick this up from the overall context of the tweet. This is due to

Table 5 Classification report of DistilBERT

	Precision	Recall	F1	Support
0	0.95	0.98	0.97	2398
1	0.94	0.98	0.96	2388
2	0.90	0.84	0.87	2384
3	0.66	0.52	0.58	2381
4	0.62	0.71	0.66	2347
5	0.94	0.97	0.96	2399
Accuracy			0.84	14,297
Macro avg.	0.83	0.84	0.83	14,297
Weighted avg.	0.84	0.84	0.83	14,297

Table 6 Classification report of the model

	Accuracy	Precision	Recall	F1
BERT-cased	0.83	0.83	0.83	0.82
Bert-uncased	0.83	0.84	0.84	0.84
ALBERT	0.81	0.81	0.81	0.81
RoBERTa	0.85	0.85	0.85	0.84
DistilBERT	0.84	0.83	0.84	0.83

the kind of language used nowadays in social media and the frequent use of irony, sarcasm, etc. In Table 6, we have a comparison of how the four models perform in the four performance metrics.

5 Conclusion

The classification task seems to work fairly well for shorter bodies of text. Good F1-scores and accuracy measures were achieved across the board, even without much fine-tuning. To extend this work, we could consider fine-tuning each of the models, and this would lead to a more significant difference in their results. Future applications of transformer-based pre-trained BERT variants for the categorization of various forms of cyberbullying could involve the optimization of current models, the incorporation of domain-specific data, the use of a multimodal approach, the handling of imbalanced datasets, and a focus on interpretability. Another task would be to consider a dataset consisting of larger bodies of text and try to classify them. This would highlight the working of the various BERT transformers to a higher degree, giving a larger variation in the results and scores.

References

1. Dixon S (2023) Number of Social Media Users Worldwide 2010–2021. Statista. www.statista.com/statistics/278414/number-of-worldwide-social-network-users/. Accessed 26 Feb 2023
2. Auxier B, Anderson M (2021) Social media use in 2021. Pew Research Center, Washington, DC
3. Auxier B, Anderson M (2018) Teens, social media and technology 2018. Pew Research Center: Internet, Science and Technology. www.pewresearch.org/internet/2018/05/31/teens-social-media-technology-2018/. Accessed 27 Feb 2023
4. Duggan M et al (2017) Online harassment 2017. Pew Research Center: Internet, Science and Technology. www.pewresearch.org/internet/2017/07/11/online-harassment-2017/. Accessed 1 Mar 2023
5. National Cybersecurity Alliance (2016) National cyber security alliance survey reveals the complex digital lives of American teens and parents. National Cybersecurity Alliance. staysafeonline.org/news-press/press-release/survey-reveals-complex-digital-lives/. Accessed 4 Mar 2023
6. Centers for Disease Control and Prevention (2014) The relationship between bullying and suicide: what we know and what it means for schools
7. Slonje R, Smith PK (2008) Cyberbullying: another main type of bullying? *Scand J Psychol* 49(2):147–154. <https://doi.org/10.1111/j.1467-9450.2007.00611.x>
8. Dadvar M, Eckert K (2018) Cyberbullying detection in social networks using deep learning based models; a reproducibility study. *arXiv.org*. <https://doi.org/10.48550/arXiv.1812.08046>
9. Islam MM et al (2020) Cyberbullying detection on social networks using machine learning approaches. In: 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). <https://doi.org/10.1109/csde50874.2020.9411601>
10. Ali A, Syed A (2022) Cyberbullying detection using machine learning. *Pak J Eng Technol* 3(2):45–50. <https://doi.org/10.51846/vol3iss2pp45-50>
11. Giumetti GW, Kowalski RM (2022) Cyberbullying via social media and well-being. *Curr Opin Psychol* 45:101314. <https://doi.org/10.1016/j.copsyc.2022.101314>
12. Neelakandan S et al (2022) Deep learning approaches for cyberbullying detection and classification on social media. *Comput Intell Neurosci* 2022:1–13. <https://doi.org/10.1155/2022/2163458>.
13. Agrawal S, Awekar A (2018) Deep learning for detecting cyberbullying across multiple social media platforms. In: *Lecture Notes in Computer Science*, pp 141–153. https://doi.org/10.1007/978-3-319-76941-7_11
14. Al-Garadi MA et al (2019) Predicting cyberbullying on social media in the big data era using machine learning algorithms: review of literature and open challenges. *IEEE Access* 7:70701–70718. <https://doi.org/10.1109/access.2019.2918354>
15. Hani J et al (2019) Social media cyberbullying detection using machine learning. *Int J Adv Comput Sci Appl* 10(5):100587. <https://doi.org/10.14569/ijacsa.2019.0100587>
16. Wang J et al (2020) SOSNet: a graph convolutional network approach to fine-grained cyberbullying detection. In: 2020 IEEE International Conference on Big Data (Big Data). <https://doi.org/10.1109/bigdata50022.2020.9378065>.
17. Reynolds K et al (2011) Using machine learning to detect cyberbullying. In: 2011 10th international conference on machine learning and applications and workshops. <https://doi.org/10.1109/icmla.2011.152>
18. Devlin J et al (2019) BERT: pre-training of deep bidirectional transformers for language understanding
19. Gupta M (2022) Understanding BERT variants: Part 1. *Data Science in Your Pocket*. <https://medium.com/data-science-in-your-pocket/understanding-bert-variants-part-1-740fee88616>. Accessed 15 Mar 2023
20. Kotamraju S (2022) Everything you need to know about ALBERT, RoBERTa, and DistilBERT. *Medium*. <https://towardsdatascience.com/everything-you-need-to-know-about-albert-roberta-and-distilbert-11a74334b2da>. Accessed 18 Mar 2023

Design and Development of Evolutionary Algorithms for MPPT in a Solar PV System



Anurag Singh, Anirudh Saxena, and Anshika

Abstract This research study describes the need of harnessing solar energy and reaching the Maximum Power Point (MPP) to optimize the performance of PV Systems. Maximum Power Point is a unique point of operation, defined by a specified voltage and current values, at which maximum power is supplied by the system. This work analyses the efficiency of particle swarm optimization algorithm to reach MPP. The power extracted from the system has been recorded at variable irradiances to study the impact of changing weather conditions. To validate the performance of PSO, the results of this technique are compared with the results of conventional techniques of P&O and INC. The observations affirms that PSO tracks maximum power among all the discussed algorithms.

Keywords Incremental conductance algorithm · Maximum power point tracking · Particle swarm optimization · PV system

1 Introduction

With the advent of industrialization and urbanization, the demand for energy to power new technologies and machines increased significantly. Earlier, a major portion of energy requirements was met by the combustion of fossil fuels like coal, gas, and oil. These sources of energy are not only depleting rapidly but also causing great harm to the ecology of this planet by emitting hazardous greenhouse gases on their utilization. This urged the civilization to move towards renewable sources of energy, which are sustainable, environmental-friendly, and a better alternative to conventional

A. Singh (✉) · A. Saxena · Anshika
Department of Electrical Engineering, Delhi Technological University, New Delhi, India
e-mail: anuragsingh_2k19ee051@dtu.ac.in

A. Saxena
e-mail: anirudhsaxena_2k19ee041@dtu.ac.in

Anshika
e-mail: anshika_2k19ee047@dtu.ac.in

fossil fuels. Solar energy is a sustainable and eco-friendly energy source that is gaining popularity as the globe attempts to lessen its dependency on fossil fuels and tackle climate change. Photovoltaic (PV) cells, which convert sunlight into electricity, are used to harvest solar energy. Solar energy may greatly reduce greenhouse gas emissions and air pollution since it emits no hazardous by-products or pollutants. The detection of point of maximum power. Power will have a crucial and irreplaceable role in proliferating a global shift towards renewable sources energy. It increases the efficiency of SPV system and reduces the cost of electricity generation, enabling more and more nations to invest in this technology. Moreover, its integration with energy systems will help in storage of excess solar energy in the batteries during the day and its further utilization during low production times.

Numerous investigations, explorations, and scientific investments have been conducted for enhancing the energy output of a PV system. For efficient power generation, many MPPT algorithms have been devised and compared. According to research conducted by Larasati [1], a comparison was made between PSO and P&O algorithms using authentic weather data obtained from Taiwan's meteorological bureau. The study concluded that PSO algorithm outperformed P&O algorithm in producing the maximum power output in varying conditions. Bouderrès [2] performed a study using MATLAB/Simulink to evaluate the power generated by PSO and INC algorithms when the partial shading conditions are applied to the system. Mohamed [3] conducted a study in which a performance comparison of standard approaches P&O and INC for PV systems was carried out. Some researchers, including Merchaou et al. [4], proposed modifying the existing PSO technique by decreasing the inertia weight non-linearly, which was also evaluated under partial shading situations. In order to improve the system tracking time, Mirhassani [5], suggested a modification to the traditional PSO approach by incorporating a variable sampling time strategy. They tested the performance of this technique using MATLAB simulations and also implemented it on a real boost converter by using a digital signal controller. Chen [6] developed a model for a three-phase photovoltaic grid-connected system and used an enhanced PSO algorithm with improved parameter coordination to address the algorithm's limitation of falling in local maxima. Figueiredo [7], modelled a hybrid of PSO and P&O techniques employed in PV system that undergoes partial shading conditions as well. Amri [8] developed a hybrid of INC and PSO techniques, the algorithm implemented the INC technique during the initial tracking phase and then switched to the PSO technique for the final phase. The algorithm's performance was validated by simulating it in Xilinx software.

In this research paper, we discussed the PSO MPPT algorithms for stand-alone PVS with changing solar irradiation and simulated them using MATLAB/Simulink. Under the same conditions, our approach was compared against conventional algorithms such as P&O and INC for validation.

2 Photovoltaic System

The PV array in a photovoltaic (PV) system is normally made up of many PV modules that are linked in series and or parallel to obtain the optimal voltage and current levels [9, 10]. To implement MPPT in a PV system, a basic conceptual diagram is shown (in Fig. 1) to illustrate the process. The diagram commonly illustrates a DC–DC converter, a PV module, and an MPPT controller. The PV module produces voltage and current, which are sent into the DC–DC converter.

2.1 Photovoltaic Cell

A PV cell refers to a semiconductor device that transforms sunlight into electrical energy. It can be modelled as an ideal current source connected parallel to a diode [9]. The ideal source of current represents the current created by the PV cell in response to incoming light (in Fig. 2).

Current I is proportional to the incident power and can be mathematically presented by the following expression:

$$I = I_l - I_D - I_{sh} \tag{1}$$

Fig. 1 Block diagram for MPPT implementation in SPV system [9]

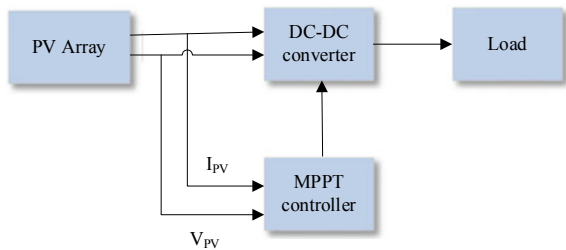
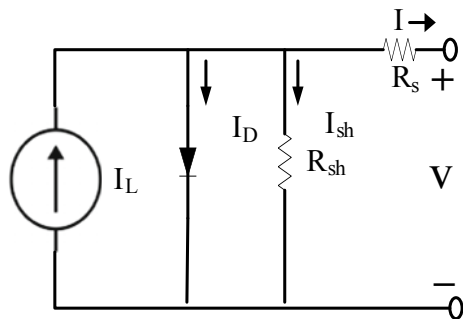


Fig. 2 Electrical circuit diagram of PV cell [9]



$$I_d = I_o \cdot e^{\left(V + \frac{IR_s}{nV_T}\right)} - 1 \tag{2}$$

where V depicts resultant voltage, the current at output end is depicted as I , I_o depicts the saturation current, T depicts the cell temperature.

2.2 Photovoltaic Panel

Several photovoltaic cells are linked to form modules, which are then linked together to make a PV panel which is also known as a solar panel as shown (in Fig. 3) [9].

Simulink model incorporates the utilization of SunPower SPR-305 WHT-U PV module. The array is designed with a parallel string of 6 modules that are connected in series to attain the desired terminal voltage. A 244.62 W array was employed to build the PV system. Table 1 presents the PV array’s electrical parameters under STC conditions.

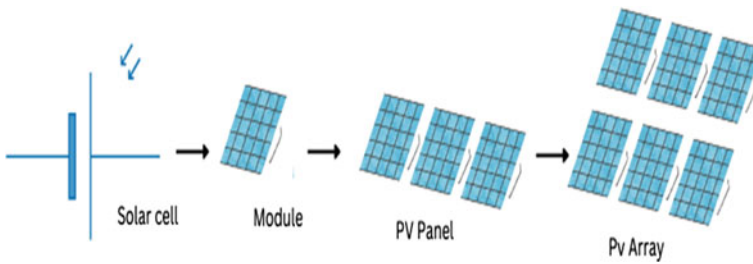


Fig. 3 Photovoltaic panel [9]

Table 1 The electrical specifications related to photovoltaic modules (SunPower SPR-305 WHT-U)

Parameter	Value
Max. power (P_{max})	245 W
Short circuit current (I_{sc})	8.6 A
Open circuit voltage (V_{oc})	37.29 V
Max. current at (P_{max})	8.1 A
Voltage at (P_{max})	30.2 V
The number of modules Linked in a series for each string	6
Overall parallel string count	1
STC condition	1000 W/m ² irradiance at 25 °C

2.3 Boost Converter

The boost converter is a DC–DC converter used to boost the voltage at the output. It stores energy in an inductor and then releases it at a greater voltage to the output. Overall, it is a reliable, flexible, and efficient choice for increasing the voltage of a DC power source.

3 Maximum Power Point Tracking

The MPPT algorithm regulates the voltage and current at the output end to guarantee that it operates at its MPP under various atmospheric circumstances such as variations in solar irradiation and temperature. The MPPT allows the PV system to function at peak efficiency by tracking the MPP to generate the maximum power under a given set of environmental circumstances by controlling the duty cycle (α) automatically to the required value to maximize the output power [9].

4 Perturb and Observe Algorithm

The Perturb and Observe method is utilized because its straightforward implementation. It relies on the correlation between the $P_{O/P}$ and $V_{O/P}$ of the PV module. The MPP can be attained by automatically controlling the duty cycle.

The operational point lies at left hand side of the MPP when a +ve voltage rise (V_{pv}) results in higher power (P_{pv}). If power drops, the operational point lies at right hand side of the MPP [9]. By analysing the effect of altering the voltage, we can determine the functioning operation with respect to the MPP and achieve P_{max} by adjusting the $I_{O/P}$ of the PV module to an appropriate value. Flowchart of this algorithm is illustrated (in Fig. 4).

5 Incremental Conductance Algorithm

The incremental conductance technique is orchestrated on the rate at which the conductance changes of PV array w.r.t PV array voltage. It uses the instantaneous value of voltage and current of the PV array to determine the conductance and then compares it with the previous conductance value to determine the direction of the voltage change required to track the PMP [9]. The flowchart depicting this technique is illustrated (in Fig. 5).

$$\frac{dP}{dV} = \frac{d(V \cdot I)}{dV} = I + \frac{dI}{dV} \quad (3)$$

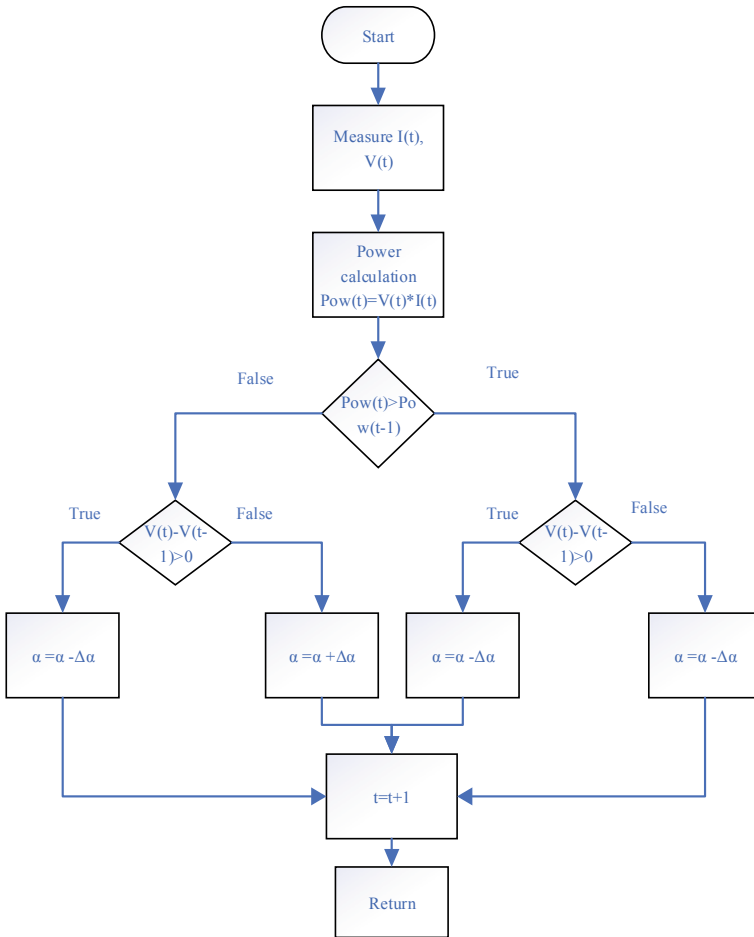


Fig. 4 Flowchart of P&O [9]

For MPP to reach, $dP/dV = 0$

$$\frac{dI}{dV} = \frac{-I}{V} \tag{4}$$

$$\frac{dP}{dV} > 0 \text{ then } V_p < V_{mpp}$$

$$\frac{dP}{dV} = 0 \text{ then } V_p = V_{mpp}$$

$$\frac{dP}{dV} < 0 \text{ then } V_p > V_{mpp}$$

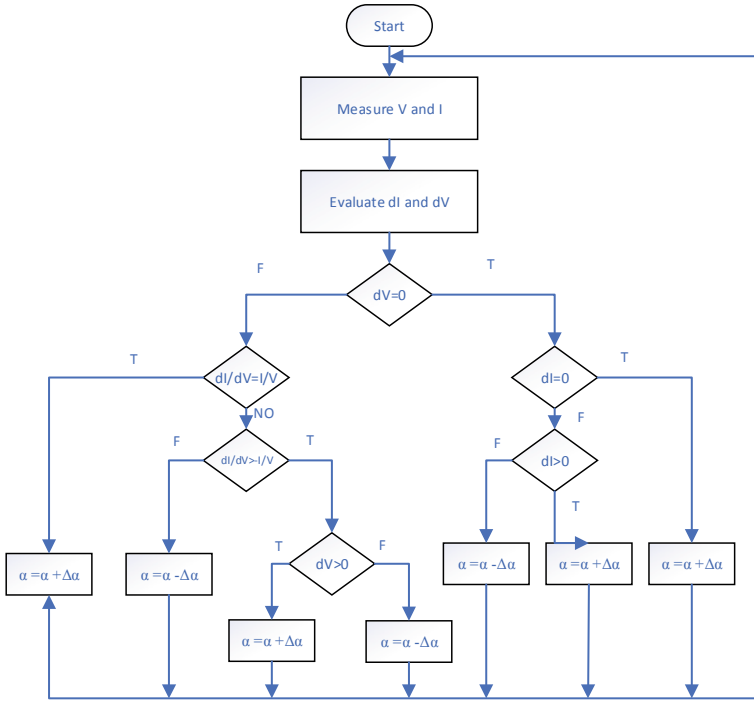


Fig. 5 Flowchart of incremental conductance method [9]

6 Particle Swarm Optimization

In order to solve computationally challenging optimization problems, the particle swarm optimization (PSO) method uses evolutionary and resilient stochastic principles inspired by nature. In 1995 James Kennedy and Russ Eberhart developed this technique [11].

PSO technique is based on the movement and reasoning of swarms. Each particle adjusts its position during a PSO iteration based on its prior experiences as well as the experiences of its surrounding particles. Following that, the particle moves at a speed that is determined by both its best position P_b and the best position of its group G_b [11, 12].

v_n^k signifies the velocity of the n th particles at the k th iteration, then the Eqs. (5 and 6) below will, respectively, give the new velocity and position at the $(k + 1)$ th iteration.

$$v_n^{k+1} = w \cdot v_n^k + c_1 \cdot \text{rand}_1(P_b - x_n^k) + c_2 \cdot \text{rand}_2(G_b - x_n^k) \tag{5}$$

$$x_n^{k+1} = x_n^k + v_n^{k+1} \tag{6}$$

where w is inertia weight, P_b is self best position, G_b is Global best position, k is iteration count, x_n is current location of particle n , $rand_1$, $rand_2$ are random numbers, v_n is current velocity of particle, c_1 is cognitive coefficient, c_2 is social coefficient.

The equation can be expressed as follows if the position of the particle is viewed of as the duty cycle itself and the velocity as any disturbance or change within that cycle. [11]

$$D_n^{k+1} = D_n^k + v_n^{k+1} \tag{7}$$

The process of utilizing particle swarm optimization to achieve maximum power is illustrated in the flow chart depicted (in Fig. 6).

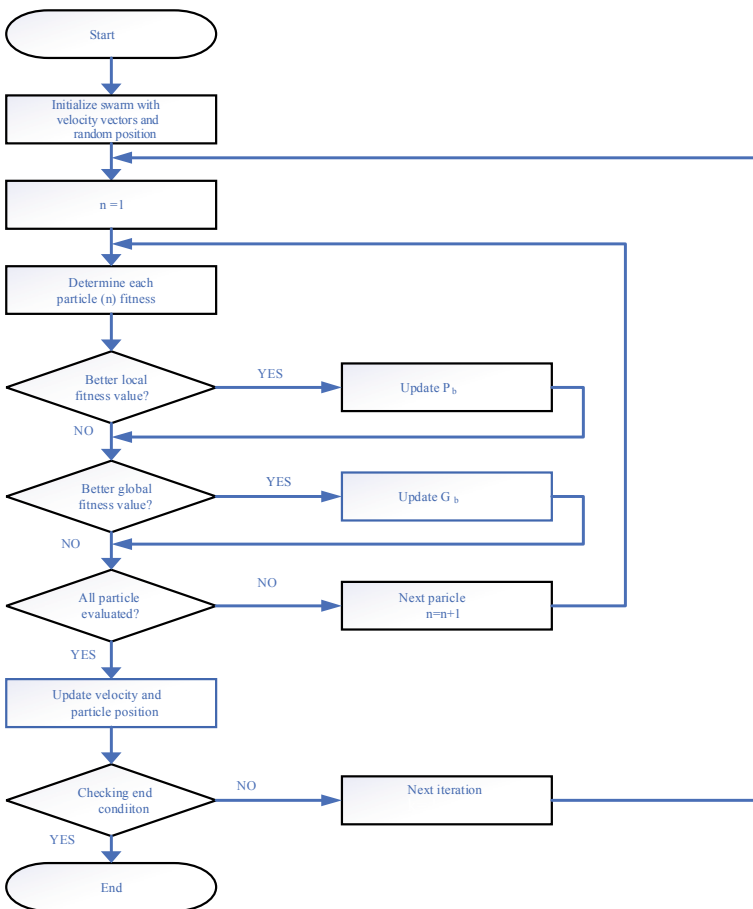


Fig. 6 Flowchart of PSO algorithm [11]

7 Simulation and Results

Within this header, the focus is on presenting simulation results of PSO to obtain maximum power for photovoltaic (PV) systems that are subject to varying irradiation levels at a fixed temperature of 25 °C. The system’s simulation has been carried out considering variable solar radiation levels, ranging from 600 to 1000 W/m² with increments of 200 W/m². The results are verified by comparing them under similar conditions with both the INC and the P&O.

7.1 A Simulation Modelling and Result Analysis of PSO, INC, and P&O MPPT Algorithm with Varying Solar Irradiation

MATLAB Simulink model of all three discussed algorithm PSO, INC, and P&O are shown (in Figs. 7, 8, and 9), respectively.

The performance of PV systems using three different MPPT techniques (PSO, INC, and P&O) was evaluated under varying irradiation levels (1000, 800, and 600) W/m² at 25 °C temperature. Figure 10 demonstrate variation in output power when the level of irradiation is changed at various levels.

Table 2 presents an analysis of the outcomes obtained by tracking power for all three techniques, as shown in the Fig. 10.

According to the data in Table 2 and power curves for various techniques, it’s clear that PSO produces the most significant output power, with an average of approximately 1.42 kW in STC conditions, and consistently achieves the maximum power output in other conditions. On the other hand, INC produces an output of 1.40 kW, and P&O only 1.36 kW, comparatively lower than PSO.

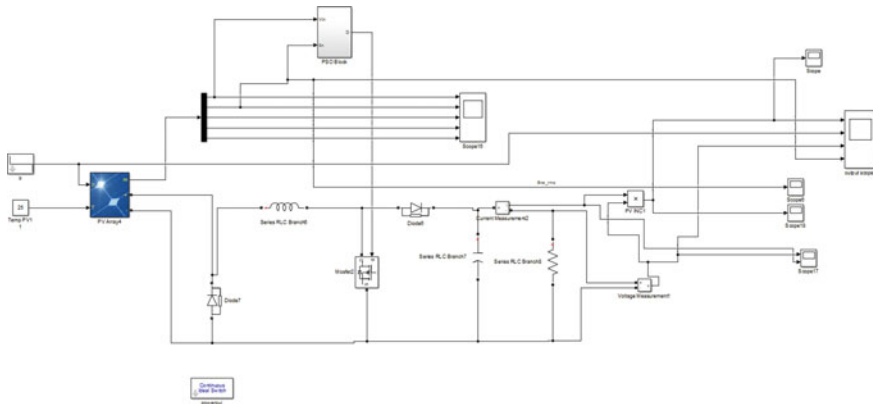


Fig. 7 System modelling with PSO method in Simulink

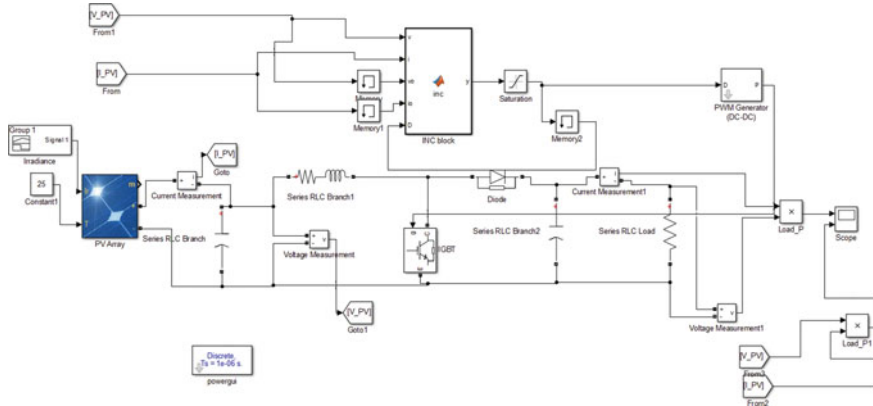


Fig. 8 System modelling with INC method in Simulink

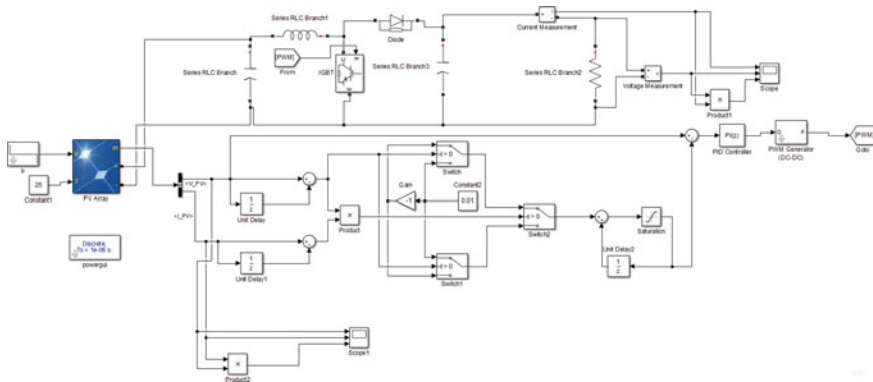
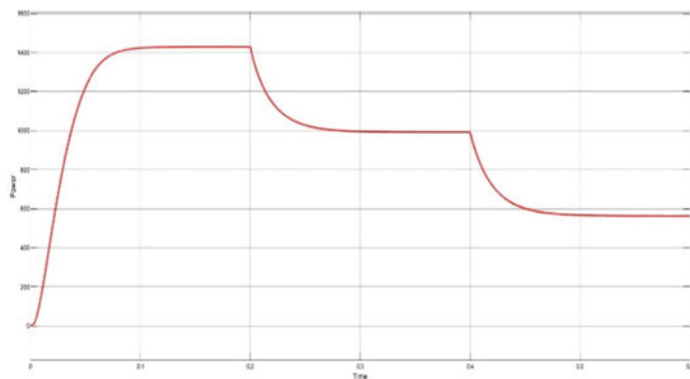


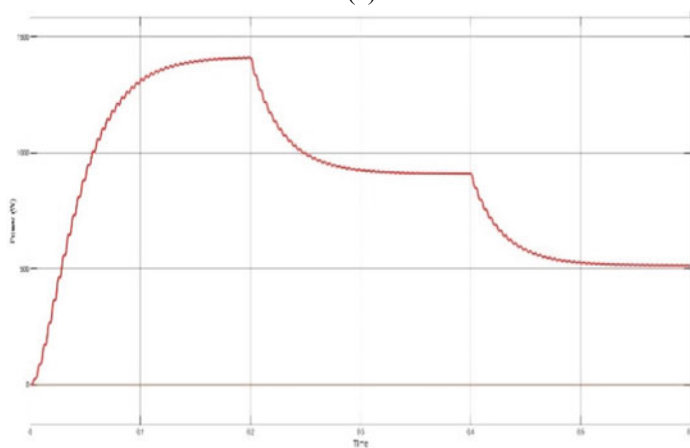
Fig. 9 System modelling with P&O method in Simulink

Furthermore, the study revealed that both INC and PSO performed well under varying irradiation conditions, but PSO exhibited more effective performance parameters than INC. PSO settled in less time and had a higher tracking speed, which made it more efficient in varying atmospheric conditions. On the other hand, P&O's performance was suboptimal due to its continuous duty cycle adjustment, leading to oscillations around MPP.

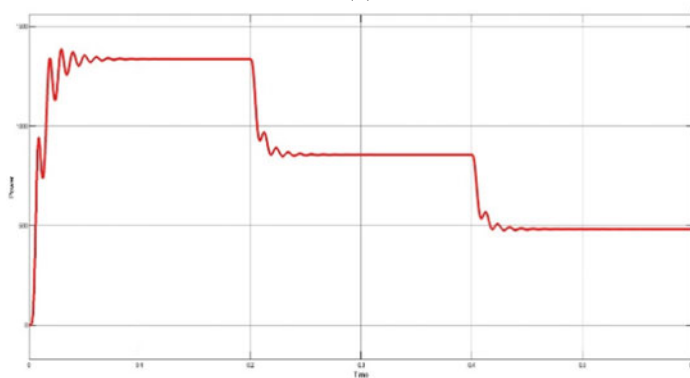
Although PSO techniques show promise in achieving MPPT in PV systems, it is important to consider their limitations as well. It converges to local maxima under partial shading conditions, which leads to low system efficiency. The computational resources required for PSO can be substantial, making it difficult to implement in real-time systems. The algorithm's performance is highly dependent on the selection of its parameters, which can be challenging to optimize. PSO may take time to adjust, it may not be the most suitable option for rapidly changing systems. Therefore, it is



(a)



(b)



(c)

Fig. 10 PV array maximum output power for variable irradiation with (i) PSO (ii) INC (iii) P&O algorithm

Table 2 Comparison table of obtained power

Techniques	Particle swarm optimization (kW)	Incremental conductance (kW)	Perturb and observe (kW)
<i>Irradiance (W/m^2)</i>			
1000	1.428	1.406	1.337
800	0.991	0.909	0.856
600	0.563	0.513	0.481

important to take these limitations into account and explore further modifications to enhance the system's performance.

8 Conclusion

This research paper presents a comparative analysis of three distinct MPPT techniques—INC, P&O, and PSO algorithms—applied to photovoltaic systems through experimental investigation under varying irradiance. The P&O method is highly sensitive to initial conditions and tends to oscillate around the optimal value. The algorithm incremental conductance reacts faster than P&O, seems to be an improvement over the algorithm P&O, despite the complexity of the algorithm, it is able to handle sudden weather changes more effectively. Out of all these algorithms, incremental conductance and P&O are commonly employed, but particle swarm optimization method has outperformed other methods in terms of performance, accuracy, and tracking speed with inexpensive microcontroller.

References

1. Larasati DA, Teng J-H, Chen C-R (2020) Comparative analysis of maximum power point tracking algorithm for photovoltaic systems. In: 2020 IEEE international conference on consumer electronics—Taiwan (ICCE-Taiwan). <https://doi.org/10.1109/icce-taiwan49838.2020.9258089>
2. Bouderrès N, Kerdoun D, Chiheb S et al (2022) PV array power optimization under shading condition using PSO and IncCond MPPT control. In: 2022 19th international multi-conference on systems, signals & devices (SSD). <https://doi.org/10.1109/ssd54932.2022.9955935>
3. Mohamed SA, Abd El Sattar M (2019) A comparative study of P&O and INC maximum power point tracking techniques for grid-connected PV systems. SN Appl Sci 1. <https://doi.org/10.1007/s42452-018-0134-4>
4. Merchaoui M, Sakly A, Mimouni MF (2018) Improved fast particle swarm optimization based PV MPPT. In: 2018 9th international renewable energy congress (IREC). <https://doi.org/10.1109/irec.2018.8362525>
5. Mirhassani SM, Golroodbari SZM, Golroodbari SMM, Mekhilef S (2015) An improved particle swarm optimization based maximum power point tracking strategy with variable sampling time. Int J Electr Power Energy Syst 64:761–770. <https://doi.org/10.1016/j.ijepes.2014.07.074>

6. Chen X, Chen D, Li J, Lin Y (2022) MPPT control strategy of photovoltaic based on improved particle swarm optimization. In: 2022 international conference on manufacturing, industrial automation and electronics (ICMIAE). <https://doi.org/10.1109/icmiae57032.2022.00073>
7. Figueiredo S, Nayana Alencar Leao e Silva Aquino R (2021) Hybrid MPPT technique PSO-P&O applied to photovoltaic systems under uniform and partial shading conditions. *IEEE Latin Am Trans* 19:1610–1617. <https://doi.org/10.1109/tla.2021.9477222>
8. Amri A, Moussa I, Khedher A (2022) Design and simulation of a PV system controlled through a hybrid INC-PSO algorithm using XSG tool. In: 2022 IEEE 9th international conference on sciences of electronics, technologies of information and telecommunications (SETIT). <https://doi.org/10.1109/setit54465.2022.9875738>
9. Assiya L, Aziz D, Ahmed H (2020) Comparative study of P&O and INC MPPT algorithms for DC-DC converter based PV system on MATLAB/SIMULINK. In: 2020 IEEE 2nd international conference on electronics, control, optimization and computer science (ICECOCS). <https://doi.org/10.1109/icecocs50124.2020.9314498>
10. Jain K, Gupta M, Kumar Bohre A (2018) Implementation and comparative analysis of P&O and INC MPPT method for PV system. In: 2018 8th IEEE India international conference on power electronics (IICPE). <https://doi.org/10.1109/iicpe.2018.8709519>
11. Chaieb H, Sakly A (2015) Comparison between P&O and P.S.O methods based MPPT algorithm for photovoltaic systems. In: 2015 16th international conference on sciences and techniques of automatic control and computer engineering (STA). <https://doi.org/10.1109/sta.2015.7505205>
12. Liu Y, Xia D, He Z (2011) MPPT of a PV system based on the particle swarm optimization. In: 2011 4th international conference on electric utility deregulation and restructuring and power technologies (DRPT). <https://doi.org/10.1109/drpt.2011.5994058>
13. Hossam EL-Din A, Mekhamer SS, M.El-Helw H (2020) Comparison of MPPT algorithms for photovoltaic systems under uniform irradiance between PSO and P&O. *Int J Eng Technol Manage Res* 4:68–77. <https://doi.org/10.29121/ijetmr.v4.i10.2017.108>

A Blockchain-Based Custom Clearance Solution for International Trade Using IPFS and Non-fungible Tokens



Mansimran Rehal, Rohit Ahuja , Divya Gandhi, and Ayush Sharma

Abstract Imports and exports of a nation heavily impact its GDP, interest rates, exchange rate, and inflation rate. With the ongoing transition from traditional to digital methods, the procedure of importing-exporting goods also witnessed a trend in the paper-based traditional approach where the custom department's officers verify all of the paperwork in accordance with national customs law before approving imports and exports. However, this approach is extremely time-consuming and has a higher risk of manipulation/tampering with the documents to make illegal imports and exports happen. To digitalize the previous approach, the development of systems with centralized servers came into existence which could connect the exports and imports and thus, reduce the time factor in communication but increase the risk of data(documents) getting mutated or lost (in case of server crash). With the evolution of blockchain technology, an efficient alternative to providing solutions for the above-mentioned drawbacks in import-export exists. Security, encryption, and transactions amongst individuals (such as exporters, importers, banks, customs offices) that cannot be changed are all provided by blockchain technology. This paper presents a blockchain-based solution for automating Customs Clearance between trading countries by employing non-fungible tokens (NFTs) for Certificate of Origin (COO) and Letter of Credit (LCs) and communicating agreement files on the blockchain network using Inter Planetary File System (IPFS).

Keywords Blockchain · International trade · Digitalization · NFT · Certificate of Origin (COO) · Letter of Credit (LC)

M. Rehal (✉) · R. Ahuja · D. Gandhi · A. Sharma
Thapar Institute of Engineering and Technology Patiala, Patiala, India
e-mail: mrehal_be19@thapar.edu

R. Ahuja
e-mail: rohit.ahuja@thapar.edu

A. Sharma
e-mail: asharma16_be21@thapar.edu

1 Introduction

A number of interested parties, including banks, logistics firms, companies that support international trade, and government agencies, participate in sophisticated mechanisms used in export–import transactions, making transaction costs high and the delivery process slow from the manufacturer to the customer [1, 2]. In the process, there can be some human errors as documents are issued physically, where the original documents can be tampered, and also because of being unaware of permissions to use those certificates, some certificates can be used illegally [3]. Some digitization has been done to reduce manual work, but the system is centralized, which leads to an increase in workload on the system, crashing the system [4, 5]. Here the blockchain comes into play, which will reduce problems such as lowering transaction costs, and delivery time, and keeping track of orders without altering the data in the documents. The decentralized applications (DApps) can be operated on the blockchain network [6, 7]. Blockchain is the system where we can store all our documents regarding shipping events between parties in an electronically distributed and immutable ledger that can detect any discrepancy if changes are made to the documents [8–10]. In the customs clearance of goods, blockchain is used for message exchange. All agencies must participate in the blockchain network as nodes in order for the customs authority to use blockchain. To make any export, the exporter has to file for a Certificate of Origin (COO), for which a commercial invoice and bill of lading are to be submitted [11]. When any import is made into the country, the importer must submit an import declaration to the customs authority. Customs officers evaluate the risk associated with imported items and determine the corresponding customs taxes [12]. Documents like original invoices from the exporter or importer, import/export permission from other government agencies, lab reports, licenses from foreign trade departments, bonds, country of origin certificates, duty payment receipts, and so on must be physically verified as part of the goods clearance process [13]. Therefore, this article aims to design an Ethereum smart contract based on the blockchain among cooperating agencies to automate the cross-validation of documents. The COO is to be minted as a non-fungible token (NFT). Ethereum is a blockchain platform that is open-sourced for creating DApps, which are lines of code known as smart contracts that specify how value should be handled [12, 14]. In order to stop denial-of-service attacks and encourage the use of effective smart contracts, Ethereum charges a fee for each transaction. The Ethereum Virtual Machine (EVM), which is independent of hardware and environment, is used to run the contract code. Following validation, every operation is carried out in an EVM full node before being appended to a block in the chain.

2 Literature Review

In [11] A COO management framework and smart contract are proposed where an application will produce a certificate ID. Transactions are written in blocks. The application fetches data from the blockchain, digitally authenticates and signs it, and sends the data back to the blockchain as a transaction. The sender sends the data from the ledger and uses it for its purpose. IPFS hashes are valid for all countries which will gain access to the same agreement assigned to them. They decided that their solution would help importers and exporters send certificates safely and quickly, and would prevent the use of government funds by creating fake COO. In [12] they discussed the possibilities and thoughts on how blockchain could be used in many businesses, especially regarding customs, including the interests of traders, financial companies, and insurance companies. Discuss ongoing initiatives on cross-border trade and the use of blockchain. For example, a seller (exporter) and a buyer (importer) agree on international trade. It will be possible for both parties to achieve their goals thanks to self-service smart contracts, which should be considered in the works related to the use of blockchain. Using the information distributed, they will also be allowed to see the same information about the status of the shipment; so it will be easier for them to contact the union directly and quickly when a shipping problem occurs (even if there is no mutual trust). They conclude that the power of blockchain will increase the "traceability" and "connectivity" of connected devices, and customs can gain a broader and clearer view of the global economy using distributed information. In [2] they discussed various past studies (some notable researchers Mougayar, Nakamoto, Okazaki, Clark, and Burstall, etc.) and how they used technology to provide a supply chain and customs environment to achieve the goals and principles of SAFE standards. According to their content, thanks to the features of the blockchain, governments, customs, and enterprises can improve the accuracy and predictability of global supply, strengthen customs/trade cooperation, and overcome challenges and opportunities for products. They concluded that features will enhance cooperation between customs authorities and other government agencies related to international trade and security, for example, through a single window. Finally, it will be easier for all participants in international trade to contribute to the supply of nonconforming goods through a secure international trade chain. In [8] they discussed how we can use smart contracts to automate the import–export declaration by aggregating information from different places with much accuracy and efficiency, this will digitalize and automate transactions and reduce fraud. A study says the customs authority in the Netherlands processes over 160 million statements every year which are projected to increase to 500 million in upcoming years, manually cross-checking each statement is not possible. The smart contract reduces cross-validation that is done manually. In 2008 Nakamoto introduced blockchain technology that made both parties who do not completely trust one another exchange information without communicating details without keeping transaction records in central records. To examine the authenticity of the statement filed by the importer (Freight Forwarder that often acts as an importer) approach of cross-checking the import statement with other documents like commer-

cial invoices or a Bill of Lading is used. A customs authority recognize something suspicious in the filed declaration it can appeal other documents that are used for collecting other information like commercial invoice requested from the exporter. Both documents are differentiated to decide if the values are the same or not. If all data is not there can be possible fraud by undervaluation in the invoice. Data points to cargo via ID to authorize grouping of data to make import declaration of a shipment. The customs authority must sign the declaration to file an import declaration and every transaction also needs a signature from the customs authority to confirm the transaction. Ledger stored data is used in the smart contract. In [9], blockchain technology in international trade will reduce costs and provides more security and transparency in communication and trading platforms. Customs clearance will be easy due to blockchain technology. In export–import, transactions for foreign trade are high-valued and occur among manufacturing, trading, and service companies in the legal framework that involves packing goods, loading into the container, customs clearance, main transport, and unloading of goods. Any discrepancy cause additional charges and the image of the company. In the EU, the letter of credit is rarely utilized, because a huge amount of time is consumed to manage the payment, due to the verification by the bank. There is an important role of documents in export–import through which the shift of rights over goods is carried out, on the one hand, and the payment of their counter value, on the other hand.

3 Existing Scheme

Fig. 1 demonstrates the existing system followed for customs clearance.

Step 1: Filing Certificate of Origin (COO): The COO is issued by the Ministry of Goods, to obtain it, the exporter has to present the commercial invoice and the Bill of Lading.

Step 2: Goods are allowed to be exported from the source country: After all document verification, the goods are allowed for export by customs departments on the exporter side.

Step 3: Document verification, calculation of import duties at the destination country: The customs department on the importer side does all the document verification, and calculations of all import duties and taxes.

Step 4: Importer files for LC: The Importer's bank drafts the Letter of Credit (LC) when the importer submits import bills and proof of delivered goods.

Step 5: Declaration by the importer bank to the exporter bank for payment: The Letter of Credit is reviewed and approved by the exporter's bank before being forwarded to the exporter.

Step 6: Final payment to the exporter: The final payment is made to the exporter by the exporter's bank.

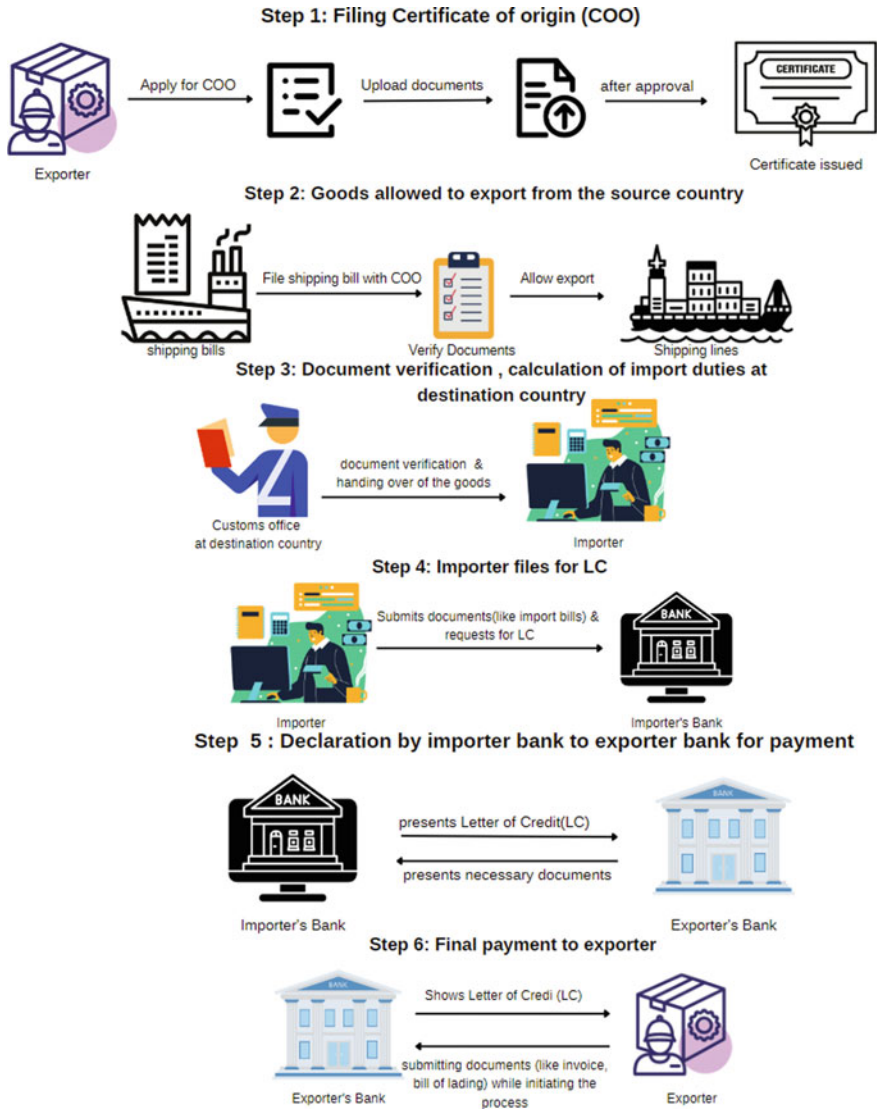


Fig. 1 Demonstration of the traditional customs clearance process

3.1 *Deficiencies of the Existing Scheme*

- Opaque process: The importer cannot track the order, and other entities too cannot track the process in real time.
- Manipulation of data: In the current scenario, the customs department is all paperwork, and the documents can be manipulated and misused.
- Prone to human errors: Due to paperwork there can be errors during entering some data, which can cause huge losses.

4 Proposed Framework

The Government of India created the Indian Customs Electronic Data Interchange System (ICES) which helps to file import and export bills, which is a central platform that interacts with multiple parties like banks, ports, and shipping lines to export items. As the blockchain system is used so the process becomes decentralized to prevent fraud and communication delays during the transmission and information can be viewed on all the nodes. All parties involved are used as nodes in the network and a smart contract is created among all nodes and Proof-of-Work (PoW) is needed to commit the block in the network. The smart contracts are written on the Ethereum platform. The transaction fee for computations is deducted called Gas. We are aware that the content of the document within the smart contract is stored using the peer-to-peer (P2P) file system known as Inter Planetary File System (IPFS). It stores signed agreements among different parties in the country.

1. The COO is minted as a non-fungible tokens (NFT) and is pushed to the ledger by the exporter.
2. The goods after all the calculations and verification are allowed to export.
3. The shipping lines are the next entity that comes into play, the goods are tracked and the location of goods at the checkpoints is updated as the COO also contains the route of the good along with the source and destination. This is updated on the blockchain and a new block is added each time location is updated or some new change is made the block is linked to the previous block in the blockchain as each block has a hash of its own hence it can't be tempered this is the advantage of the blockchain. It is automated, smart contracts are deployed on each node in the network and it limits who can perform the actions.
4. After the shipping lines, customs of the importer, where verification of all documents and goods is done again. Now that the goods have been unloaded and reached the importer, the importer fetches the COO from the blockchain, pushes the import bills on the blockchain, and gets the benefits of the COO. Here, the ownership of goods is changed by the exporter through the COO. Here the importer files the new COO as the ownership has been changed, which is now pushed to the ledger. The importer now approaches the bank to make the payment as the smart contract is deployed, so the payment has to be done automatically.

5. As the exporter has to file for COO, the importer has to file for a Letter of Credit (LC). To file LC documents such as proofs that goods have been shipped, commercial invoices are submitted. LC is a declaration that the payment will be made by the importer to the exporter. The bank fetches the COO, and the LC is minted and pushed on the blockchain ledger. LC is authorized, the exporter bank fetches the LC from the blockchain, and the payment is made to the exporter.

As the system is decentralized, each node has a copy of all the documents and transactions and is a write-once, read-only shared ledger. Transactions are made in real-time and are transparent to all nodes. The COO and LC are two certificates that are used for both importers and exporters and are important as they contain all the information about the trade. Before issuing them, almost all documents are used. The blockchain-based system has helped us reduce the time of verification of documents and goods, which leads to no delay in the export/import at the customs, and has eased the payment system too, as payment verification is also automated, which, on the other hand, has also increased the security from the tempering of the data.

4.1 Participating Nodes

Below is a description of the participating nodes present in Fig. 2.

1. Exporter/Seller: The exporter has to initiate the process by providing the commercial invoice and bill of lading to obtain a COO in the form of NFT. Then, apply to the customs office. The payment for the export is made upon receiving the LC from the importer.

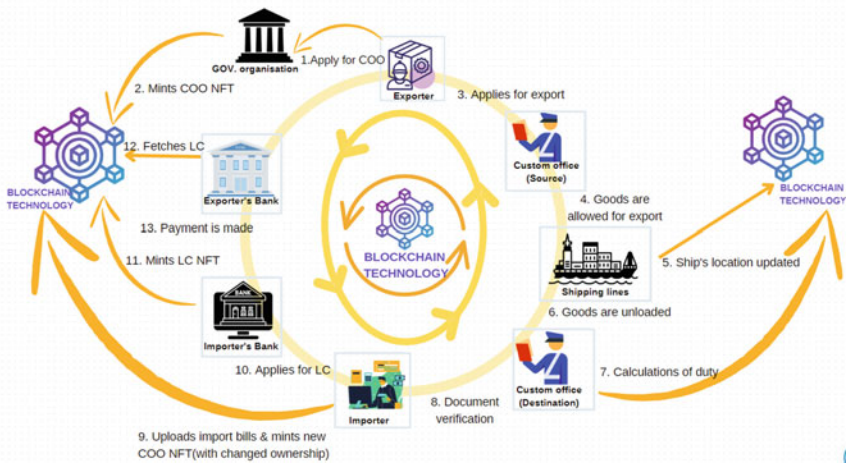


Fig. 2 Proposed system

2. Exporter's customs office: Upon verification of the COO, the customs office will calculate the applicable duty, and the same will be visible to the exporter.
3. Shipping lines: The ship operator will update the location of the goods from time to time.
4. Importer/Buyer: The importer has to push the import bills on the blockchain (for getting the ownership changed) upon receiving the goods and request for the LC to be pushed onto the blockchain in the account (upon agreeing to make the payment to the exporter).
5. Importer's customs office: Upon verification of the COO, the customs office would calculate the applicable tariffs and the same would be visible to the importer.
6. Importer's bank: An importer bank offers financial services to make it easier for commodities to be imported into a nation.
7. The exporter's bank: Provides financial services to exporters and supports international trade operations.

5 Implementations

The remix Ethereum IDE has been used to compile the solidity code, which is written in the high-level programming language solidity for the smart contract. The Metamask extension is added to the Chrome browser after the compilation of smart contracts. For testing blockchain implementation, a local secure account is created and topped up with 1 ether. This is because each activity has a transaction cost in terms of Ether, the cryptocurrency used by Ethereum. The Ethereum and conventional web user interfaces are separated via Metamask. Without deploying a full Ethereum node, Metamask makes it possible to execute decentralized applications. The blockchain-based solution for automating customs clearance smart contracts pseudocode is shown in the snapshot that follows. Six methods are suggested in this work to mint Certificates (LC and COO), calculate duties if UAE is an importer country, update the current location of the goods, calculation of duties if India is an importer country, update COO certificate, and extract LC certificate data from blockchain hyper ledger.

Architecture of Proposed System

As shown in Fig. 3, our system is based on blockchain, a decentralized network of peer-to-peer (P2P) systems. It consists of 5 entities: Importer, Exporter, Shipping lines, Customs department (source country), and Customs department (destination country). The customs department of both the source country and destination country acts as an intermediary between the importer and exporter and allows secure import–export activity to happen. Peer-to-peer (P2P) technology eliminates the need for a middleman, or central server by enabling network users to execute transactions freely. Similarly, no administrator is needed to keep track of user transactions on the network in blockchain. Each node maintains a complete copy of the ledger and

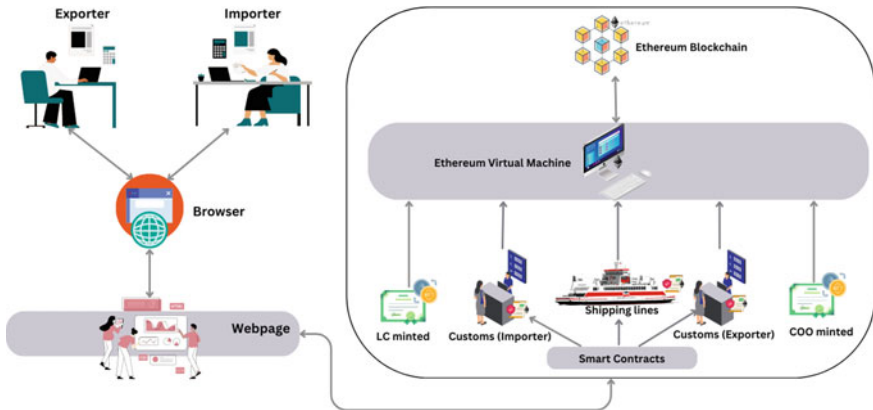


Fig. 3 Architecture proposed system

checks its validity with other nodes to ensure that data is accurate. It functions as a decentralized ledger for one or more digital assets. Furthermore, it will automatically reject any node (peer) that tries to interfere within the network.

6 Discussion and Results

In this section, we will be covering practical points like experiment results, performance evaluation, and security analysis are also discussed.

6.1 Feature Analysis

The features of the proposed framework are contrasted with the existing scheme [11, 15, 16]. Table 1 compares the proposed scheme with the existing work and [11, 15, 16]. Figure 4 demonstrates the feature analysis of the proposed scheme with the existing work and [11, 15, 16] in graphical form.

6.2 Experimental Results

The outcomes of the experiment have been discussed in this section. A smart contract that mints NFT, and update certificate details on the blockchain ledger. Use some test Ether in your Metamask account to deploy the contract. Therefore, 0.05136267 ETH is deployed for the Sepolia network and the contract. The COO-minted contract's

Table 1 Comparison of existing [11, 15, 16] and proposed technique

Features	Existing	[11]	[15]	[16]	Proposed
Decentralization	-	✓	✓	✓	✓
Security	Low	High	High	High	High
Transfer ownership of COO	✓	-	-	-	✓
Transparency	Low	High	High	High	High
Fault tolerant	Low	High	High	High	High
Boost trade at the global level	Low	Low	High	-	High
Cost-effectiveness	Low	Low	Low	Low	Low
Certificate of origin (COO) minted into non-fungible tokens (NFT)	-	-	-	-	✓
Letter of credit (LC) certificate minted into non-fungible tokens (NFT)	-	-	-	-	✓
Updating the location of the goods (via shipping line)	Low	-	✓	✓	✓

where -: absence of feature, ✓: presence of feature

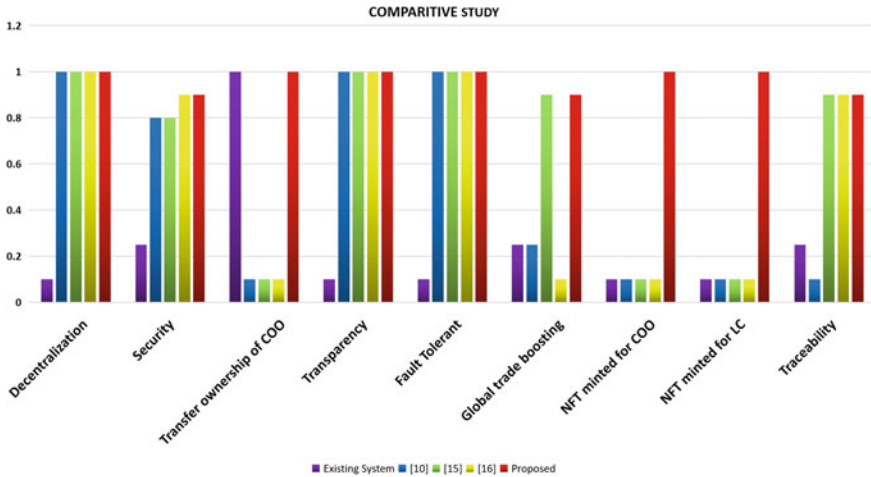


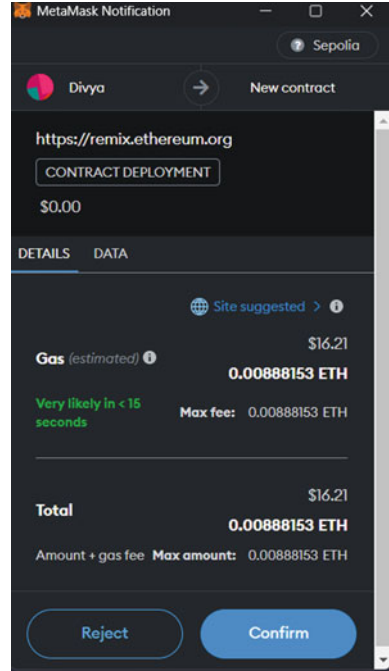
Fig. 4 Comparative analysis of the proposed framework are contrasted with the existing scheme [11, 15, 16]

deployment expense is 0.00888153 ETH. The deployment cost and ether transaction cost are displayed in Fig. 5. This transaction was obtained from the Sepolia testnet network of Etherscan. The first column shows the transaction hash for creating contracts, and the fifth column shows the address for the Metamask account. the final column shows transaction fees.

Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0x33ceea29a34175ba...	Contract Creation	3404179	47 secs ago	0x7A2563...3Fc05DF5	Contract Creation	0 ETH	0.0097381
0xd75cbfb74a24b9c94...	Contract Creation	3404172	2 mins ago	0x7A2563...3Fc05DF5	Contract Creation	0 ETH	0.0094677
0xd44748f9b30da3290...	Contract Creation	3404162	4 mins ago	0x7A2563...3Fc05DF5	Contract Creation	0 ETH	0.00783024
0xa37119f034112671e...	Transfer	3404110	15 mins ago	0x6Cc939...7Ba5F455	0x7A2563...3Fc05DF5	0.04335 ETH	0.000042
0x38202b7b455b823f1...	Contract Creation	3358018	6 days 23 hrs ago	0x7A2563...3Fc05DF5	Contract Creation	0 ETH	0.0088152
0xc5607acc19e1649...	Transfer	3358003	6 days 23 hrs ago	0x6Cc939...7Ba5F455	0x7A2563...3Fc05DF5	0.05136267 ETH	0.000042

Fig. 5 Costs for deploying smart contracts and conducting ether tests

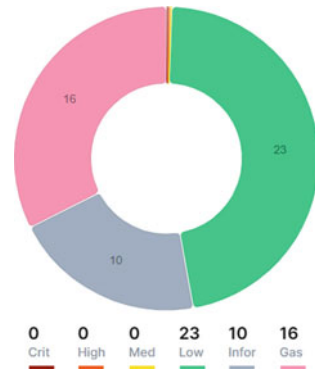
Fig. 6 Snapshot of gas fees paid through metamask



6.3 Performance Evaluation

The suggested framework’s operational costs can be assessed. Calling every smart contract in the operation sequence will yield the total cost. Each operation in Ethereum has a set gas cost. The overall cost of gas is also regulated in this way. The cost of gas is measured in this case using the Etherscan tool. The total cost of the gas can be calculated using this tool. 1 gas costs 2.5×10^{-9} ETH, or 4.6×10^{-6} USD/gas. An explicit blockchain network needs to be developed between nations and parties to reduce cost because the proposed model has a greater cost because many functions are being called. Figure 6 shows the gas fees paid via Metamask for the proposed scheme.

Fig. 7 Pie-chart representation generated using SolidityScan



6.4 Security Analysis

It is a requirement for any smart contract to analyze security flaws. A well-known exploit called Decentralized Autonomous Organization (DAO) calls the function repeatedly without finishing the preceding one, causing a loss of 3.6 million Ethers. SolidityScan is an online tool for evaluating smart contracts security against flaws and vulnerabilities. SolidityScan produces an analysis report that highlights security flaws such as code handling errors and scores, issue count, duration, lines of code, and severity of vulnerabilities found (high/ medium/low). Figure 7 shows a pie-chart representation of the security scan of smart contracts.

6.5 Limitations and Issues

Smart contracts have some restrictions:

1. The immutability of blockchain technology can make it challenging to amend smart contracts due to shifting business conditions.
2. Since smart contracts get transmitted to every node and every transaction is logged for every node, anyone can execute them. There is no confidentiality agreement.
3. It will take a while for smart contracts to be implemented quickly and effectively. Therefore, this procedure limits the usage of smart contracts.

Also, Solidity language is being used to code smart contracts which do not deal with decimals and floating points. This poses an issue in the calculation of tariffs/duties of goods with large quantities as there will be a loss of precision in the final cost that would be calculated. Assuming that the exporter would share in COO number with the importer so that he could make the request for the change in ownership.

6.6 Future Scope

1. Payment: The payments can be made directly through crypto wallets (Metamask wallets) when the conditions are met. The amount on the invoice will be deducted directly from the importer's Metamask wallet to the address of the exporter's Metamask wallet.
2. Tracking of goods: The tracking of the shipments can be automated by the use of the GPS installed in the carrier ships, which will be updated on the blockchain in real time.
3. Handling of decimal values: With the advancement in the versions of Solidity, more accurate calculations of the tariffs/duties can be expected (provided the loss of precision issue existing in the current versions of Solidity gets fixed).

7 Conclusion

In conclusion, the suggested solution would significantly increase the effectiveness and openness of the procedure for customs clearance by automating clearance of customs and enabling the production of COO and LC. The solution will make sure that all parties participating in the procedure can access and verify the relevant information by using smart contracts, eliminating the requirement for human verification and potential inaccuracies. For the development and supervision of the COO and LC, the technique will also offer a secure and impenetrable platform, ensuring the validity and correctness of the documents. Overall, by expediting the customs clearance procedure and lowering the costs and risks involved, this project has a chance to significantly benefit enterprises engaged in international trade.

References

1. Bakari S, Mabrouki M (2017) Impact of exports and imports on economic growth: new evidence from Panama. *J Smart Econ Growth* 2(1):67–79
2. Yaren H (2020) Implementing blockchain technology in the customs environment to support the SAFE framework of standards. *World Customs J* 14(1):127–138
3. Seyoum B (2013) *Export-import theory, practices, and procedures*. Routledge
4. Paul J, Aserkar R (2013) *Export import management*. OUP Catalogue
5. Segers L et al (2019) The use of a blockchain-based smart import declaration to reduce the need for manual cross-validation by customs authorities. ACM, Dubai, United Arab Emirates
6. Gürkaynak G, Yılmaz I, Yeşilaltay B, Bengi B (2018) Intellectual property law and practice in the blockchain realm. *Comput Law Secur Rev* 34(4):847–862
7. Pauletto C (2021) Blockchain in international e-government processes: opportunities for recognition of foreign qualifications. *Res Global* 3:100034
8. Popa I, Belu MG, Paraschiv DM, Marinoiu AM (2015) Best practices in customs procedures. *Amfiteatru Econ J* 17(40):1095–1107

9. Belu MG (2020) Blockchain technology and customs procedures. *Romanian Econ J* 23(78):13–26
10. Belu MG (2019) Application of blockchain in international trade: an overview. *Romanian Econ J* XXII(71)
11. Tyagi Nitin K, Goyal Mukta (2021) Blockchain-based smart contract for issuance of country of origin certificate for Indian customs exports clearance. *Concurrency Computat Pract Exper* 2021:e6249
12. Okazaki Y (2018) Unveiling the potential of blockchain for customs. In: *World Custom Organization*. WCO research paper no. 45
13. Sawhney R, Sumukadas N (2005) Coping with customs clearance uncertainties in global sourcing. *Int J Phys Distrib Logistics Manage* 35(4):278–295
14. McDaniel CA, Norberg HC (2019) Can blockchain technology facilitate international trade? *Mercatus Res Pap*
15. Elmay FK, Salah K, Yaqoob I, Jayaraman R, Battah A, Maleh Y (2022) Blockchain-based traceability for shipping containers in unimodal and multimodal logistics. *IEEE Access* 10:133539–133556
16. Kim S, Kim D (2023) Securing the cyber resilience of a blockchain-based railroad non-stop customs clearance system. *Sensors* 23(6):2914

Probability-Based Load-Distribution Framework: To Reduce Latency in Fog Computing



Isha Dubey, Ekta Singh, Monika, and Deepak Kumar Sharma

Abstract With the advent of smart devices in many areas of industry, the need for fog computing has increased. The purpose of using fog computing is to reduce the time it takes to effectively perform the tasks generated by the smart devices' sensors. In this work, we propose a 5-stage architecture that uses the concept of load balancing to allocate tasks at different tiers. We discuss the case when the same users are in range of multiple master nodes. Assigning tasks to all master nodes at the same time would increase the buffer time of the overall execution. Therefore, we connect via Wi-Fi with master nodes using the concept of a three-way handshake and the threshold time set by the end user. The end user uses the master node prediction algorithm that we propose. The main purpose of using Wi-Fi for wireless networks is that with the advent of 5G networks, our architecture will become faster and more efficient, and our architecture will take advantage of this and become more efficient over time. This algorithm reduces latency by helping the end user determine the specific master node before submitting their task for execution, thereby also helping to reduce the number of waiting requests at the master nodes. Previous articles required more efficient load balancing algorithms, which we implemented by implementing our own architecture. Our multi-layer architecture reduces latency at the master node by further reallocating the tasks to the sub-master nodes based on the tasks' QoS values. Other fog nodes perform the task along with the volunteer nodes. We have demonstrated the algorithm for each individual component of our architecture.

1 Introduction

The proliferation of smart devices has boosted the manufacturing sector and the scientific community's appreciation of the brilliance of prediction by proactively taking preventative actions to save an abundance of assets. The processing workload for a large amount of data has, however, increased exponentially as a result of the

I. Dubey (✉) · E. Singh · Monika · D. K. Sharma
Department of Information Technology, Indira Gandhi Delhi Technical University for Women,
Delhi, India
e-mail: ishadubey200@gmail.com

tremendous proliferation of terminal devices. There is an increasing amount of data that needs to be processed, saved, and examined as more devices get connected to the Internet. This can put a strain on existing computing and storage infrastructure and may require organisations to invest in new technologies, such as fog computing, to handle the increased data volumes.

Fog computing is a decentralised computing model that brings computational and storage capabilities in close proximity to the devices that rely on them, including IoT devices like sensors and cameras. This contrasts with conventional cloud computing, where these resources are typically located in centralised data centres that are often far away from edge devices. These devices generate a huge amount of data, and sending all of this data to the cloud for processing and storage can be prohibitively expensive and time-consuming. Fog computing allows for much of this data to be processed and stored closer to the edge, reducing the amount of bandwidth and latency that are required.

Another reason for the growing importance of fog computing is the need for real-time processing and decision-making. In many cases, the information produced by IoT devices needs to be processed and analysed quickly in order to take appropriate action. For example, in a factory setting, sensors might be used to monitor the condition of equipment, and if a problem is detected, the fog computing system can quickly analyse the data and send an alert to maintenance personnel. This is much faster than sending all of the data to the cloud, where it would need to be processed and analysed before any action could be taken. Overall, fog computing offers a number of benefits over traditional cloud computing, including reduced latency, improved scalability and flexibility, and better support for real-time decision-making.

1.1 Problem Statement

After an extensive literature review, we enumerated the limitations of various research papers and tried to work on them for our research paper. Previous papers needed more efficient load balancing algorithms, which we have implemented by implementing our own architecture. Our multi-layer architecture reduces latency at the master node by allocating the tasks further to the sub-master nodes based on the QoS values of the tasks. Further fog nodes, along with the volunteer nodes, carry out the task. We have demonstrated the algorithm for each and every component of our architecture.

1.2 Motivation

In the current research, we propose an architecture and algorithms in fog computing to reduce latency, which are motivated by past efforts and their shortcomings.

There are numerous potential advantages to using fog computing to reduce latency, such as improved system performance, reduced network congestion, and lower

energy consumption. Some of the main beneficiaries of fog computing in these cases may include:

End users: By reducing the amount of time it takes to process data and deliver it, fog computing can improve the user experience. This is especially useful in applications requiring real-time processing, such as online gaming or video streaming.

Service Providers: Fog computing can help service providers deliver more reliable and efficient services to their customers by reducing latency and improving system performance. This can improve customer satisfaction and retention while also giving you a competitive advantage over other providers.

IoT devices: Because IoT devices frequently have limited processing power and storage capacity, fog computing can be especially beneficial. Fog computing can help IoT devices operate more efficiently by moving data processing to the network's edge.

Overall, fog computing has the potential to benefit a wide range of stakeholders by improving system performance and reducing latency in applications and services that require real-time processing.

1.3 Main Contributions

The main contributions to our work are listed below:

- We propose an architecture that will enable overloaded fog nodes to get real-time load balancing in order to reduce the communication overhead between the fog nodes.
- We propose an algorithm for the end users that allows them to send their requests to the master fog node based on their distance and the number of pending requests.
- We propose another algorithm that allows the master fog node to allocate the requests to the fog and volunteer nodes and carry out priority-based scheduling on the requests in the waiting queue.
- The efficacy of the system was assessed through a comprehensive evaluation of its performance with respect to various parameters like execution time and network usage.

1.4 Organisation

In Sect. 2 of our report, we explore previous studies that focus on reducing latency and distributing load in the fog layer. Section 3 introduces our proposed system model, architecture, and various algorithms. Section 4 provides details about the simulation setup, while Sect. 5 presents the obtained results. Section 6 delves into the paper's scope and limitations. Lastly, Sect. 7 concludes the report and outlines future research directions.

2 Related Work

Recent years have seen the emergence of many approaches for lowering the latency of fog computing. This article introduces a novel method for achieving real-time load balancing in overloaded fog nodes by implementing a distributed scheme that minimises transmission costs between them [1]. Furthermore, a fog registration centre (FRC) is proposed to authenticate and establish trust among the fog nodes, facilitating the creation of a dependable load-sharing network. The protocol's performance is evaluated using metrics such as latency per node, average latency, and the successful completion rate of jobs. In order to comprehend, assess, and model service delays in IoT-fog-cloud application situations, this study presented a universal framework for delay [2]. For the IoT-fog-cloud application situations, it also established a minimising policy and an analytical model to comprehend, assess, and model service latency. It described the design and prototype implementation of Follow me Fog (FMF) and contrasted how Follow me Fog (FMF) was unique from the pre-existing frameworks [3]. In order to enhance the computing capabilities at the network's edge, this paper offers Volunteer Support Fog Computing (VSFC), a new computing paradigm for Internet of Things applications that combines Fog Computing (FC) with Volunteer Computing (VC) [4]. The fog device effectively runs time-sensitive jobs across nearby volunteer devices instead of transmitting them to the federated cloud.

Recently, a number of fog computing strategies have been developed to reduce traffic between the cloud and users in order to conserve network bandwidth, cut down on energy use, and employ resources appropriately [5]. These strategies offer scalable data storage for enhancing processing and support capabilities. To properly divide the load among the available fog nodes and speed up job processing, a load balancing method was devised [6]. In this method, the control nodes are designated as dependable networking resources such as routers, switches, and base stations. On the basis of the average response time, the findings are compared. To guarantee that applications' quality of service (QoS) complies with service delivery deadlines and to maximise resource utilisation in the fog environment, more latency-related research was advised [7]. An iFogSim-simulated fog environment was also set up to evaluate the suggested regulation.

To control and optimise the overall fog network, edge frameworks with intelligence have been incorporated into several research investigations [8]. It has been demonstrated that human-driven and machine-driven intelligence can result in less jitter and delay. Later, reinforcement learning and neural network evolution approaches were combined with fuzzy interference in a recommended intelligent FC analytical model and method for data packet allocation and selection in an IoT-FC environment [9]. Python and the simulators Spyder and iFogSim (Net-Beans) are used to evaluate the strategy. The experience of mobile users may drastically deteriorate if a large number of offloading users compete for the limited connectivity and compute resources [10]. A strategy based on mixed integer non-linear programming, probability analysis, queuing theory, and convex optimisation theory was put forward

to address this issue [11]. A significant obstacle for edge computing is the provision of computer services with low service blockage and latency. A study suggested Cooload as a solution to this problem. It is a cooperative system between two data centres located at the network's edge that trades general computing requests when one of them momentarily becomes overloaded. There was also an established mathematical model that illustrates how the service blockage probability and service delay can both be lowered simultaneously.

Several experiments revealed that it was critical to be able to spread radio and compute resources in order to accommodate as many user demands as possible, reduce SCC power usage, and simplify the process [12]. They advised dividing the resource allocation process into two crucial steps. An initial allocation of resources was made to service SCs in line with a certain scheduling rule based on metrics. In a subsequent step, computer clusters were built while following a scheduling guideline and optimising in the best way possible for unmet demands [13]. The authors of the proposed study were the first to formally formulate and examine the workload allocation-power consumption-delay trade-off problem in the cloud-fog computing system. Additionally, they developed an approximate solution that divides the main issue into three smaller issues with related subsystems, each of which can be resolved on its own. The usefulness of our scheme was rigorously tested in this study using extensive simulations, and the findings indicated that the fog can greatly supplement the cloud while having significantly shorter communication latency. Here, a precise energy consumption model is developed and applied to fog nodes using the capabilities of intelligent equipment [14]. A multi-agent system is incorporated into the proposed fog node-based energy-aware load balancing and scheduling approach to increase the smart factory's negotiation autonomy. When compared to the cloud platform, fog computing considerably reduces the latency of instruction transmission, and the deployment of fog nodes meets the real-time requirements for industrial dynamic order analysis and equipment scheduling.

Here, a new monitoring system is developed that checks for resources that have the ability to disrupt planning [15]. Monitoring disruptions is done using a federated machine learning model. When a system disruption is detected, rescheduling of the tasks takes place to enhance efficiency and reduce latency [16]. In numerous instances, the fog nodes and the VMs they contain may encounter failures and it is crucial to have an efficient repair system and a dedicated repair facility to promptly address these issues and uphold the desired level of Quality of Service (QoS) for the system. Here, the authors have developed a repair system for fog nodes and Virtual machines (VMs) within an IoT-fog system in smart cities. This research presents an orchestration and data processing framework for distributed fog clouds for creating on-demand process engine data flow (PEDF) across numerous devices by leveraging the semantics of both cloud and fog platforms [17]. Here, the authors have discussed a three-tier vehicular fog network, which is further divided into single and multi-region offloading decisions [18]. To resolve the single-region offloading choice, they determine the preliminary offloading decisions for consumers in each RSU coverage area. Knapsack is used to solve the multi-region resource allocation problem, resulting in optimal resource allocation.

3 Proposed System Model and Architecture

We have proposed a five-layer architecture for reducing latency through load balancing in fog computing. The layers of the architecture represented in the figure are explained in detail below (Fig. 1):

- End users: They consist of devices that request real-time services from the master nodes. They also store the details of their closest master nodes. They are the driving components of this architecture. Below are two types of users:
 - Shared users: Users that are in the range of multiple master nodes need to determine which master node must be selected to carry out their task in a more efficient way.
 - Terminal users: Users that are in the range of only one master node. They send their tasks to that master node, making it even more necessary for the shared nodes to judiciously choose their respective master nodes to reduce the buffer time for task execution.

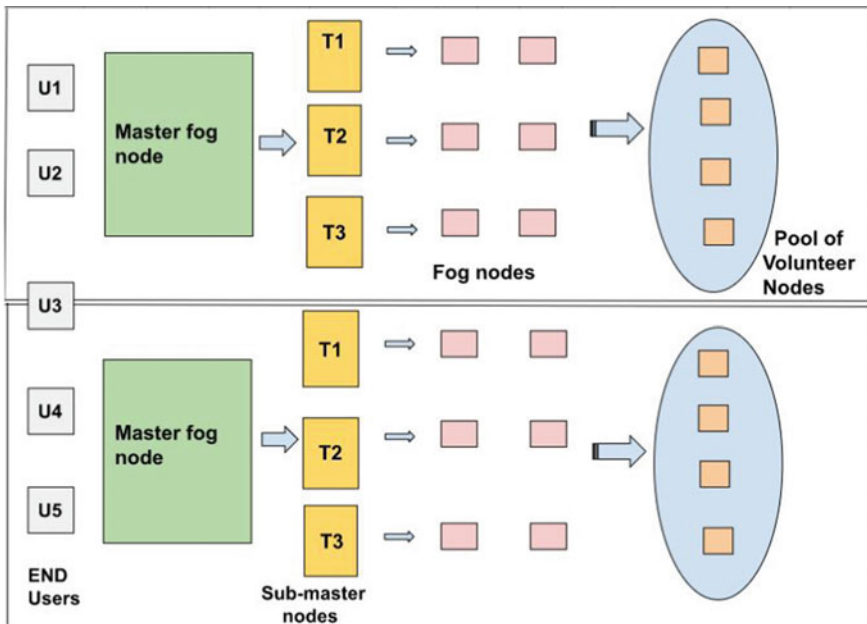


Fig. 1 Proposed architecture—this image depicts the architecture designed by us to reduce latency. It consists of five components, mainly the users, master fog nodes, sub-master fog nodes, fog nodes, and volunteer nodes. The user sends the task to the master fog node using the master node prediction algorithm, and then the data is further sent to the sub-master node based on the type of incoming data. The data is further processed by the fog node or volunteer node based on the traffic.

- Master fog node: This is the node that is closest to the end user. The information that needs to be processed is sent by the user to the master fog node. The master fog node is responsible for carrying out the tasks mentioned below:
 - Segregate the tasks based on their priorities and send them to their respective sub-master nodes.
 - The remaining requests (which couldn't be sent to sub-master nodes because the capacity got full) are sent to the waiting queue.
 - The tasks are prioritised in the waiting queue based on the QoS values.
 - If the neighbouring master fog nodes are available, then the waiting requests are sent to the neighbouring master fog node.
 - The popularity index (PI) of the master node determines the number of nodes in its range.
- Sub-master nodes: They are responsible for sending out the requests to the fog nodes. But each sub-master node is responsible for a particular type of data. For example:
 - T1 processes video
 - T2 processes audio
 - T3 processes text
- Fog nodes under the sub-master nodes have the capability to process only the type of data type that is supported by their sub-master nodes.
- Volunteer nodes are capable of processing each data type and are used for processing when the fog nodes under a particular master node exhaust.
- End user + master fog node + sub-master node + fog nodes + volunteer nodes = cluster

3.1 End User-Master Node Identification Phase

End users determine the closest master nodes by broadcasting their signals and keeping a record of the master nodes that reply within a predefined threshold time. Acknowledgements received after the threshold time aren't recorded. After accepting and updating the information of the master node, the end user sends the acceptance to the master node. As a result, the popularity index of the master node increases (Figs. 2 and 3).

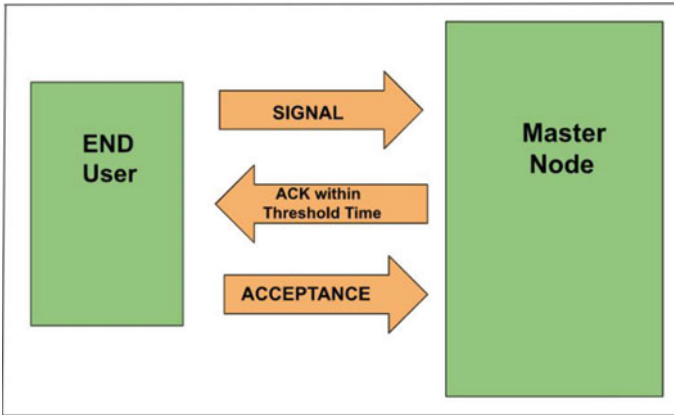


Fig. 2 Establishing a connection between the end user and the master node

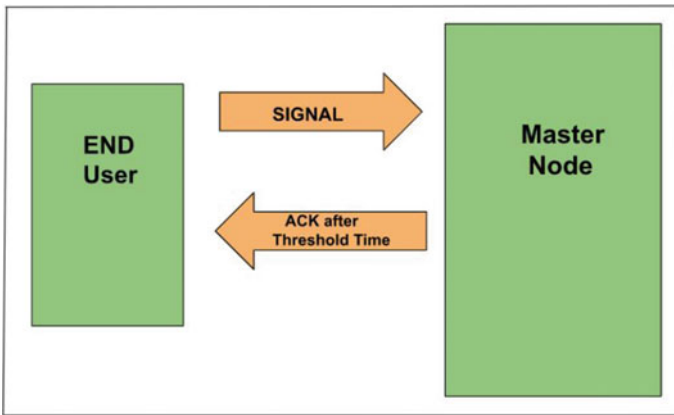


Fig. 3 When the end user receives the acknowledgement after the threshold time, the connection is not established

3.2 Proposed Algorithm 2: Selection of the Best Master Node for an End User (Master Node Prediction Algorithm)

The challenge that this concept of probability overcomes is that all the nodes have equal probability of sending their tasks to the master node within their reach. For example, if all the seven nodes inquire about the number of requests from their respective master node, then M1 would send the number of requests to be 5, and as a result, all the nodes would send their tasks to M1, thus increasing the buffer to 12 (5 + 7), which is now apparently way less than M2 (with the buffer standing at 5). (Some of them could have sent it to M2 instead.) Due to this, buffer time

would increase, increasing the overall latency. To overcome this problem, we apply conditional probability to the nodes that are in the range of multiple master nodes.

The probability that any event A would happen after an event B in relation to A has already happened is known as conditional probability.

We are using Bayes' Theorem in our model.

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} \quad (1)$$

where

$P(H)$ = The probability of H occurring

$P(H | T)$ = The probability of H given T

$P(H \cap T)$ = The probability of both H and T occurring

This helps the node choose the more appropriate master node with a lesser number of requests and, thereby, less time for execution. For example, if node 4 is in the vicinity of both M1 and M2, with 5 and 8 requests, respectively, then it would use conditional probability to decide which master node to pursue.

With reference to the above image, we find out the probability of any user's task being to go to their closest master nodes based on the empty slots. We also cover the concept of shared nodes (nodes that are in the range of multiple master nodes) and how they determine the most efficient master node to carry out their task. The tasks of terminal nodes are also taken into consideration. We assume that all the master nodes have K slots that are used for carrying out the tasks sent by the end user. The waiting list of the master node determines the tasks that are yet to be executed. So at any given time, the number of empty slots (ES) would be: $ES = K - (\text{number of waiting requests})$

Assuming the popularity index of the master nodes of $M1$, $M2$ and $M3$ be $PI1, PI2$ and $PI3$.

Let $PI1 = v1$, $PI2 = v2$ and $PI3 = v3$. Let the number of terminal nodes be N . Then the number of shared nodes would be $(PI-N)$. Since we are assuming that the terminal nodes send their tasks to only one master node, they add them to the buffer list of the master node. **Therefore, total waiting requests = previous requests + N (no. of terminal nodes)**

Working of the above algorithm can be understood using an example. Figure 4 shows that user1 is in the range of 3 master nodes, whereas user2 is in the range of 2 master nodes. The empty slots of the master nodes can be expressed as $ES(M1)$, $ES(M2)$ and $ES(M3)$ respectively (Fig. 5).

Let the total number of empty slots be

$$TES = ES(M1) + ES(M2) + ES(M3)$$

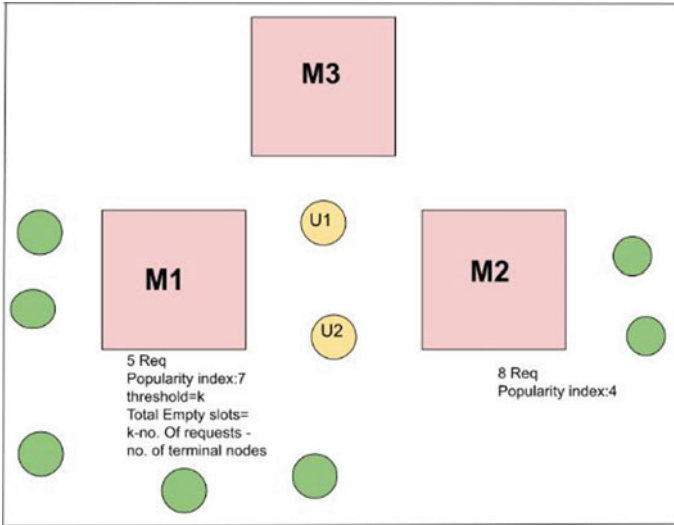


Fig. 4 This figure represents three master nodes and end users in their vicinity. The green circles represent end users in the range of one master node, whereas the yellow circles represent end users in the range of multiple master nodes

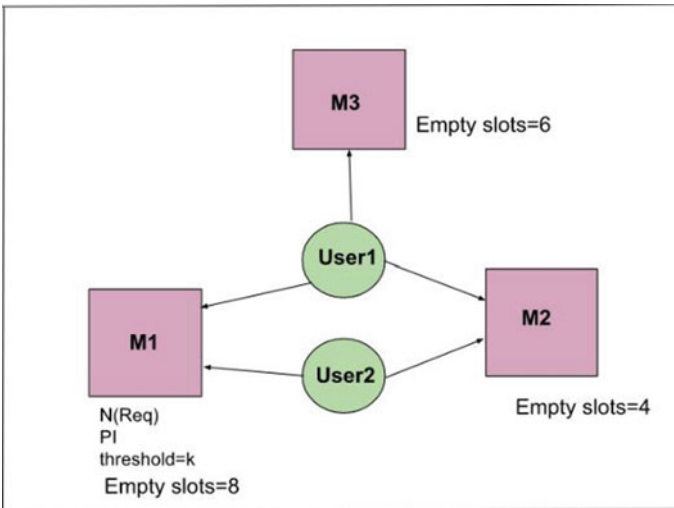


Fig. 5 This represents the master nodes and nodes in their vicinity, along with the pending requests of the master nodes and the empty slots present in the master nodes after the tasks of the terminal nodes have been taken into consideration

The probability of a task of common nodes getting accepted by $M1$ is given by:

$$P(M1) = \frac{ES(M1)}{TES}$$

The probability of a task of common nodes getting accepted by $M2$ is given by:

$$P(M2) = \frac{ES(M2)}{TES}$$

The probability of a task of common nodes getting accepted by $M3$ is given by:

$$P(M3) = \frac{ES(M3)}{TES}$$

Below, we have shown the numerical for the same.

For user1: In the above situation, we assume that user1 is in the range of 3 master nodes, i.e. it can send its task to any one of the master nodes for execution.

$P(M_i)$ represents the i th master node to which the task of user 1 would be sent.

$$P(M1) = \frac{8}{8 + 6 + 4} \quad (2)$$

$$P(M2) = \frac{4}{18} \quad (3)$$

$$P(M3) = \frac{6}{18} \quad (4)$$

$$P(M1) + P(M2) + P(M3) = 1 \quad (5)$$

For user2: In the above situation, we assume that user2 is in the range of 2 master nodes, i.e. it can send its task to any one of the master nodes for execution. $P(M_i)$ represents the i th master node to which the task of user2 would be sent.

$$P(M1) = \frac{8}{8 + 4} \quad (6)$$

$$P(M2) = \frac{4}{12} \quad (7)$$

Case (I) If the task of user1 goes to $M1$, then the probability of the task of user2 going to $M1$ $P(T|H) = 7/11$ (empty slots in $M1$ gets reduced to 7 and empty slots in $M2$ are 4)

$P(H) = 8/18$ (Represents the probability of task of user 1 going to $M1$)

$P(T) = 8/12$ (Represents the probability of task of user 2 going to $M1$)

Using Bayes' Theorem: The probability of the task of user1 going to $M1$ after the task of user2 has already gone to $M1$.

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} = \frac{14}{33} = 0.42 \quad (8)$$

Case (II) If user1 goes to $M1$, then the probability of user2 going to $M2$.

$P(T|H) = 4/11$

$P(H) = 8/18$ (Represents the probability of the task of user 1 going to $M1$)

$P(T) = 4/12$ (Represents the probability of the task of user 2 going to $M2$)

$P(H|T)$ represents the probability of the task of user1 going to $M1$ when user2 has already gone to $M2$. Using Bayes' Theorem,

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} = \frac{12}{25} = 0.48 \quad (9)$$

Case (III) If user1 goes to $M2$, then the probability of user2 going to $M1$.

$P(T|H) = 8/11$

$P(H) = 4/18$ (Represents the probability of the task of user 1 going to $M2$)

$P(T) = 8/12$ (Represents the probability of the task of user 2 going to $M1$)

$P(H|T)$ represents the probability of the task of user1 going to $M2$ when user2 has already gone to $M1$. Using Bayes' Theorem,

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} = \frac{6}{25} = 0.24 \quad (10)$$

Case (IV) If user1 goes to $M2$, then the probability of user2 going to $M2$

$P(T|H) = 3/11$

$P(H) = 4/18$ (Represents the probability of the task of user 1 going to $M2$)

$P(T) = 4/12$ (Represents the probability of the task of user 2 going to $M2$)

$P(H|T)$ represents the probability of the task of user1 going to $M2$ when user2 has already gone to $M2$. Using Bayes' Theorem,

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} = 0.18 \quad (11)$$

Case (V) If user1 goes to $M3$ then the probability of user2 going to $M1$ $P(T|H) = 8/12$

$P(H) = 6/18$ (Represents the probability of the task of user 1 going to $M3$)

$P(T) = 8/12$ (Represents the probability of the task of user 2 going to $M1$)

$P(H|T)$ represents the probability of the task of user1 going to $M3$ when user2 has already gone to $M1$.

Using Bayes' Theorem,

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} = \frac{1}{3} = 0.33 \quad (12)$$

Case (VI)

Using Bayes' Theorem, if user1 goes to $M3$ then the probability of user2 going to $M2$ $P(T|H) = 4/12$

$P(H) = 6/18$ (Represents the probability of task of user 1 going to $M3$)

$P(T) = 4/12$ (Represents the probability of task of user 2 going to $M2$)

$P(H|T)$ represents the probability of task of user1 going to $M3$ when user2 has already gone to $M2$.

$$P(H|T) = \frac{P(H)P(T|H)}{P(T)} = \frac{1}{3} = 0.33 \quad (13)$$

3.3 Proposed Algorithm 3: Allocating the Tasks from the Master Node to the Sub-master Nodes

Tasks: 1 to 100 Let the number of fog nodes under each sub-master node be N .

The tasks are sorted on the basis of their QoS values, i.e. the task with higher priority is at the top of the list. When there is availability of a fog node, this algorithm allocates tasks queued in the waiting list to their respective fog nodes.

```

1:  $W_l$ 
2: for  $t$  from 1 to 100 do
3:   if  $t ==$  "video" then
4:     introduce new  $V_i$ ;
5:     if  $\sum V_i \leq N$  then
6:        $t \rightarrow T1$ 
7:     else
8:        $t \rightarrow W_l$ 
9:     end if
10:  end if
11:  if  $t ==$  "audio" then
12:    introduce new  $A_i$ ;
13:    if  $\sum A_i \leq N$  then
14:       $t \rightarrow T2$ 
15:    else
16:       $t \rightarrow W_l$ 
17:    end if
18:  end if
19:  if  $t ==$  "text" then
20:    introduce new  $T_i$ ;
21:    if  $\sum T_i \leq N$  then

```

```

22:      $t \rightarrow T3$ 
23:     else
24:          $t \rightarrow W_l$ 
25:     end if
26: end if
27: end for

```

3.4 Proposed Algorithm 4: Processing at the Fog Node

As soon as the tasks get processed at the fog node, we decrease the total number of V_i , A_i or T_i depending on the type of task so that we have space for new incoming tasks.

```

1: if task is completed then
2:   if  $t == \text{"video"}$  then
3:     Decrease a  $V_i$ ;
4:   end if
5:   if  $t == \text{"audio"}$  then
6:     Decrease an  $A_i$ ;
7:   end if
8:   if  $t == \text{"text"}$  then
9:     Decrease a  $T_i$ ;
10:  end if
11: end if

```

3.5 Proposed Algorithm 5: Allocating the Tasks of the Waiting List to Volunteer Nodes

Since the waiting list uses priority based scheduling, the tasks are sorted in the list based on their QoS value. V_n volunteer nodes.

```

1: for tasks in  $W_l$  in the range of 1 to  $k$  do
2:    $i = 0$ ;
3:   while  $i \leq V$  and not( $W_l$  is empty) do
4:      $tasks \leftarrow V_n$ 
5:      $i \leftarrow i + 1$ ;
6:   end while
7:   if  $W_l$  is empty then
8:     return;
9:   end if
10: end for

```

Table 1 Our model configurations

No. of users	8
No. of Master nodes	3
Processing power of a fog node	4000 MIPS
Processing power of cloud	40,000 MIPS
Cloud RAM	40000
Cloud busy power	16 * 103 W
Cloud idle power	16 * 83.26 W
Fog busy power	87.54 W
Fog idle power	82.45 W

4 Simulation Setup

The efficacy of the suggested framework was assessed using iFogSim, which is built on top of the CloudSim simulator. iFogSim is a popular simulation platform for simulating fog computing. We simulated the proposed user architecture in order to compare the outcomes to the cloud and SSLB designs. We have compared the execution times of all three, keeping the number of nodes the same. The fog nodes under each sub-master node are assigned a fixed number of volunteer nodes that are capable of processing every type of incoming data. When all the fog nodes are exhausted due to heavy traffic, the task is forwarded to the volunteer node instead of the cloud. Since the fog nodes assigned under each sub-master node are capable of processing only one type of data, this improves the performance of execution. Experimental Setup (Table 1).

5 Results

- **Latency:** The graphs Figs. 6, 7 and 8 represents the execution time of our proposed architecture over 45 iterations. In the graphs, we have compared the total execution time taken by the cloud, the SSLB architecture and our architecture. The iterations have been done on three different devices with the specifications mentioned in the table. The spikes occurring in the graph represent the increase in traffic load at that particular iteration. Here, we can clearly see that our proposed architecture and algorithm have outperformed both SSLB and the cloud in terms of execution time. Our model has reduced the delay by 21% when compared to the traditional cloud-based model (Table 2).
- **Network Usage:** From Fig. 9, We can easily observe that network usage is increasing as the number of users increases. Our network utilisation is directly related

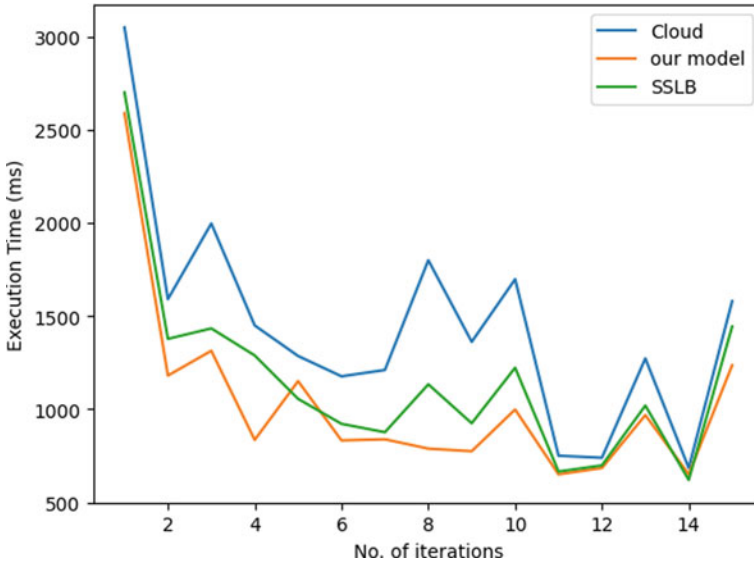


Fig. 6 Comparative analysis of execution time taken by our model, cloud architecture and SSLB (an existing fog model architecture) from device 1

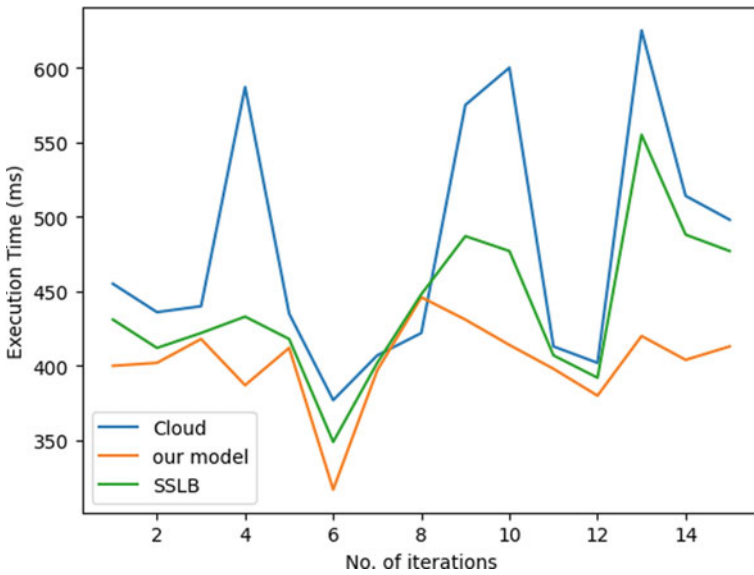


Fig. 7 Comparative analysis of execution time taken by our model, cloud architecture and SSLB (an existing fog model architecture) from device 2

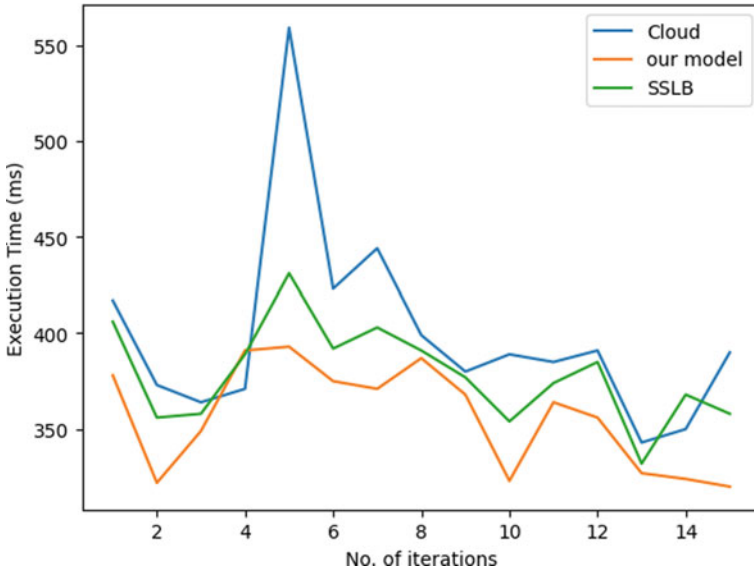


Fig. 8 Comparative analysis of execution time taken by our model, cloud architecture and SSLB (an existing fog model architecture) from device 3

Table 2 Devices used for testing

Device	RAM	OS	Processor
1	4 GB	Windows 11	Intel i3
2	12 GB	Windows 10	Intel i5
3	16 GB	Windows 11	Intel i5

to the number of system users. When there is heavy traffic, the master node prediction algorithm evenly distributes the traffic and allocates the task to different master nodes, thereby, increasing the effective utilisation of the network. Similarly, when the sub-master node gets overloaded with the tasks, tasks are sent to the neighbouring sub-master nodes for execution. This helps reduce the overall time taken for the execution of the task. When the fog nodes that are allocated to each sub-master node for a particular type of incoming task are forwarded to the volunteer node. Therefore, the graph justifies the increase in network usage as the number of users increases.

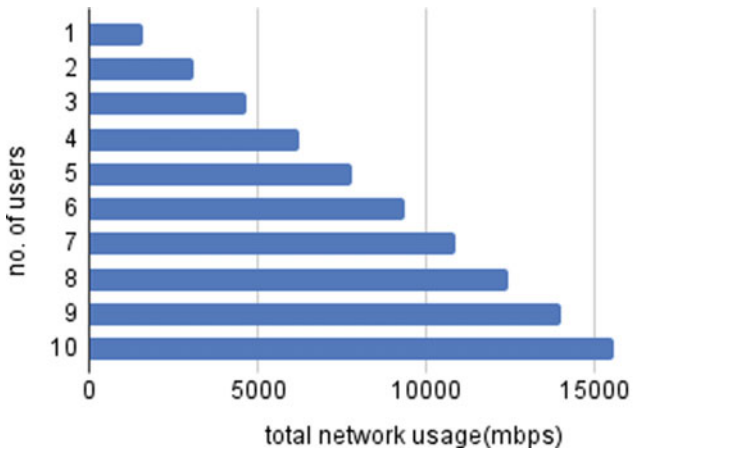


Fig. 9 Network usage with respect to users

6 Scope and Limitations

Only static users or devices are intended to be supported by this proposed methodology. In the future, it can be enhanced by adding mobile fog computing to support the mobility of devices. To encourage mobile users to offload their computing activities, the radio access networks of cellular systems can be connected to the fog nodes. At this point in time, we are not analysing and studying latency and power trade-offs, but we would like to work on and expand upon this in the future.

7 Conclusion and Future Work

In this paper, we present a novel 5-layer architecture that employs load balancing to efficiently distribute tasks across different levels. We address the issue of multiple master nodes being in the same user range, which can lead to longer overall execution times if tasks are allocated to all master nodes simultaneously. To mitigate this issue, we propose a solution that utilises a three-way handshake connection established over Wi-Fi, along with a user-defined threshold time. We also introduce the Master Node Prediction algorithm, which enables the end user to determine the optimal master node for task execution before sending the task, thus reducing latency and the number of waiting requests at master nodes. Our multi-layer architecture further reduces latency by allocating tasks to sub-master nodes based on the Quality of Service (QoS) values of the tasks. Additionally, we leverage fog and volunteer nodes to further carry out tasks in our proposed architecture. This decreased our execution time by 21% as compared to the traditional model.

The proposed methodology is intended to support static users or devices. To expand its capabilities to support mobile devices, mobile fog computing can be added in the future. To persuade mobile users to offload their computing tasks, the radio access networks of cellular systems can be connected with fog nodes. Currently, our focus is not on analysing the trade-off between latency and power, but we intend to investigate this in future work. We plan to explore ways to conserve energy and promote green fog computing.

References

1. Mazumdar N, Nag A, Singh JP (2021) Trust-based load-offloading protocol to reduce service delays in fog-computing-empowered IoT. *Comput Electr Eng* 93:107223. ISSN 0045-7906
2. Yousefpour A, Ishigaki G, Jue JP (2017) Fog computing: towards minimizing delay in the Internet of Things. In: *IEEE international conference on edge computing (EDGE)*. IEEE, pp 17–24. <https://doi.org/10.1109/IEEE.EDGE.2017.12>
3. Bao W et al (2017) Follow me fog: toward seamless handover timing schemes in a fog computing environment. *IEEE Commun Mag* 55(11):72–78. <https://doi.org/10.1109/MCOM.2017.1700363>. Nov.
4. Ali B, Adeel Pasha M, Islam SU, Song H, Buyya R (2021) A volunteer-supported fog computing environment for delay-sensitive IoT applications. *IEEE Internet Things J* 8(5):3822–3830. <https://doi.org/10.1109/JIOT.2020.3024823>
5. Caiza G, Saeteros M, Oñate W, Garcia MV (2020) Fog computing at industrial level, architecture, latency, energy, and security: a review. *Heliyon* 6(4):e03706. ISSN 2405-8440
6. Manju AB, Sumathy S (2019) Efficient load balancing algorithm for task preprocessing in fog computing environment. In: Satapathy S, Bhateja V, Das S (eds) *Smart intelligent computing and applications*. Smart innovation, systems and technologies, vol 105. Springer, Singapore
7. Mahmud M, Kotagiri R, Rajkumar B (2018) Latency-aware application module management for fog computing environments. *ACM Trans Internet Technol* 19. <https://doi.org/10.1145/3186592>
8. La QD, Ngo MV, Dinh T, Quek TQS, Shin H (2018) Enabling intelligence in fog computing to achieve energy and latency reduction. <https://doi.org/10.1016/j.dcan.2018.10.008>
9. Saurabh S, Fadzil HM, Khalid K, Tang L, Azlan A (2019) An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment. *PLOS One* 14:e0224934. <https://doi.org/10.1371/journal.pone.0224934>
10. Li Q, Wang S, Zhou A, Ma X, Yang F, Liu AX (2022) QoS driven task offloading with statistical guarantee in mobile edge computing. *IEEE Trans Mobile Comput* 21(1):278–290. <https://doi.org/10.1109/TMC.2020.3004225>
11. Beraldi R, Mtibaa A, Alnuweiri H (2017) Cooperative load balancing scheme for edge computing resources. In: *Second international conference on fog and mobile edge computing (FMEC)*, pp 94–100. <https://doi.org/10.1109/FMEC.2017.7946414>
12. J. Oueis ECS, Barbarossa S (2015) The fog balancing: load distribution for small cell cloud computing. In: *2015 IEEE 81st vehicular technology conference (VTC Spring)*, pp 1–6. <https://doi.org/10.1109/VTCSpring.2015.7146129>
13. Deng R, Lu R, Lai C, Luan TH (2015) Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing. In: *IEEE international conference on communications (ICC)*, pp 3909–3914. <https://doi.org/10.1109/ICC.2015.7248934>
14. Wan J, Chen B, Wang S, Xia M, Li D, Liu C (2018) Fog computing for energy-aware load balancing and scheduling in smart factory. *IEEE Trans Industr Inform* 14(10):4548–4556. <https://doi.org/10.1109/TII.2018.2818932>. Oct.

15. Brik B, Messaadia M, Sahnoun M, Bettayeb B, Benatia MA (2022) Fog-supported low-latency monitoring of system disruptions in industry 4.0: a federated learning approach. *ACM Trans Cyber-Phys Syst* 6(2):23. <https://doi.org/10.1145/3477272>
16. Goswami V, Sharma B, Patra SS, Chowdhury S, Barik RK, Dhaou IB (2023) IoT-fog computing sustainable system for smart cities: a queueing-based approach. In: 2023 1st international conference on advanced innovations in smart cities (ICAISC), Jeddah, Saudi Arabia, pp 1–6. <https://doi.org/10.1109/ICAISC56366.2023.10085238>
17. Lan D, Liu Y, Taherkordi A, Eliassen F, Delbruel S, Lei L (2021) A federated fog-cloud framework for data processing and orchestration: a case study in smart cities. In: Proceedings of the 36th annual ACM symposium on applied computing (SAC'21). Association for Computing Machinery, New York, NY, USA, 729–736. <https://doi.org/10.1145/3412841.3444962>
18. Yang Y, Cheng W, Wang J (2022) Computation offloading for latency reduction in regionalized hierarchical vehicular fog network. In: GLOBECOM 2022—2022 IEEE global communications conference, Rio de Janeiro, Brazil, pp 5807–5812. <https://doi.org/10.1109/GLOBECOM48099.2022.10001703>

Edge-Graph Convolution Network: An Intrusion Detection Approach for Industrial IoT



Nilutpol Bora and Anamika Chauhan

Abstract Cyber-attacks on Industrial IoT systems can result in severe consequences such as production loss, equipment damage and even human casualties, and hence, security is of utmost concern in this application of IoT. This paper presents an approach for network security, intrusion detection that utilizes the spatial attributes of a network. For this graph-based neural network has been used that was seen promising in modelling complex relationships between graphical entities, making them a suitable approach for IDS in interconnected systems. Our approach leverages a graph representation of network traffic that is used as an input for neural network through the use of convolution operation. Our approach makes use of flow features of the network in relation with the neighbouring flows in contrast to other machine learning models that use flow features independent to each other. This work has been evaluated on Edge-IIoT 2022, dataset and compared with existing well-known machine learning methods. The results show that our approach achieved average 5.49% improved F_1 -score, compared with other standard existing methods with our model having the highest F_1 -score of 0.996.

Keywords Graph convolution · IIoT · IDS · GNN

1 Introduction

Industrial Internet of Things is being a system of devices interconnected through a communication channel serving the purpose of intelligent decisions optimizing performance of an industrial ecosystem. This network like any other has the risk of a cyber-attack, which affects the optimal utilization of the resources, and evidently, the need to detect such disasters from happening proper measures is to be placed.

N. Bora (✉) · A. Chauhan

Department of Information Technology, Delhi Technological University, New Delhi, Delhi
10042, India

e-mail: pnilut@gmail.com

A. Chauhan

e-mail: anamika@dtu.ac.in

The initial step to securing a network is to prepare for any adversaries and detect any suspicious activities in the system, this detection of such sceptical incidents is known as intrusion detection, and the set of procedures are known as the intrusion detection systems. This study focuses on the detection of attacks occurring in a network of Industrial Internet of Things by proposing a methodology that can identify anomalies in the network among the benign traffic. Studies on intrusion detection systems are abundantly available, ranging from traditional signature and anomaly-based, simple machine learning to deep learning-based techniques. Further improvement includes a combination of multiple methods to improve the intrusion detection performance. Although one key point was noticed in the majority of papers was the use of network flow records independent to each other to identify the inference based on the features of the individual record, making no assumption of the relation among the records.

Machine learning and deep learning methods have evolved tremendously in the past few decades to be able to give good results using only this assumption, although without taking into consideration the relation among the different flow records, the spatial characteristics of the network can tremendously help in identifying underlying patterns of a newer generation of attacks. The knowledge of the network flows gives better understanding of how the relationships among the flow define an attack pattern, as the flow in the network inherently characterizes many attacks.

1.1 Research Contributions

The main contributions of this paper are as follows:

- We have proposed a methodology to utilize the network traffic datasets that are generally available as flow-based datasets as a graph object, to extract the spatial features of the network. This helped us in utilizing hidden patterns from the network traffic graph that can help us better identify an attack from a benign network flow.
- The graph convolution model from Deep Graph Library does not support edge classification, for this we have proposed a modified graph convolution model to use edge features in its learning as well as produce scores for each edge connection to classify them.
- We have demonstrated the effectiveness of our proposed methodology through experimental evaluations over benchmark network datasets Edge-IIoT (2022) that could simulate networks relevant to today's world.

1.2 Paper Structure

This paper is further organized ahead describing the various papers studied on network IDS which helped us to better understand the problem in Sect. 2, regarding

the integration of graphs for improving the performance of the intrusion detection systems. Section 3 provides the proposed methodology for the intrusion detection model describing the algorithm and implementation process on a network dataset. This methodology has been evaluated on various network traffic datasets, and the results obtained have been discussed and compared with other standard machine learning models in Sect. 4. Finally, the conclusion for the paper is given in Sect. 5.

2 Literature Review

An Industrial Internet of Things network or a computer network, in general, is a set of devices (nodes) in a plane that are connected with each other sharing resources over some communication protocol; however, the topology in which the devices are connected can affect its throughput and reliability. The information on how the devices are connected and how they are dependent in the network helps in identifying which device or traffic is affecting others in the network (Fig. 1).

Attackers have been evolving their methods for intruding the system, especially in cases of the Industrial IoT, that can become victims of espionage in the present day, where organizations are in ruthless competition for their share in the market. Attacks as distributed port scans, DNS amplification, botnet attacks and multi-flow attacks are more sophisticated in nature that methods as simpler machine learning models or deep learning models will fail to recognize [1]. The knowledge of the network flow is at a global level so as to have better understanding on how the relationship among the flow can better define an attack pattern, as many attacks are inherently characterized by the flow in the network. We have studied and given in Table 1 this context of IDS to improve on the performance; different papers were discussed where novel approaches for incorporating spatial features into intrusion detection systems were used.

Fig. 1 Graph representation of a network

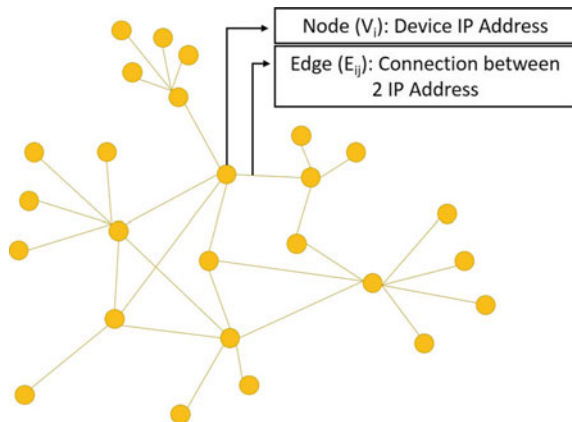


Table 1 Related works in spatial intrusion detection systems

Paper	Dataset used	Algorithm used	Results
Halbouni et al. [9]	CIC-IDS2017	Convolution neural network	Achieved 99.55% detection rate on multiclass attack classification
Iacovazzi et al. [2]	Mix-2022, Cui-2020	Graph representation used in RF	Achieved 0.898, 0.846 macro F_1 on multiclass attack classification
Zhu et al. [6]	TON_loT, BoT-loT, and UNSW-NB 15	Line-GraphSAGE	Achieved 98.8, 99.9, 99.6 accuracy on respective datasets
Islam et al. [4]	Real rawCAN dataset, OpelAstra data set	Graph-based Gaussian Naive Bayes	Achieved 98.1 and 99.57% on multiclass attack classification
Lo et al. [7]	TON_loT, BoT-loT	Graph neural network	Achieved 1.0 and 0.87 F_1 -score in detecting different attack types
Chang et al. [10]	UNSW-NB15, CIC-DarkNet, CSE-CIC-IDS, ToN-IoT	Graph-based neural and attention network	Improved 2, 3% F_1 -score for binary and multi-attack over base model
Otoum et al. [8]	NSL-KDD	DBSCAN	Achieved 95.6% accuracy on multiclass attack classification
Islam et al. [5]	Real CAN dataset	Graph-based model	Achieved 97.53% accuracy on multiclass attack classification

One of the most popular models for extracting spatial features is convolution neural network (CNN) that makes use of neighbouring information to give a spatial view of the network; this was used by Iacovazzi and Raza [2] to utilize the topological information of the network in order to classify network traffic as benign or anomalous on CIC-IDS 2017 dataset. They were able to obtain high detection rate for various attack types in the dataset. However, CNN has drawbacks when it comes to represent a network topology as convolution kernels are of fixed size in contrast to computer networks, which were solved with the usage of graphs as the input data. Various researchers have used graphs in different fields [3] to detect anomalies in a system. In view of intrusion detection of a computer network, graph representation can be used with machine learning models as Gaussian Naive Bayes, random forest in [2, 4, 5] or deep learning models as in [6–8]. The graph being only a way of representing finds its applicability with various models and has proved to improve on the performance of the system.

Graph is, on the other hand, much computationally more expensive to process, and hence, different researchers have been working to make it efficient for a wider range of applications. The study published by Lo et al. [7] proposed sampling-based algorithm that classifies graph edges; this model was an extension to graph sampling and aggregation network proposed by Hamilton et al. [11] to support edge classification. Chang et al. [10] further enhanced this to add residual learning to this model aiming to improve on the minority attack classes performance by dealing with the high-class imbalance in datasets. They also proposed another algorithm attention-based method, which resulted in better overall performance of the intrusion detection model as well as that of minority classes.

These studies demonstrate the potential of incorporating spatial features in intrusion detection systems to improve their accuracy. However, they also highlight some of the limitations of such approaches, such as the need for accurate information about the physical layout of the system, the requirement for a large amount of training data and the need for significant computational resources. Nevertheless, the use of spatial features in intrusion detection systems holds great promise for improving the accuracy and effectiveness of these systems. The proposed approaches in the discussed papers offer new insights and potential solutions to the challenges of incorporating spatial features into intrusion detection systems.

3 Proposed Methodology

Our proposed model uses graph convolution to generate better informed representation of data by considering the information of not only the node itself but its neighbours as well. This information can help in finding hidden patterns in the spatial domain of the network that may not be inferred otherwise using other deep learning models.

The flow from the network traffic data to the classification of traffic records into benign and attack data is divided into four steps as follows: data preprocessing, creation of graph object, E-graph convolution model and classification of edges for final output. These steps have been explained in the following subsections (Fig. 2):

3.1 Data Preprocessing

The traffic in a computer network is stored in the form of data flow, where a given flow identifies the source and destination of the data, and other fields that explain the data flow as flow duration, data size, protocols, etc. However, the recording of a network flow is not always perfect, and some faults may occur during the process. If this unchecked data is forwarded to a learning model, the model can make incorrect inferences from such unintended variations in the traffic data and provide unreliable output, so as to avoid such cases the dataset is cleaned for any faulty data as

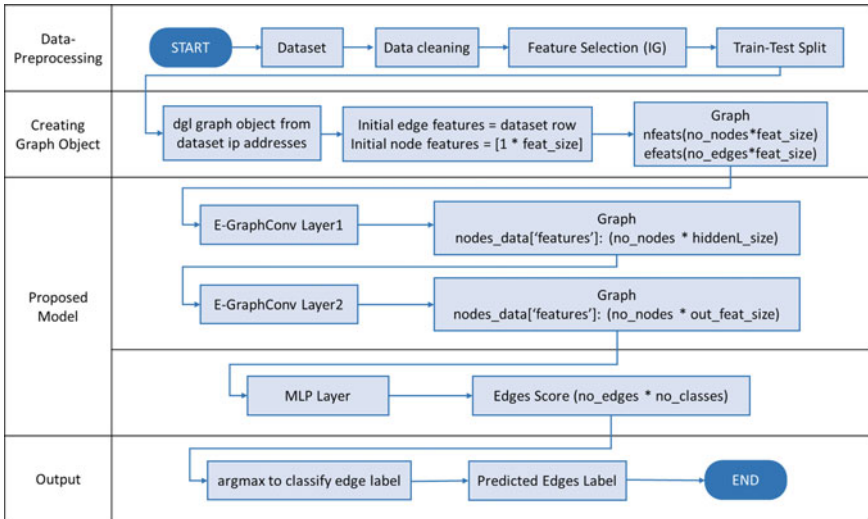


Fig. 2 Proposed E-Graph convolution model

missing, corrupted, or duplicate records. Further feature selection was performed on the dataset as for a given network traffic flow a record can have as much as hundreds of features; however, there are only much that are relevant considering our problem statement. Selective features were used to train our model using information gain in CIC-IDS-17 dataset only that had large number of features. The datasets have also been normalized as the convolution operation may lead to overflows.

3.2 Creating Network Graph Object

The network traffic as records of data flow is more widely used technique to capture the traffic where each record in the dataset is the set of attributes between the receiver and the sender IP addresses. Transforming the initial dataset into a graph object represents the data closer to real-world network traffic. The dataset is first transformed into a graph object $G(V, E, X_V, X_E)$, where V is the set of nodes (hosts), E is set of edges, X_V is set of features of node V , and X_E is set of network features of edge E connecting the node V_i and V_j . To construct the graph, the distinct set of private IP addresses in the dataset along with single merged IP address of external IP determines the set of nodes in the graph having initial features as ones of size that of number of attributes in the dataset. The edges between the nodes have features set of attributes of the record connecting the hosts. This translates our problem of intrusion detection in a network to that of edge classification, where we have to determine whether an edge in the network graph is anomalous or not, i.e. a binary classification of the edges.

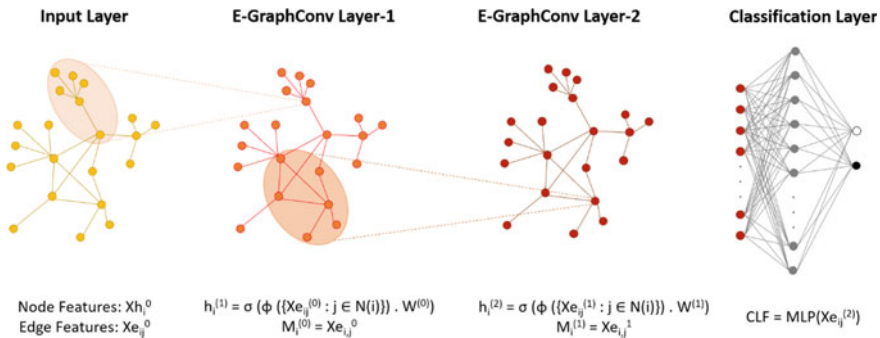


Fig. 3 Proposed E-GraphConv model

3.3 E-Graph Convolution Model

The proposed model of E-graph convolution model is a modification of graph convolutional network where in a graph convolutional Network, the term convolution is used to describe the operation that propagates information across the nodes of a graph. The concept of convolution in GCNs is inspired by traditional convolutional neural networks (CNNs) used for image processing, but adapted to work with graph-structured data.

In image-based CNNs, convolution involves applying a filter or a kernel to a local receptive field of pixels in the input image. This filter performs a dot product with the pixel values in the receptive field, producing a new feature representation that captures local patterns and spatial relationships. In GCNs, convolution is adapted to work with graph structures rather than regular grid-like image data. Instead of convolving with a fixed filter over local patches, GCNs perform convolution by aggregating and transforming information from neighbouring nodes in the graph.

The proposed model of E-graph convolution model is a simple neural network which fed the transformed graph dataset, where we have incorporated edge features to be utilized in the model in contrast to graph neural networks that inherently works with nodes. This allows the model to learn from its edge features as the model trains, on which the final classification is performed through a multi-layer perceptron layer. The proposed model can be subdivided into two parts as E-graph convolution layer and classification layer as shown in Fig. 3.

3.3.1 Edge-Graph Convolution Layer

E-graph convolution layer is the fundamental block of our model that makes use of information propagation to capture and encode the relational dependencies among edges in the graph. It operates on a graph structure and performs message passing and linear operation to update edge representations based on the information from neighbours.

Our model uses the transformed graph G with multidimensional edge features, in order to learn from these features, the E-graph convolution layer uses update function and message passing function.

Update Function: The update function computes the updated edge features based on the messages from neighbours and applies a nonlinear activation function. The update function can be represented as

$$h_i^{(l+1)} = \sigma(\phi(\{Xe_{ij}^{(l)} : j \in N(i)\}).W^{(l)}) \quad (1)$$

where

$h_i^{(l+1)}$	is the updated feature vector for node i at layer $l + 1$.
ϕ	is the linear transformation function the model uses to represent the information of the neighbouring nodes; for our proposed model, we have used the mean of information from neighbouring nodes to create a summarized representation for each node.
$Xe_{i,j}^{(l)} : j \in N(i)$	represents the set of edge features from neighbouring edges.
$W^{(l)}$	represents the weight matrix that maps the aggregated information to a new feature space. The aggregated information is transformed using learnable parameters to generate new node features.
σ	is the nonlinear activation function applied element-wise to introduce nonlinearities into the transformed node features. In our model, we have used ReLU to introduce nonlinearity to the layer.

Message Passing Function: Each edge in the graph uses information from its neighbouring nodes to update the edge representations which are passed using the message passing function. The message passing function can be represented as

$$M_i^l = Xe_{i,j} \quad (2)$$

where

M_i^l	is the message sent from edge between nodes i and j at layer l to its neighbouring edges.
$Xe_{i,j}$	is the edge feature between nodes i and j at layer l .

Through stacking multiple layers, the model can capture information from multiple hops in the graph, allowing for the modelling of complex relationships and dependencies. Each layer propagates and aggregates information from neighbouring nodes, refining the node representations with each layer. By repeatedly applying this convolution operation across multiple layers, the model can capture increasingly complex patterns and dependencies in the graph structure, allowing for tasks such as edge classification. However, stacking up too many layers can result in traversing the whole graph making the information saturated and results unreliable.

Our proposed model uses two E-graph convolution layers, which allows the model to learn through the features of two hop neighbours of the network graph.

3.4 *Multi-layer Perceptron Layer*

Multi-layer perceptron, or MLP layer, is stacked in addition to our E-graph conv layer as the resulting edge embeddings from the above layer are of size of the number of attributes in the network traffic flow; in order to classify them, scores are to be given to these edges. The MLP takes these inputs to update all the edges in the graph to a vector of size two, which implies the score towards the edge being benign or anomalous; i.e. the MLP layer classifies the edge embeddings into two classes, which can be used to compare with the labels provided in the network traffic dataset and tune the model during back propagation of the model.

4 Experimental Evaluation

The experimental evaluation of the proposed methodology for edge-based graph convolution network is performed on a Industrial IoT dataset, to substantiate our assumptions of improved detection rates while considering spatial knowledge of the IoT network. The experiments conducted to evaluate the performance of our model have been carried out in Google Colab tool using Python as the coding language.

4.1 *Dataset*

For our experiments on the methodology proposed, we require network traffic data among many datasets; following parameters were used for selection of dataset:

- The network traffic is from a network of Industrial Internet of Things or Internet of Things devices, preferably having a heterogenous set of devices in the network.
- The dataset offers a varied and up-to-date range of attacks to better evaluate the model.
- The data streams collected are to have IP addresses for the formation of graph object.

Based on the characteristics desired for evaluating our model, the Edge-IIoT (2022) dataset has been selected, which have the following characteristics:

- Size: The dataset contains 1,909,671 samples.
- Labels: The dataset is labelled with 14 different types of attacks, backdoor, DDoS (HTTP, ICMP, TCP, UDP), fingerprinting, MITM, password, port scanning, ransomware, SQL, uploading, vulnerability scanner and XSS.
- Sources: The dataset contains data from a variety of sources as flame sensor, heart rate sensor, IR sensor, Modbus sensor, soil moisture sensor, sound sensor, temperature humidity sensor, water level sensor and pH sensor.
- Features: The dataset contains 63 different feature attributes defining the network traffic records.

Table 2 Experimental evaluation results

Algorithm	Accuracy	Precision	Detection rate	F_1 -score
Proposed	0.997	0.998	0.997	0.996
CNN	0.98	0.996	0.93	0.968
MLP	0.95	0.96	0.91	0.93
RF	0.95	0.94	0.93	0.94
DT	0.95	0.97	0.91	0.94

Bold means proposed algorithm result

4.2 Performance Measures

The performance of the proposed model is determined using the metrics used for evaluating the classification-based machine learning model. This is in relation to the model proposed in this study which represents the output in the form of a Boolean. The performance metrics hence is described using the confusion matrix for each dataset, using accuracy, defined as the ratio of number of correctly classified traffic to the total number of network traffic records; precision, as the correctly identified attacks to number of identified attacks in the traffic; detection rate as the number of correctly identified attacks to total number of attacks in the traffic; and F_1 -score as the harmonic mean of precision and detection rate.

4.3 Experimental Results

In this section, we compare the performance of different models against our proposed model against various performance metrics mentioned in Sect. 4.2. The dataset selected as in accordance with the requirements described in the previous section was served as input to the model, where each dataset was preprocessed to remove redundancies and noise and scaled using minmax to normalize the data values between zero and one. The data was normalized as the proposed model is using message passing function that aggregates the neighbouring edge feature values in intermediate steps, having high values could lead to overflow in certain scenarios.

The obtained processed data is given to the model in the form of a graph to train the model, as well as test the effectiveness of the model using the performance measures described in the above subsection. It is also to be noted that datasets used in this paper include label in terms of the data being benign or anomalous in nature, which reduces the output as a binary classification problem and hence the model proposed is evaluated and compared in terms of performance metrics relevant to classification (Table 2).

The performance measures obtained for the dataset and its confusion matrix for the testing portion is shown in Fig. 4. To compare our achieved results, the same dataset has been trained and tested for some of the well-known classification models as

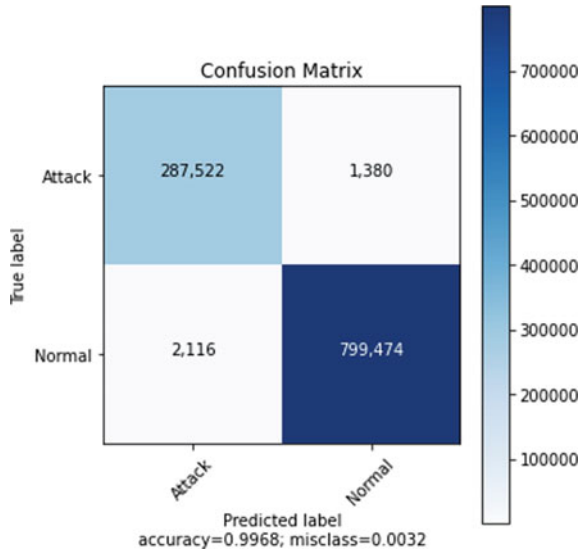


Fig. 4 Confusion matrix for proposed model on Edge-IIoT dataset

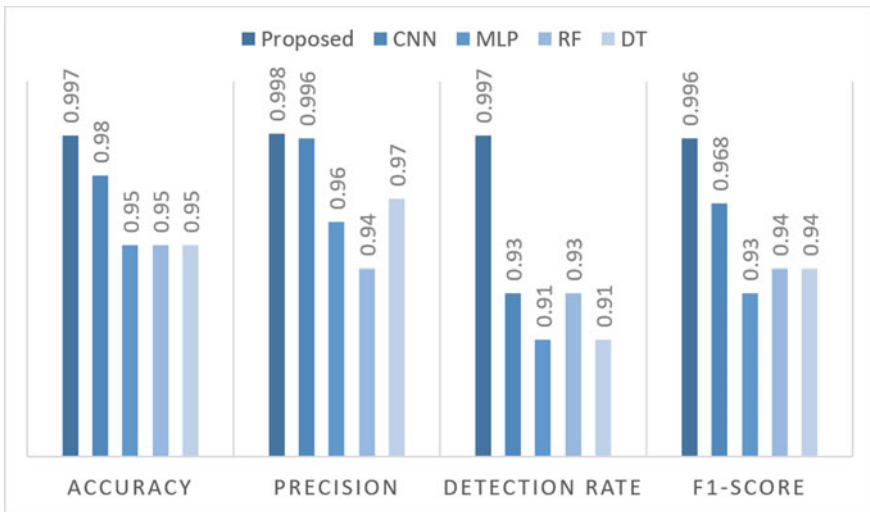


Fig. 5 Experimental results for edge-IIoT dataset

Naïve Bayes, decision tree, random forest and multi-layer perceptron. The evaluation results have also been graphically represented in Fig.5, substantiating our model performance to be better than that of the popular classification algorithms.

4.3.1 Key Findings

The key findings inferred from the results are as follows:

- E-graph convolution has demonstrated best performance in comparison with the highest F_1 -score of 0.996, leveraging its ability to effectively capture and model graph-structured data. Its success can be attributed to its unique architecture, which combines graph convolutional layers with nonlinear activation functions to extract meaningful representations from graph data. The incorporation of edge features aggregation and feature propagation further enhances its expressive power.
- CNNs are generally better at tasks that require local processing, such as image classification, whereas graph-based models as ours are generally better at tasks that require global processing. Its ability to capture both local and global graph structures enabling it to learn hierarchical representations of the underlying graph is verified from 2.89% improvement of F_1 score to 7.2% detection rate. This hierarchical modelling facilitates the understanding of complex relationships and dependencies among graph elements, leading to enhanced predictive accuracy.
- Notably, E-GCN's performance surpassed traditional machine learning techniques in terms of accuracy, predictive power and generalization ability, with 5.9% increase in tree-based model, and 7.1% in neural network model. This suggests popular models as random forest, decision tree and MLP which usually works on linear data structures struggles to exploit this structural information.

4.3.2 Limitations

The limitations discovered during our study for the model are as follows:

- Our proposed model is sensitive to the size of graph, as the methodology can be computationally expensive, especially for large graphs.
- Graph-based models as our E-graph conv are sensitive to the choice of hyperparameters making it difficult to find a set of hyperparameters that work well for a particular dataset. For instance in very large or dynamic graphs, two hop neighbours might not be sufficient to capture enough details from far neighbours.
- The performance relies on the underlying graph structure that means even small changes in the graph topology, such as node reordering or edge modifications, can impact the model's predictions.
- Our model is not suited for dynamic graphs as it assumes a fixed graph structure. Adding or removing nodes or edges in a graph may require retraining the model from scratch.

5 Conclusion and Future Scope

In this study, we have presented a graph-based network intrusion detection model. The network traffic data was transformed to network graph of hosts and connections representing the features that were used to classify between an anomalous and benign edge. The focus on this paper was the intrusion detection on Industrial Internet of Things owing to which the methodology was tested against datasets that can relate to IIoT applications.

The essence of the method proposed is the utilization of spatial features extracted from the graph that can help unveil attacks patterns in regard to features of a network record as well as its relation to its surrounding nodes. This approach achieved better performance results when compared with other standard classification models, which was verified through extensive evaluation through multiple datasets in the results section. The convolution method for recognizing the patterns in combination with deep learning allowed us to train a model that can identify network anomalies with the consideration of the global structural view of the network through convolution.

In future, we plan to work on optimizing the graph construction algorithm from network datasets that can help in reducing the time required to transform network traffic. Another aspect in terms of detecting an attack is temporal nature of the network, as the networks are ever evolving, we plan to work on an algorithm that takes into consideration the spatial as well as temporal varying information in a network.

References

1. Pujol-Perich D, Suarez-Varela J, Cabellos-Aparicio A, Barlet-Ros P (2022) Unveiling the potential of graph neural networks for robust intrusion detection. *ACM SIGMETRICS Perform Eval Rev* 49(4):111–117
2. Iacovazzi A, Raza S (2022) Ensemble of random and isolation forests for graph-based intrusion detection in containers. In: 2022 IEEE international conference on cyber security and resilience (CSR). IEEE, pp 30–37
3. Ma X, Wu J, Xue S, Yang J, Zhou C, Sheng QZ, Xiong H, Akoglu L (2021) A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans Knowl Data Eng*
4. Islam R, Devnath MK, Samad MD, Al Kadry SMdJ (2022) GGNB: graph-based gaussian Naive Bayes intrusion detection system for can bus. *Veh Commun* 33:100442
5. Islam R, Refat RUD, Yerram SM, Malik H (2020) Graph-based intrusion detection system for controller area networks. *IEEE Trans Intell Transp Syst* 23(3):1727–1736
6. Zhu H, Lu J (2022) Graph-based intrusion detection system using general behavior learning. In: *GLOBECOM 2022—2022 IEEE global communications conference*. IEEE, pp 2621–2626
7. Lo WW, Layeghy S, Sarhan M, Gallagher M, Portmann M (2022) E-graphsage: a graph neural network based intrusion detection system for IoT. In: *NOMS 2022—2022 IEEE/IFIP network operations and management symposium*. IEEE, pp 1–9
8. Otoum S, Kantarci B, Mouftah HT (2020) A novel ensemble method for advanced intrusion detection in wireless sensor networks. In: *ICC 2020—2020 IEEE international conference on communications (ICC)*. IEEE, pp 1–6

9. Halbouni AH, Gunawan TS, Halbouni M, Assaig FAA, Effendi MR, Ismail N (2022) CNN-IDS: convolutional neural network for network intrusion detection system. In: 2022 8th International conference on wireless and telematics (ICWT). IEEE, pp 1–4
10. Chang L, Branco P (2021) Graph-based solutions with residuals for intrusion detection: the modified E-GraphSAGE and E-ResGAT algorithms. arXiv preprint [arXiv:2111.13597](https://arxiv.org/abs/2111.13597)
11. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems, vol 30

Retinal Blood Vessel Segmentation Using an EDADCN Architecture—Encoder–Decoder Architecture with Dilated Convolutions and Attention Mechanism



M. J. Carmel Mary Belinda, S. Alex David, E. Kannan, and N. Ruth Naveena

Abstract The diagnosis and treatment of different retinal diseases depend heavily on the ability to segment retinal blood vessels. Deep learning approaches take extensively used in recent years to segment retinal blood vessels. The encoder–decoder architecture, attention mechanism, dilated convolutions, and capsule networks are the four main elements that make up the EDADCN architecture, which is used to divide retinal blood vessels. The segmentation map is created by using the encoder–decoder architecture to extract high-level features from the input picture. Dilated convolutions capture multi-scale information for segmenting structures of various sizes and shapes, while the attention mechanism allows selective focus on important areas of the picture. Capsule networks are used to manage the various blood vessel sizes and shapes. The evaluation findings show that the proposed architecture outperforms cutting edge techniques for blood vessel segmentation. The suggested design achieves high segmentation accuracy, sensitivity, and specificity, which are crucial for accurate detection and treatment of retinal diseases. Additional strategies like transfer learning and ensemble methods could be incorporated to improve the efficiency of the architecture even more. The blood vessel segmentation automated from retinal images can be made possible by the suggested deep learning architecture, allowing for the speedy detection and treatment of retinal diseases.

M. J. C. M. Belinda

Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India

S. A. David (✉) · E. Kannan

Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunathala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India
e-mail: adstechlearning@gmail.com

N. R. Naveena

Department of Mathematics, Hindustan Institute of Technology & Science, Chennai, Tamil Nadu, India

Keywords Retinal blood vessel segmentation · Deep learning · Encoder–decoder architecture · Dilated convolutions · Capsule networks

1 Introduction

In medical image analysis, retinal blood vessel separation is a fundamental assignment that involves identifying blood vessels in retinal fundus images. Traditional methods for segmentation of blood vessels relied on handcrafted features and algorithms like Support Vector Machines (SVMs), Random Forests, and Decision Trees [1]. However, these approaches have limited accuracy and require extensive feature engineering, making them challenging to apply to large datasets and time consuming. With recent advancements in deep learning, more accurate and robust segmentation methods have emerged that can automatically learn relevant features from the data. The use of CNNs has significantly improved retinal blood vessel segmentation accuracy, while reducing the need for extensive feature engineering. This paper provides an overview of retinal anatomy and explores the clinical significance of blood vessel segmentation. It then explores the challenges associated with this task and how deep learning has addressed these issues.

1.1 Retinal Anatomy and Clinical Significance

The retina, a delicate tissue-lining layer in the back eye, is in charge of detecting light and transmitting messages to the brain. The retinal blood vessels consist of arteries and veins that originate from the optic disc and branch out toward cover the entire retina [2]. Consider diabetic retinopathy as an example, where damage to the retinal blood vessels can trigger the formation of abnormal vessels, leading to leakage and ultimately causing vision loss. Hypertensive retinopathy causes a narrowing of the blood vessels in the retina, resulting in decreased blood flow and a decline in vision. Another prevalent ocular condition that can cause vision loss due to damage to the macula, the area of the retina responsible for sharp vision, is age-related macular degeneration.

1.2 Challenges in Retinal Blood Vessel Segmentation

Interesting task is the low distinction of the vessels, the presence of noise and artifacts, and the varying sizes and shapes of the vessels. Additionally, the high variability of retinal images across different patients, imaging devices, and imaging conditions poses a significant challenge to the development of robust segmentation algorithms. Traditional techniques for segmenting retinal blood vessels depend on

manually created features that are intended to capture the distinctive qualities of the vessels, such as their size, shape, and orientation. However, these techniques are time consuming and difficult to use with big datasets because of their low accuracy and extensive feature engineering requirements.

1.3 Deep Learning-Based Retinal Blood Vessel Segmentation

Recent developments in deep learning have resulted in the creation of more precise and robust segmentation methods that can automatically learn relevant features from the data CNNs, which have demonstrated to be incredibly effective in processing images and identifying features pertinent to the segmentation task, which are frequently used for retinal blood vessel segmentation. A three-layer neural network was used in retinal images to segment the vessels, in one of the early approaches for segmenting retinal blood vessels [3]. Since then, a wide range of deep learning-based techniques have been put forth, yielding appreciable gains in precision and effectiveness. Utilizing CNNs is the most typical method for segmenting retinal blood vessels using deep learning. CNNs are made up of multiple fully connected layers for classification or segmentation and multiple convolutional layers for feature removal from the image. The U-Net architecture, which was suggested by [4], is one of the most effective CNN-based methods for segmenting retinal blood vessels. A fully convolutional network with an encoder–decoder design is called U-Net. It has been demonstrated that U-Net performs at the cutting edge on a number of retinal blood vessel segmentation standards. The DeepVesselNet design, suggested by [5], is another well-liked CNN-based method for segmentation of retinal blood vessels [6]. The architecture was designed toward handling of low contrast and noise present in retinal image.

1.4 Advantages and Limitations

Several advantages over traditional methods are including improved accuracy and efficiency, reduced need for feature engineering, and the ability to handle the high variability of retinal images [7]. They do, however, have some shortcomings that must be fixed. Deep learning-based techniques also cost a lot to compute, needing strong GPUs and specialized hardware to execute in real time. The inability of deep learning-based techniques to be interpreted is another drawback. This can be a serious problem in medical applications where the patient may suffer significantly as a result of the algorithm's choices.

These deep learning-based methods typically use CNNs, which are highly effective in processing images and detecting relevant features [8]. Although they offer several advantages over traditional methods, deep learning-based methods also have limitations that need to be addressed, such as the requirement for large amounts of

labeled data, computational expenses, and lack of interpretability. Further research is needed to improve the segmentation algorithms' accuracy and efficiency. Additionally, the clinical usefulness of these methods needs to be evaluated in real-world settings to determine their impact on the diagnosis and treatment.

1.5 Future Directions

Future studies can focus on a number of areas which include:

- Data augmentation.
- Multi-modality integration.
- Transfer learning.
- Interpretability.
- Real-time segmentation.
- Clinical validation.

Traditional techniques for segmenting retinal blood vessels depend on manually created features and machine learning techniques that are inaccurate and demand a lot of feature engineering. The creation of more precise and reliable segmentation techniques that can automatically learn pertinent features from the data is the result of recent developments in deep learning.

CNNs have proven to be exceptionally effective in image processing and feature identification tasks relevant to separation. Though deep learning approaches have some benefits over more conventional approaches, there are some drawbacks that must be overcome. These restrictions include the need for a lot of annotated data for training, the cost of the models' computations, and their uninterpretability.

1.6 Motivation, Justification, and Contribution

The early detection and treatment of these diseases can be aided by accurate segmentation of blood vessels, which could eventually improve patient outcomes. However, manually segmenting blood vessels requires a lot of time and effort, so automated segmentation techniques are required. To achieve high accuracy in segmenting blood vessels, the architecture proposed has provided a novel method for segmenting retinal blood vessels that combines four essential elements. The attention mechanism aids in focusing on the most informative features, while the encoder–decoder design enables the extraction of high-level features and creation of a segmentation map. Dilated convolutions capture information on multiple scales, and capsule networks manage variation in blood vessel size and shape. These four elements working together could result in segmenting retinal blood vessels at the cutting edge of technology.

The proposed architecture offers several advantages over existing methods. The generation of a segmentation map and the effective extraction of high-level features

are made possible by the use of an encoder–decoder architecture. The incorporation of an attention mechanism helps to progress the accuracy of the segmentation by focusing on the most informative descriptions. Dilated convolutions enable the model to apprehension multi-scale information and advance the separation. The use of capsule networks is particularly novel and offers several advantages over traditional convolutional neural networks. By representing blood vessels as capsules, the model is better able to capture their orientation and deformation, which is particularly important for thin and elongated constructions such as blood vessels. In order to handle the variability in the structure and size of blood vessels, capsule networks must also be able to handle differences in entity orientation and deformation.

The main contributions are:

- By utilizing an encoder–decoder with skip connection architecture, it is possible to reserve low-level structures while extracting high-level structures, improving the accuracy of the segmentation map and capturing more details in the retinal picture.
- The encoder–decoder architecture’s attention mechanism in the skip connections helps with selective emphasis on key features.
- The use of dilated convolutions in both the encoder and decoder parts of the system enables the segmentation of structures with various sizes and shapes.

Further improving uses the capsule networks to represent the blood vessels as capsules, which helps to manage variations in their orientation and deformation.

2 Related Works

Automated retinal vessel separation is vital for detecting diabetic retinopathy and vascular occlusion, but manual segmentation is inefficient and lying to mistakes. Researchers have been developing new techniques using supervised and unsupervised methods, but they often have limitations such as thin or fake branches. In this artifact, a novel restricted generative adversarial network called M-GAN is proposed [9], which outperforms earlier work in accuracy and other metrics using four publicly available datasets. The authors suggest using M-GAN for other medical image segmentations and industrial fields. Accurate segmentation is essential for early disease diagnosis, but it can be challenging due to differences in vessel morphology and low-slung. To address this, a Multi-Scale Convolutional Neural Network with Attention Mechanisms (MSCNN-AM) is suggested [10].

The Dense [11] design uses dense blocks instead of skip connections to fuse structures from low to deep layers, and the Inception module derives vessel features with various convolution kernel sizes. GANs are also used in the training, with a GAN loss and segmentation. Evaluation on the DRIVE dataset shows promising results, with a Dice rate of 82.15% and AU-ROC and AU-PR of 0.9772 and 0.9058, respectively. The U-Net architecture performs well, but becomes complicated as more layers are added [12]. The study proposes VG-DropDNet, which achieved a high accuracy of 95.36%

on the DRIVE dataset and 98.56% on the STARE dataset. The study highlights the importance of choosing the right architectures for retinal vessel segmentation and suggests VG-DropDNet as a useful technique. Overall, VG-DropDNet yields promising outcomes for automated segmentation of retinal vessel. The work in [13] proposes an unsupervised method which combining Hessian-based and intensity transformation techniques. The suggested approach uses Contrast-Limited Adaptive Histogram Equalization (CLAHE) to enhance contrast, and an improved Particle Swarm Optimization (PSO) algorithm for contextual region tuning. Thick and thin vessel-enhanced images are extracted using the Hessian matrix and then subjected to global and local thresholding. Non-vessel components are eliminated using a post-processing phase based on region parameters. The method outperforms several unsupervised methods on CHASE_DB1 and DRIVE datasets, with accuracy values of 0.9505 and 0.9559, respectively.

The precision of retinal vessel segmentation is improved by a suggested method called consideration-guided U-Net with atrous convolution (AA-U-Net) [14]. The model creates an attention mask to distinguish between vessel and non-vessel pixels, which is used as a weighting function to multiply the differential feature map. To minimize computation and expand the receptive field, atrous convolution is used. Two shortcuts are added to the atrous convolution to emphasize the vessel's intricacies. The model is evaluated on DRIVE, STARE, and CHASE_DB1 datasets, achieving accuracies of 0.9558, 0.9640, and 0.9608, and AUCs of 0.9847, 0.9824, and 0.9865, respectively.

A new deep learning-based method is suggested in this study to progress the accuracy of retinal vessel segmentation while decreasing the computational difficulty and memory overhead. The proposed method in [15] uses skip connections and semantic pixel-wise segmentation to address the limitations of fully convolutional neural networks. The sensitivity, specificity and accuracy performance of the suggested method are 0.8252, 0.8440, and 0.8397; 0.9792, 0.9810, and 0.9887; 0.9649, 0.9722, and 0.9659; and 0.9780, 0.9830, and 0.9810, respectively. ResWnet, which replaces convolution layers with residual blocks and employs an encoding–decoding–encoding–decoding assembly with skip connections, is also recommended for segmenting small vessels [16].

3 Proposed Methodology

3.1 Dataset

This study uses the Digital Retinal Images for Vessel Extraction (DRIVE) [17] and Structured Analysis of the Retina (STARE) datasets [18].

Image augmentation is a technique that is often used in computer vision tasks to increase the variety of the training data. The methods employed for generating augmentation are illustrated in Fig. 1.

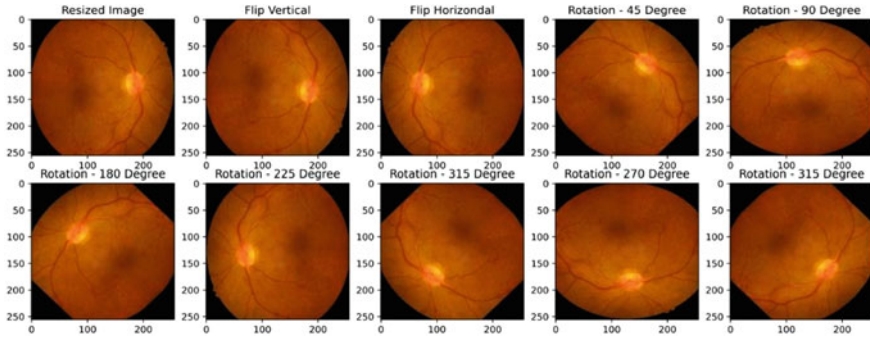


Fig. 1 Original image and augmented images of a fundus image

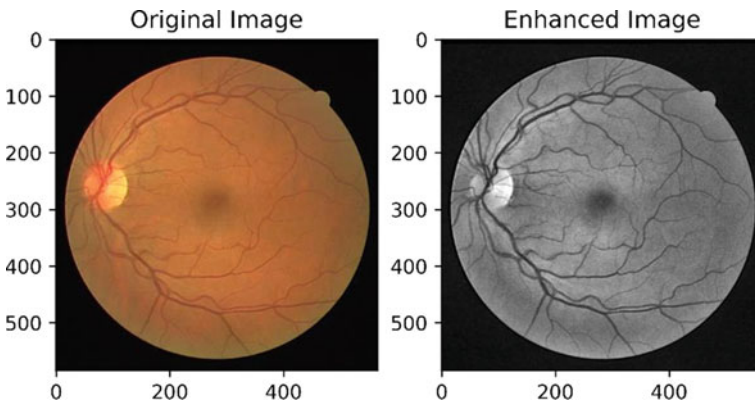


Fig. 2 CLAHE-enhanced fundus image

3.2 Image Enhancement

Contrast-Limited Adaptive Histogram Equalization (CLAHE) uses a clip limit that serves as a contrast limiter to enhance image clarity. The input picture was divided into a number of small regions known as tiles by CLAHE. The histogram’s local areas would be redistributed with the truncated pixels [19]. The enhanced image is shown in Fig. 2.

3.3 EDADCN Architecture

The proposed EDADCN architecture for retinal blood vessel has four components and each component has been described below.

- **Encoder–Decoder Architecture:** The encoder–decoder architecture is helped to extract high-level information from the data and produce the segmentation map. The encoder part of the architecture is responsible for down sampling the image and extracting the features, while the decoder part of the architecture is responsible for up sampling the features and generating the segmentation map. Skip connections are helped to preserve the low-level information and helped the model to better capture the details in the image.
- **Attention Mechanism:** The attention mechanism is used to selectively emphasis on important regions of the image. Attention gates are used in the skip connections of the encoder–decoder architecture to identify the most informative details and use them to improve the segmentation accuracy.
- **Dilated Convolutions:** Dilated convolutions are used in both encoder and decoder to capture multi-scale information for segmenting structures of varying sizes and shapes. The model is better able to segment thin and elongated structures like blood vessels by collecting multi-scale information.
- **Capsule Networks:** Capsule networks are used to represent objects as capsules and handle variations in object orientation and deformation. By representing the blood vessels as capsules, the model is better able to capture their orientation and deformation, which helps to advance the accuracy of the segmentation map.

The proposed architecture consists of four main component shown in Fig. 3. The encoder down samples and extracts the features from the image, while the decoder up samples the features and generates the segmentation map. By concentrating on the important areas of the image, the consideration mechanism helps the model to better differentiate between blood vessels and background regions, resulting in improved segmentation accuracy. Dilated convolutions capture multi-scale information for segmenting structures of varying sizes and shapes. Capsule networks represent objects as capsules and handle variations in object orientation and deformation.

The combination of these four components contributes to the improved accuracy of the retinal blood vessel segmentation model by extracting high-level features, selectively focusing on important regions, capturing multi-scale information, and handling variations in object orientation and deformation. Figure 4 shows the original image and segmented image

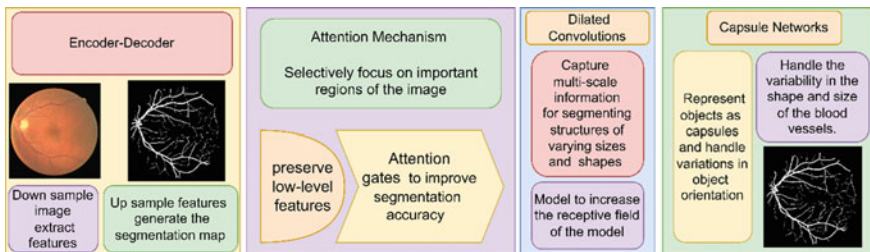


Fig. 3 EDADCN architecture for retinal blood vessel segmentation

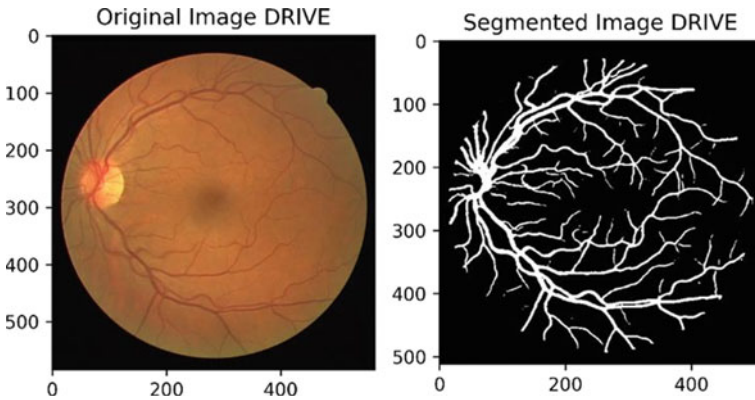


Fig. 4 Segmented blood vessel from the fundus image

3.4 Mathematical Models for EDADCN Architecture

Encoder–decoder Architecture:

Let a be the input image, and let b be the corresponding segmentation map. The encoder–decoder architecture can be represented mathematically as follows:

$$b = f(a; \theta),$$

where f is a neural network with learnable parameters θ . The encoder part of the network can be written as:

$$z = g(x; \theta e),$$

where z is the output of the encoder, and g is a function that down samples the input image and extracts high-level features. The decoder part of the network can be written as:

$$b = h(z; \theta d),$$

where h is a function that up samples the features and generates the segmentation map.

Skip connections are used to preserve low-level features in the network. The output of the encoder is concatenated with the output of the corresponding decoder layer to produce the skip connection.

Algorithm

Input:

- Image I
- Encoder network E with parameters θ_E
- Decoder network D with parameters θ_D

Output:

- Segmentation map S

Steps:

- Pass the image I through the encoder network E with parameters θ_E to get the encoded features $F_{enc} = E(I; \theta_E)$.
- Pass the encoded features F_{enc} through the decoder network D with parameters θ_D to get the segmentation map $S = D(F_{enc}; \theta_D)$.
- Return the segmentation map S.

Attention Mechanism:

The attention mechanism can be represented mathematically as follows:

$$z' = A(z; \theta_a),$$

where z is the input feature map, A is the attention function with learnable parameters θ_a , and z' is the output feature map. The attention function applies a learned attention mask to the input feature map to selectively focus on important regions of the image.

Algorithm

Input:

- Skip connection feature maps F_{skip}
- Skip connection feature maps F_{enc}
- Convolutional layer C
- Parameters θ_C

Output:

- Attended feature maps F_{att}

Steps:

- Concatenate the feature maps F_{skip} and F_{enc} along the channel axis to get F_{concat} .
- Apply the convolutional layer C with parameters θ_C to F_{concat} to get the attention map $A = C(F_{concat}; \theta_C)$.
- Compute the attention weights W by passing the attention map A through a softmax function along the channel axis.

- Multiply the feature maps F_{enc} by the attention weights W to get the attended feature maps $F_{att} = F_{enc} * W$.
- Return the attended feature maps F_{att} .

Dilated Convolutions:

Dilated convolutions can be represented mathematically as follows:

$$y = D(x; \theta d),$$

where x is the input feature map, y is the output feature map, D is the dilation function with learnable parameters θd , and the output y is computed as a function of the input x and the dilation kernel. Dilated convolutions are used to capture multi-scale information for segmenting structures of varying sizes and shapes.

Algorithm

Input:

- Feature maps F
- Dilation rate r
- Convolutional layer C
- Parameters θC

Output:

- Dilated feature maps F_{dil}

Steps:

- Apply the convolutional layer C with parameters θC to the feature maps F to get the intermediate feature maps $F_{int} = C(F; \theta C)$.
- Apply dilated convolution to the intermediate feature maps F_{int} with dilation rate r to get the dilated feature maps $F_{dil} = C(F_{int}; \theta C, r)$.
- Return the dilated feature maps F_{dil} .

Capsule Networks

Capsule networks can be represented mathematically as follows:

$$v = C(x; \theta c),$$

where x is the input feature map, v is the output capsule vector, C is the capsule function with learnable parameters θc , and the output capsule vector represents the properties of the object. In the context of retinal blood vessel segmentation, capsule networks are used to handle the variability in the shape and size of the blood vessels and to capture their orientation and deformation.

Algorithm

Input:

- Feature maps F
- Capsule layer C
- Parameters θ_C

Output:

- Capsule output O

Steps:

- Apply the capsule layer C with parameters θ_C to the feature maps F to get the capsule output $O = C(F; \theta_C)$.
- Return the capsule output O .

4 Performance Comparison

The segmented blood vessels from the original image are shown in Fig. 4. Sensitivity, specificity, accuracy, and area under the receiver operating characteristic curve are some of the evaluation criteria frequently used for jobs involving retinal blood vessel segmentation (AUC-ROC). Table 1 shows the performance analysis using the DRIVE dataset.

The proposed method performs the best in terms of Se, Sp, and AUC. It achieves a Se of 0.8521, Sp of 0.9785, and an AUC of 0.9758. The proposed method also achieves the highest accuracy of 0.9715 among all the methods listed in the table. However, the F1-score for the proposed method is relatively low compared to some

Table 1 Performance comparison based on DRIVE dataset

Method	Se	Sp	Acc	F1	AUC
Park et al.	0.8346	0.8302	0.9706	0.8324	0.9868
Qilong et al.	0.8342	0.9732	0.9555	0.8267	0.9795
Anita et al.	0.7974	0.9761	0.9536	0.8144	–
Musaed et al.	0.7851	0.9724	0.9559	–	0.8787
Tariq et al.	0.8252	0.9787	0.9646	–	0.9780
Tang et al.	0.8162	0.9756	0.9554	–	0.9799
Khan et al.	0.7852	0.9671	0.9522	–	0.9651
Noh et al.	0.835	0.975	0.957	0.831	0.978
Aurangzeb et al.	0.8491	0.9774	0.9659	–	0.9850
Proposed	0.8521	0.9785	0.9715	0.8265	0.9758

of the other methods. The high values achieved in the performance evaluation metrics has been highlighted in the Tables 1, 2 and 3.

Table 2 presents each method evaluated based on different metrics such as sensitivity (Se), specificity (Sp), accuracy (Acc), F1-score (F1), and area under the curve (AUC). Looking at the results, it appears that most of the methods have achieved high Se and Sp values, which indicates that they are able to detect both positive and negative cases with high accuracy. However, the Acc values vary among the methods, with some achieving higher values than others. The F1-scores also show some variations, where some methods have achieved high F1 values, while others have not reported them. The AUC values also vary among the methods, indicating the tradeoff between Se and Sp. In terms of individual methods, it seems that the proposed method has achieved the highest Se, Sp, and AUC values, as well as the highest accuracy, which indicates its overall effectiveness. However, its F1-score is relatively low compared to some other methods, which suggests that it may have some weaknesses in terms of precision and recall.

Table 3 lists the results of several methods on different datasets for a retinal blood vessel segmentation. The table includes several evaluation metrics, including sensitivity (Se), specificity (Sp), accuracy (Acc), F1-score (F1), and area under the curve (AUC). Looking at the results, we can see that some methods perform better than other on specific datasets. For example, Park et al. achieve high accuracy on both Chase-DB1 and HRF datasets, while Qilong et al. achieve high Se, Sp, Acc, and AUC on Chase-DB1. Meanwhile, K. J. Noh et al. achieve high Se, Sp, and AUC on the HRF dataset. The proposed method achieves high Se, Sp, and AUC on both the DRIVE and STARE datasets, indicating good performance on these datasets. However, the F1-score is relatively low, which means that there may be some imbalance between precision and recall. Overall, the table shows that different methods have different strengths and weaknesses and that their performance can

Table 2 Performance of the proposed method and other methods on STARE

Method	Se	Sp	Acc	F1	AUC
Park et al.	0.8324	0.9938	0.9876	0.8370	0.9873
Qilong et al.	0.8412	0.9807	0.9856	0.8401	0.9863
Anita et al.	0.9124	0.9941	0.9856	0.9299	–
Tariq et al.	0.8397	0.9792	0.9656	–	0.9810
Tang et al.	0.7551	0.9903	0.9723	–	0.9863
Khan et al.	0.7881	0.9662	0.9512	–	0.9471
Noh et al.	0.8541	0.9752	0.9542	0.9221	0.8431
Aurangzeb et al.	0.8573	0.9813	0.9719	–	0.9873
Proposed	0.8851	0.9885	0.9885	0.8265	0.9887

Table 3 Performance of the proposed method and other methods on various dataset

Method	Dataset	Se	Sp	Acc	F1	AUC
Park et al.	Chase-DB 1	–	–	0.9736	0.8110	0.9859
Park et al.	HRF	–	–	0.9761	0.7972	0.9852
Qilong et al.	Chase-DB 1	0.8132	0.9816	0.9644	0.8237	0.9839
Musaed et al.	Chase-DB 1	0.7776	0.9634	0.9505	0.8705	
Tariq et al.	Chase-DB 1	0.8440	0.9810	0.9722		0.9830
Khan et al.	Chase-DB 1	0.7875	0.9682	0.9521	–	–
Noh et al.	Chase-DB 1	0.8522	0.9871	0.9781	0.8402	0.9922
Noh et al.	HRF	0.8332	0.9792	0.9661	0.8192	0.9872
Aurangzeb et al.	Chase-DB 1	0.8607	0.9806	0.9731	–	0.9850
Proposed	DRIVE	0.8521	0.9785	0.9715	0.8265	0.9758
	STARE	0.8851	0.9885	0.9885	0.8265	0.9887

vary depending on the dataset used. It is important to consider multiple evaluation metrics and test on different datasets to get a comprehensive understanding of the performance of a method.

5 Conclusion

Various mathematical models have been proposed to achieve this task, which includes the encoder–decoder architecture, attention mechanism, dilated convolutions, and capsule networks. These models have been used to capture multi-scale information for segmenting structures of varying sizes and shapes, to handle the variability in the shape and size of the blood vessels, and to capture their orientation and deformation. The proposed method has provided an accurate and reliable method for retinal blood vessel segmentation by incorporating the four components of the models. The incorporation of these models has enabled the evaluation of these parameters more accurately and with improved performance. Overall, the proposed mathematical models provide a promising direction for further research in retinal blood vessel segmentation, and the use of these models is expected to improve the diagnosis of ocular diseases and provide better treatment outcomes for patients.

References

1. Ravikumar S, Kumar KA, Koteeswaran S (2018) Dismemberment of metaphors with grid scratch via Kernel K-means. *J Comput Theor Nanosci* 15(11–12):3533–3537
2. David SA, Mahesh C, Kumar VD, Polat K, Alhudhaif A, Nour M (2022) Retinal blood vessels and optic disc segmentation using U-net. *Math Probl Eng* 2022:1–11

3. Tchinda BS, Tchiotsop D, Noubom M, Louis-Dorr V, Wolf D (2021) Retinal blood vessels segmentation using classical edge detection filters and the neural network. *Inform Med Unlocked* 23:100521
4. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for bio-medical image segmentation. In: *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer International Publishing, pp 234–241
5. Tetteh G, Efremov V, Forkert ND, Schneider M, Kirschke J, Weber B, Zimmer C, Piraud M, Menze BH (2020) Deepvesselnet: vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *Front Neurosci* 14:1285
6. Duggani K, Nath MK (2021) A technical review report on deep learning approach for skin cancer detection and segmentation. In: Khanna A, Gupta D, Pólkowski Z, Bhattacharyya S, Castillo O (eds) *Data analytics and management. Lecture notes on data engineering and communications technologies*, vol 54. Springer, Singapore. https://doi.org/10.1007/978-981-15-8335-3_9
7. Yadav R, Pandey M (2022) Image segmentation techniques: a survey. In: Gupta D, Polkowski Z, Khanna A, Bhattacharyya S, Castillo O (eds) *Proceedings of data analytics and management. Lecture notes on data engineering and communications technologies*, vol 90. Springer, Singapore. https://doi.org/10.1007/978-981-16-6289-8_20
8. Jain A, Pandey M, Sahu S (2022) A deep learning-based feature extraction model for classification brain tumor. In: Gupta D, Polkowski Z, Khanna A, Bhattacharyya S, Castillo O (eds) *Proceedings of data analytics and management. Lecture notes on data engineering and communications technologies*, vol 90. Springer, Singapore. https://doi.org/10.1007/978-981-16-6289-8_42
9. Park K-B, Choi SH, Lee JY (2020) M-GAN: retinal blood vessel segmentation by balancing losses through stacked deep fully convolutional networks. *IEEE Access* 8:146308–146322. <https://doi.org/10.1109/ACCESS.2020.3015108>
10. Fu Q, Li S, Wang X (2020) MSCNN-AM: a multi-scale convolutional neural network with attention mechanisms for retinal vessel segmentation. *IEEE Access* 8:163926–163936. <https://doi.org/10.1109/ACCESS.2020.3022177>
11. Guo X et al (2020) Retinal vessel segmentation combined with generative adversarial networks and dense U-Net. *IEEE Access* 8:194551–194560. <https://doi.org/10.1109/ACCESS.2020.3033273>
12. Desiani A, Suprihatin EB, Efriliyanti F, Arhami M, Setyaningsih E (2022) VG-DropDNet a robust architecture for blood vessels segmentation on retinal image. *IEEE Access* 10:92067–92083. <https://doi.org/10.1109/ACCESS.2022.3202890>
13. Alhussein M, Aurangzeb K, Haider SI (2020) An unsupervised retinal vessel segmentation using Hessian and intensity based approach. *IEEE Access* 8:165056–165070. <https://doi.org/10.1109/ACCESS.2020.3022943>
14. Lv Y, Ma H, Li J, Liu S (2020) Attention guided U-net with atrous convolution for accurate retinal vessels segmentation. *IEEE Access* 8:32826–32839. <https://doi.org/10.1109/ACCESS.2020.2974027>
15. Khan TM, Alhussein M, Aurangzeb K, Arsalan M, Naqvi SS, Nawaz SJ (2020) Residual connection-based encoder decoder network (RCED-Net) for retinal vessel segmentation. *IEEE Access* 8:131257–131272. <https://doi.org/10.1109/ACCESS.2020.3008899>
16. Tang Y, Rui Z, Yan C, Li J, Hu J (2020) ResWnet for retinal small vessel segmentation. *IEEE Access* 8:198265–198274. <https://doi.org/10.1109/ACCESS.2020.3032453>
17. Khan MAU et al (2021) A scale normalized generalized LoG detector approach for retinal vessel segmentation. *IEEE Access* 9:44442–44452. <https://doi.org/10.1109/ACCESS.2021.3063292>
18. Noh KJ, Kim J, Park SJ, Lee S (2020) Multimodal registration of fundus images with fluorescein angiography for fine-scale vessel segmentation. *IEEE Access* 8:63757–63769. <https://doi.org/10.1109/ACCESS.2020.2984372>
19. Aurangzeb K, Alharthi RS, Haider SI, Alhussein M (2023) An efficient and light weight deep learning model for accurate retinal vessels segmentation. *IEEE Access* 11:23107–23118. <https://doi.org/10.1109/ACCESS.2022.3217782>

SDB-RGSO: Swarm-Based Data Balancing and Randomized Grid Search Optimization for IoT NetFlow Malware Detection with Ensemble Machine Learning Model



D. Santhadevi and B. Janet

Abstract Digital attacks on the Internet of Things (IoT) are increasing in frequency due to the massive growth in collaboration among various devices, apps, and connected networks. Long-lasting IoT network attacks have a detrimental impact on how user-accessible the necessary framework is, which might result in data breaches. One of the most formidable challenges for IoT systems and their impact on the environment is the robust detection of attacks and the prediction of malware within network flows. NetFlow-based malware detection around IoT systems is highly skewed which leads to falling predictions and alarms due to an imbalanced dataset. The research brought through this paper explores balancing the dataset using digital Ant-based synthetic minority over-sampling technique (DA-SMOTE) technique and tuning the hyper-parameters with randomized grid search to improve the performance metrics of the malware detection model. State-of-art provides that the ensemble machine learning algorithms (adaptive boosting, gradient boosting, histogram gradient boosting) improved the performance metrics and reduced the false warnings. It is observed that there is a significant improvement in f1-score and much reduction in the false alarms. This indicates that the framework proposed is highly effective in detecting malware attacks in the IoT system and related environments.

Keywords IoT network security · Malware detection system · SMOTE · Randomized grid search · Machine learning · Swarm intelligence

D. Santhadevi (✉)
SCOPE, VIT-AP University, Amaravathi, Andhra Pradesh, India
e-mail: santhadevi.nitt@gmail.com

B. Janet
National Institute of Technology, Tiruchirappalli, India
e-mail: janet@nitt.edu

1 Introduction

The recent exponential development in technology has resulted in innovative disruption across all the sections. IoT helps in enabling the myriad of heterogeneous business systems and their related applications [1] which can help various businesses to integrate the IoT technologies into the future investment and making the appropriate decision. Researchers explored that there is random violation in modern network systems, which are broad security components for real-time computer network monitoring. Effective planning, anticipation, and resolution for a cybersecurity incident are required to secure the current network system [2]. IoT offers a significant probable economic effect of 4–11 trillion USD a year by 2025 (according to Mckinsey). As per SonicWall, there is an increase of 5% IoT malware attacks in 2019, with a total estimate reaching 34.3 million attacks [3]. But as the new IoT devices connect each day, it is obvious that there is going to be an increase in IoT malware attacks. There shall be roughly 31 billion devices connected across the calendar year as per one of the sources. This pace of adoption and combined with lax manufacturing standards poses difficulty in managing the secure proof IoT environment [3].

Rawat et al. [4] analyzed the botnet behavior and network features to avoid the zero-day attacks. The statistical report from different security researchers concludes that there is a need for an intelligent malware detection system [5]. The system will monitor the network traffic; from that, it must give early warnings whenever abnormal behavior occurs. Authors focused their research in the area of cloud security to address malware attack detection using an ensemble learning approach [6]. The researchers found the importance of an intelligent monitoring system for malware detection in the modern era for securing real and large-scale networks [7]. As a result of a lack of sample, machine learning models are unable to recognize a particular attack. Class balancing is necessary to avoid circumstances in which the detection system must focus on different types of attacks on the network flow [8]. Several kinds of research are ongoing in this aspect; still, there needs more focus on the early detection of IoT malware attacks before it affects the entire network.

The proposed research framework focuses explicitly on detecting malware activity flow in IoT edge environments. Ensemble machine learning (ML) algorithm with DA-SMOTE and randomized grid search optimization are applied in this research.

Further, this paper is structured as follows. Section 2 describes the detailed survey of recent work navigated in this area of research. Section 3 presents the dataset and algorithms which are used in the research. Section 4 has a detailed explanation of the proposed model and techniques. In Sect. 5, results are analyzed with performance metrics and compared with the existing system. The research work is concluded in Sect. 6 with proof of results obtained from the experiment.

2 Literature Survey

The current innovation pace has generated a significant surge in computational capabilities, which has amounted to the investigation of machine learning techniques and algorithms as one the effective network security measures [9]. Moustafa and Slay did a statistical analysis on two datasets (UNSW-NB15 & KDD99) for selecting the best one for IoT network intrusion detection [10]. For assessing network intrusion detection, this benchmark dataset is used in this paper to evaluate the proposed model.

Table 1 represents the summary of sound literature review papers used to build an efficient malware detection model in the IoT edge environment. The Table 1 analysis report shows that there is a requirement to improvise the performance of malware detection through balancing the minority samples and optimizing machine learning models. This research aims to build a model with a balanced dataset using DA-SMOTE technique, its features are optimized through hyperparameters tuning with randomized grid search, until the anticipated objective function reaches negligible false warnings and whereby accuracy reaches a maximum value.

Table 1 Review of existing malware attack detection system

Ref. no	Methodology	Dataset	Metrics used	Advantages	Limitations
[8]	RF + SMOTE, J48, bagging, AdaBoost	KDD Cup 99	Accuracy, precision, AUC	Improves the detection rate on minority samples	Takes more time for training
[11]	KNN, Naive Bayes, MLP	BoT-IoT	Accuracy, ROC-AUC	DDoS attack detection at IoT network	False detection due to class imbalance
[12]	CIL + RF, SVM, NB, KNN	UCL	Precision, recall, F score, ROC	In android malware app detection, SMOTE achieved a better result than the random under-sampling	High processing time
[13]	Relief + borderline sampling, KNN, C4.5, NB	NSL-KDD	Accuracy, precision, recall, FPR, specificity, G-mean, F -score	Improved accuracy in classification of minority sample classes	High false-positive rate
[14]	BO + KNN, BO + RF, BO + SVM	ISCX 2012	Accuracy, precision, recall, FAR	Increased malware detection rate	More processing time
[15]	M-AdaBoost + PSO	KDD, NSL-KDD, KWID	Accuracy, precision, recall, G-mean	Improved malware detection accuracy	More processing time

3 Material and Method

The dataset UNSW-NB15 [16, 17] was developed by the Australian Cyber Security Centre (ACCS) cyber range lab using the IXIA PerfectStrom tool. This was done to produce a mixture of very real modern regular operations and synthetic attack behaviors. TCP dump tool was used to detect raw traffic of 100 GB data (e.g., Pcap files). Around nine types of attacks were included in this dataset, namely worms, fuzzers, DoS, generic, backdoors, reconnaissance, analysis, exploits, and shellcode. The Argus and Bro-IDS programs were used to build the class label to produce a total of 49 features. The ground truth training and testing datasets are concatenated, which are around 175,341 and 82,332 records, respectively, on different forms as attack and normal which is represented in Table 2. The dataset is in the form of binary classification [10].

3.1 Methodology—Ensemble Machine Learning Algorithms

The primary focus of machine learning is to constitute applications that learn from the data and help in improving the decision-making process which improves the predictive accuracy over a period of time [18, 19]. ML algorithm is said to be better when it provides high accuracy, decision, and prediction. Machine learning broadly falls into three primary types, i.e., supervised learning, semi-supervised and unsupervised learning, and reinforcement learning [20]. In this research, supervised ensemble learning algorithms are used to predict the malware flow in the IoT network. Ensemble algorithms are the generalized integrative technique to machine learning

Table 2 UNSW-NB15 dataset description

Categories	Training set	Testing set	Total
Normal	56,000	37,000	970,000
Analysis	2000	677	2677
Backdoor	1746	583	2329
DoS	12,264	4089	16,353
Exploits	33,393	11,132	44,525
Fuzzers	18,184	6062	24,246
Generic	40,000	18,871	40,000
Reconnaissance	10,491	3496	13,987
Shellcode	1133	378	1511
Worms	130	44	174
Total	175,341	82,332	257,673

that combines the predictions of numerous models to improve predictive performance. Boosting, bagging, and stacking are the different ensemble techniques. In this study, boosting ensemble algorithms are used, which are discussed as follows.

Adaptive Boosting

Adaptive boost algorithm [15] is used to enhance the performance of the base learners in a binary classification problem [21]. The decision tree has been used for boosting purposes because of its short and single decision classification, which is called stumps.

Algorithm 1: Optimized adaptive boosting

$D_k(i)$: Example i weight after learner k

α_k : Learner k weight

$\forall_i : D_0(i) \leftarrow \frac{1}{N}$ // Set uniform example weights

For $k = 1$ **to** k **do** // for Each base learner, do train the base learner with the weighted sample

$D \leftarrow$ data sampled with D_{k-1}

$h_k \leftarrow$ base learner trained on D

$\epsilon_k \leftarrow \sum_{i=1}^N D_{k-1}(i) \delta y_i$

End For

Were,

k base learners, N examples, α_k is the weight of the k^{th} learner, $h_k(x_i)$ is its prediction on

x_i , ϵ_k is the weighted error of the k^{th} learner

Z_k depends only on ϵ_k , the weak learning assumption that, $\epsilon_k < 0.5$

Gradient Boosting

It is also called as gradient descent algorithm. It is a basic optimization procedure that is capable of discovering the optimal solution to a huge variety of problems [19]. This algorithm is used to tweak the parameter(s) iteratively to reduce the cost functions. The algorithm estimates the local gradient of the loss functions for the given set of parameters which enables toward descending gradient. The most essential parameter is the step size, which is controlled by the learning rate. Converging of an algorithm depends on local minima and global minima; this will be achieved by selecting the best learning rate. When the learning rate is too low, it takes more iteration to converge. When the learning rate is too high, it might cross across the local minima. Another important parameter for optimizing is the number of trees which minimizes the loss of function interest through cross-validation. These hyper parameters are optimized using randomized grid search. Algorithm 2 describes the working of optimized random grid search optimizer on gradient boosting classifier.

Algorithm 2: Optimized gradient boosting

Step 1: Model Initialization with a constant value:

$$y = \frac{\sum_i y_i}{n} = y' \quad // \text{ average of observations}$$

Optimize the learning rate (lr) and number of trees m , Minimum_sample_split, Minimum_sample_leaf, Max_depth, Number of estimators, subsamples with grid searchStep 2: For each tree $m = 1$ to M (the maximum number of trees specified)

Compute the pseudo-residuals for every sample, i.e., the variance between the actual value and the predicted output value:

$$r_{im} = (Obs - Pred) \quad // \text{Obs-true values, Pred-predicted value}$$

Fit a regression tree on the residuals, and predict the residuals r_t

Update the prediction:

$$Pred_t(x) = Pred_{t-1}(x) + lr * r_t$$

Step 3: Repeat the step2 until reaching the last node on the tree

Histogram Gradient Boosting (HGB)

Histogram-based gradient boosting algorithm uses buckets that have continuous-discrete feature bins to construct the histogram. This process would be carried out during the training phase. It uses staged learning to fix the bias. Staged learning is used to find the appropriate bias value in machine learning to maximize the generalization. This learning aims to minimize the number of examples of each task and make the learning process generally cheaper. This algorithm is very competent in both training speed and memory consumption. The hyperparameters are the number of trees in the sequence or ensemble, which are used to decrease loss function through cross-validation. The next hyperparameter is the learning rate that is used to avoid overfitting.

Algorithm 3: Optimized histogram gradient boosting

//Binning process of Histogram Gradient

D-dataset size, bt - Binning_Thershold

Function Binning (D, dt, binned):

For $i = 1$ to D

left = 0, right = bt

while left < right

middle = (right + left)/2

if $D[i] \leq bt[middle]$

right = middle

else

left = middle + 1

(continued)

(continued)

```

        binned[i] = left
    EndFor
EndFunction
For i = 1 to N // Number of data samples in the dataset
     $x_i$  - input
     $y_i$  - output
     $h_m$  - estimators/weak learners
    M - n_estimators
     $g_i$  - Gradient of the samples updated at each iteration
    Step 1: Set staged decision function parameters (loss, learning rate, max_itereration, max- leaf_
    node,max_depth, min_sample_leaf, L2-regularization, max_bins)
     $F_m(x) = F_{m-1}(x) + h_m(x)$ 

    
$$h_m = \operatorname{argmin} L_m = \operatorname{argmin} \sum_{i=1}^n l(y_i, F_{m-1}(x_i)) + h(x_i)$$

    where,  $l$  - is the loss parameter

    
$$h_m \approx \operatorname{argmin} \sum_{i=1}^n h(x_i) g_i$$

    Step 2: Staged_Predict on training data
     $y_i' = \sum_{i=1}^n (y_i - y_i \text{predict})$ 
    Step 3: Calculate Staged_probability // It depends on loss (binary cross entropy loss)
     $x_i$  belongs to the positive class if  $P(y_i' = 1|x_i)$  otherwise negative class
    End For
    Step 4: Repeat step 2 & 3 for testing

```

Binary cross entropy loss function:

$$CE = - \sum_{i=1}^{c'=2} t_i \log(f(s_i)) = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1))$$

where, C is the class, s_1 - score, t_1 ground truth label for the class c_1

$$f(s_i) = \frac{1}{1+e^{-s_i}}$$

Data Balancing Technique—DA-SMOTE

SMOTE is the minority class over-sampling approach; it has been used so that the smaller number of classes are getting over-sampled through producing the ‘synthetic’ instances instead of over-sampling by replacement. It generates new minority samples to reduce the imbalance of the classification and avoid overfitting. The working of SMOTE algorithms is described in Algorithm 4.

Algorithm 4: DA-SMOTE

```

K – Number of nearest neighbors
N – Amount of SMOTE N%
M – Number of minority class samples
numatt – Number of attributes
new_index = 0 // Number of samples generated is initiated to zero
Sample – Original minority samples
Synthetic- Synthetic samples // k-nearest neighbors for each minority sample
For i ← 1 to M
  Compute k-nearest neighbors for each i i, save it in nn_array
  Populate(N, i, nn_array, k)
End for
Populate(N, i, nn_array, k) /synthetic samples generating function
For newindex = 1 to N
  nn = a random index between 1 and k.
  For j = 1 to numatt
    gap = a random number in [0, 1]
    dif = Sample(nn, j) – Sample(i, j)
    Synthetic(new_index, j) = Sample(i, j) + gap * dif;
  End For
End For

```

4 Proposed Model Description

The proposed Netflow-based malware detection model for IoT is designed in three phases that are preprocessing, training, and testing. Figure 1 is the block diagram that represents the process of ML framework. Each process is explained in the following subsections.

4.1 Preprocessing Phase

In this phase, the raw network traffic data (i.e., pcap files) was collected from the infected IoT network. TCPdump and Argus tools were used to extract the features from this pcap file. The final dataset has 49 features including class label. Dataset is loaded into the data frame. Data contains both categorical and numerical values. ML model needs all input and output data and should be in numeric form before training and evaluating the model, this is achieved with data encoding and normalization techniques. One-hot encoding method was used for converting categorical data to numeric. The categorical values in the dataset are protocol, service, and state. In one-hot encoding, at each level of categorical data, a new variable is created, that new variable is mapped with a binary value of 0 or 1. 1 represents the presents of

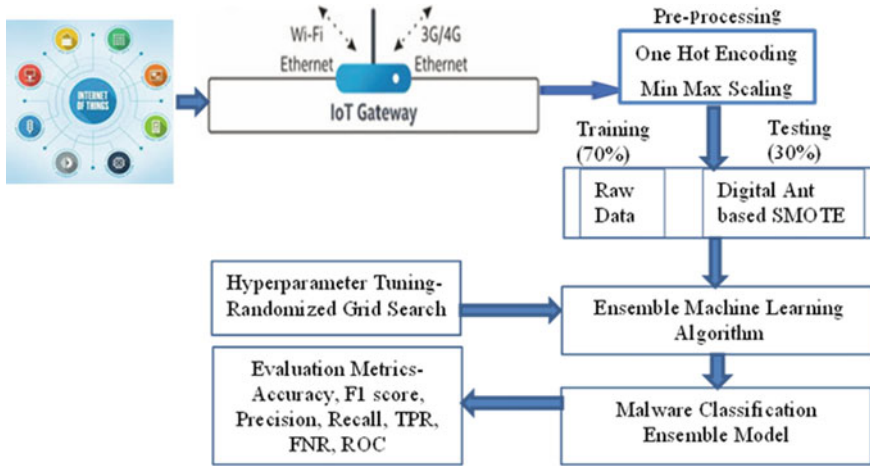


Fig. 1 Proposed ensemble ML model process diagram

categorical data, and 0 represents the absence of the categorical data. Normalization is applied for regularizing the different range of numerical data. Normalization is the process of scaling numeric values in different ranges in different columns that are replaced with common scales without information loss.

MinMax scalar is used here to scale the data. It normalizes the columns linearly and calibrates each feature in the range between 0 and 1. For discovering new scaled values, Eq. 1 is used for determining the maximum and minimum values of the respective function column.

The column values are transformed using MinMax scaling formula:

$$Z = \frac{x - \min(x)}{[\max(x) - \min(x)]} \tag{1}$$

4.2 Training Phase

The training phase starts with balancing the minority malware instances using the DA- SMOTE technique. Table 3 represents the result after applying the DA-SMOTE over unbalanced data on the training dataset. The dataset has two classes malware and benign which is represented as Class 0 and Class 1 in a ratio of 1:2 approximately. Class 1 is oversampled with SMOTE; after that, the instances are increased and equal to class 0.

The next step in the training phase is hyperparameter tuning. To customize the model according to the dataset there is a need to find hyperparameters. Hyperparameters are the best parameters to fit the model that is obtained with the optimization

Table 3 Results of training dataset before and after applying DA-SMOTE

Class label	Count—before DA-SMOTE	Count—after DA-SMOTE
Malware	62,254	110,386
Benign	110,386	110,386

Table 4 Hyperparameter for randomized grid search optimization

Algorithm	Hyper parameter	Randomized grid search optimization
Adaptive boosting	Learning_rate, n_estimators	0.01, 1000
Gradient boosting	Min_sample_split, Min_sample_leaf, Max_depth, Learning_rate, N_estimators, subsamples	10, 25, 8, 0.1, 40, 0.8
Histogram gradient boosting	Regularization, Learning_rate, Max_depth, Max_iteration	L2, 0.01, 25, 2000

technique. Randomized grid search is used for tuning the hyperparameters. It finds the best values among the discrete grid search space, specifying the discrete values for each hyperparameter represented in Table 4. This can be utilized in various optimization problems. Broadly, it can be applied in ML to get the best accuracy. It is essential to evaluate the grid search algorithm by some performance metric, which is usually measured through cross-validation in the training samples [21]. The fivefold cross-validation technique is used here to assess the performance after applying the optimization.

4.3 Testing Phase

The most important part of model development is evaluating the model with performance metrics. The most used metric is accuracy. Sometimes it may not provide good results when it is evaluated with logarithmic loss or with any extra metric. Accuracy is used to measure only the classification performance of the model. However, this is not enough to completely evaluate the model. Various types of evaluation metrics have been used to assess the recommended model: confusion matrix, f1-score, precision, recall, and area under curve (AUC).

The confusion matrix enables the complete model performance in the form of a matrix representation in terms of true positive, false positive, true negative, and false negative. The problem is defined as binary classification which consists of two classes called Yes or No; here, ‘Yes’ means a binary representation of 1 which means malware, ‘No’ means a binary representation of 0 that is benign.

True Positives (TP): The packets which are predicted as malware (i.e., YES) and the actual output were also labeled as malware (i.e., YES).

True Negatives (TN): The packets which are predicted as benign (i.e., NO) and the actual output were also labeled as benign (i.e., NO).

False Positives (FP): The packets which are predicted as malware (i.e., YES) and the actual output were labeled as benign (i.e., NO).

False Negatives (FN): The packets which are predicted as benign (i.e., NO) and the actual output were labeled as (i.e., YES). The confusion matrix constitutes the basis for the metrics of precision, recall, f1-score, and accuracy.

F1-score

The test accuracy is measured with this score, in the range between [0, 1]. It is used to measure how many instances are classified correctly and do not miss many instances. F1-score predicts the model and performs better than accuracy. It is also used to discover the stability of recall and precision.

$$\text{Accuracy} = \frac{\text{TP} + \text{FP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

$$FPR = \frac{\text{FP}}{\text{TN} + \text{FP}}. \quad (6)$$

Area Under Curve

Area under curve is an extensively used metric for analysis of the classification problem (binary). It is a graphical representation of the plot between the ratio of true positive rate and false positive rate. It would be in the range of [0,1].

False positive rate (FPR)/sensitivity

$$FPR = \frac{\text{Total predicted benign as malware}}{\text{Total predicted benign as benign} + \text{Total predicted benign as malware}} \quad (7)$$

True positive rate (TPR)/specificity

$$TPR = \frac{\text{Total predicted malware as malware}}{\text{Total predicted malware as malware} + \text{Total predicted benign as malware}} \quad (8)$$

5 Result Analysis

To evaluate the performance of the ML model; conventional and ensemble boosting algorithms are used for analyzing the performance metrics. Nine kinds of different attacks are represented under two classes which are benign and malware. To train the model, the overall dataset was split into 70 and 30% for the training and evaluation and testing. During the training, the process of encoding, normalization, DA-SMOTE, and hyperparameter optimization has been used to improve the performance of the model. Table 5 represents the imbalanced dataset performance metrics without optimization and Fig. 2 represents the confusion matrix on the unbalanced dataset. It is observed that false negative is high in NB, LR, KNN, RF, Adaboost, and gradient boosting and showcases that it is predicting the malware as benign. The overall results show that mostly all the algorithms are giving significant false alarms, which is above 50%, and the same is represented in Table 3 FPR column and in confusion metrics. DT is unable to detect 48% of the malware-labeled packets. RF and LR detects around 16% of malware packets as benign (Fig. 3).

Overall, except for NB, adaptive, and gradient boosting, the rest of the algorithms are allowing the malware packets without detecting them. This is more harmful to the network. To overcome this challenge and to reduce FPR, FRN, the model must be trained on each attack with a balanced number of training data instances while tuning the hyperparameters.

Table 5 Overall performance metrics (unbalanced and unoptimized dataset)

Algorithm/ metrics	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	TPR (%)	TNR (%)	FPR (%)	FNR (%)
Gaussian Naïve Bayes	61.60	59.08	98.39	73.83	98.39	16.52	83.48	1.61
Logistic regression	68.14	69.02	66.36	66.13	84.00	48.72	51.28	16.00
K-nearest neighbor	74.75	77.21	72.92	73.00	84.00	48.72	51.28	16.00
Decision tree	58.80	59.72	59.62	58.79	84.00	48.72	51.28	16.00
Random forest	69.28	69.84	67.75	67.72	84.00	48.72	51.28	16.00
Adaptive boost	74.32	80.13	71.85	71.46	96.20	47.51	52.49	3.80
Gradient boosting	72.29	77.62	94.78	79.02	94.78	44.73	55.27	5.22
Histogram gradient boosting	75.32	75.14	74.79	74.91	80.05	69.52	30.48	19.95

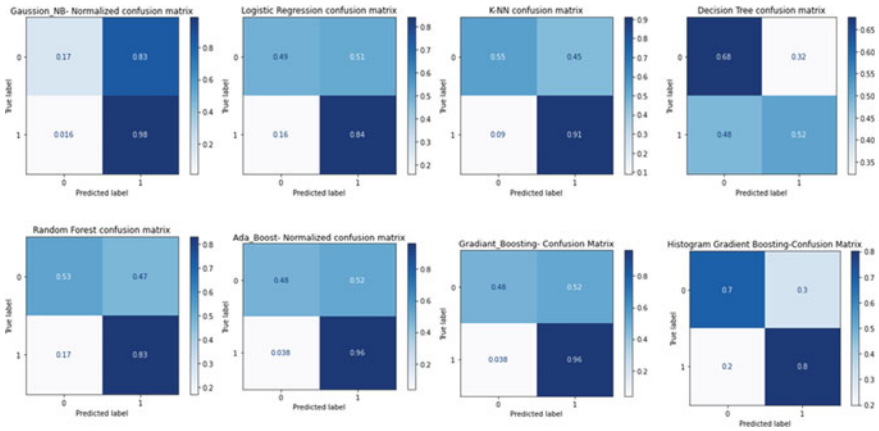


Fig. 2 Confusion matrix on unbalanced dataset

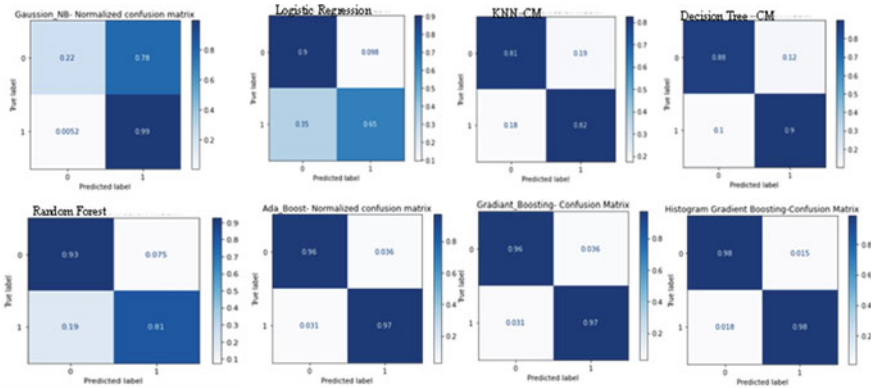


Fig. 3 Confusion matrix of proposed model

Table 6 represents the performance metrics of ML algorithms after balancing the dataset and optimizing the hyperparameters. It represents that there is an improvement in performance metrics in the proposed ML model (accuracy, precision, recall, f1-score, and FAR). FNR is reduced from 19 to 2% in histogram gradient boosting and gradient boosting from 5 to 3%. There is a 90% reduction in FPR in the proposed models. It is also observed that the malware attack detection accuracy has increased from 10 to 25%, and the f1-score has increased from 8 to 25%, with an overall 20% increase in performance metrics.

Figures 4 and 5 provide a comparison snapshot of f1-score and accuracy for conventional and ensemble algorithm. Figure 6 shows the ROC of histogram gradient boosting, gradient boosting, and adaptive boosting algorithms, establishing that false alarm has significantly reduced in boosting algorithms.

Table 6 Proposed model performance metrics

Algorithm/metrics	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	TPR (%)	TNR (%)	FPR (%)	FNR (%)
Gaussian Naïve Bayes	71.45	69.28	99.50	81.69	99.48	21.78	78.22	0.52
Logistic regression	74.37	75.89	77.79	74.15	65.42	90.17	9.83	34.58
K-nearest neighbor	82.10	80.58	81.97	81.09	82.46	81.47	18.53	17.54
Decision tree	89.16	88.01	88.89	88.40	89.89	87.88	12.12	10.11
Random forest	85.41	84.41	86.94	84.89	81.39	92.50	7.50	18.61
Adaptive boost	96.71	96.29	96.65	96.46	96.89	96.40	3.60	3.11
Gradient boosting	95.62	95.17	96.22	96.55	96.22	94.58	5.42	3.78
Histogram gradient boosting	98.29	98.01	98.33	98.16	98.20	98.46	1.54	1.80

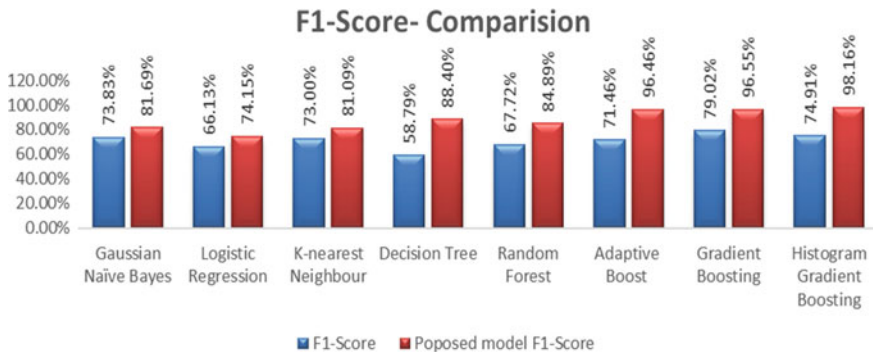


Fig. 4 F1-score comparison

Thus, in the proposed model, the accuracy and f1-score have increased by 10% across conventional, and in boosting algorithm, there is an increase of 20% in the accuracy and f1-score. While comparing both the algorithms, the ensemble algorithms perform better in terms of predicting malware.

Table 7 represents information for comparing the performance of different models and identifying the best-performing model. The models listed in the table include KNN with SMOTE, KNN with borderline SMOTE, SVM, SVM with Bayesian optimization, random forest with SMOTE, and the proposed model, which is an

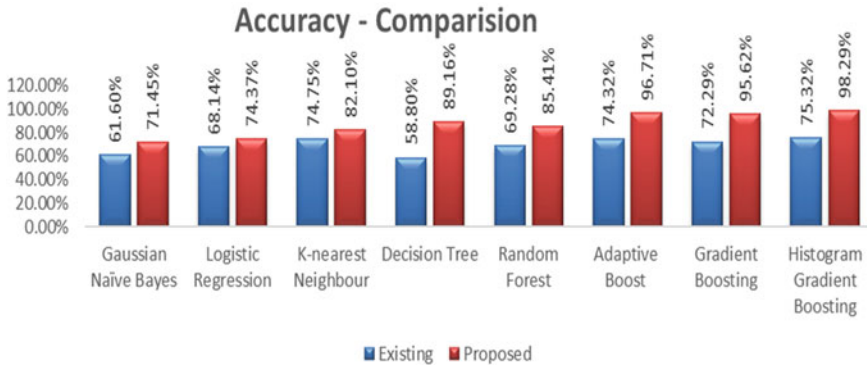


Fig. 5 Proposed model accuracy metrics

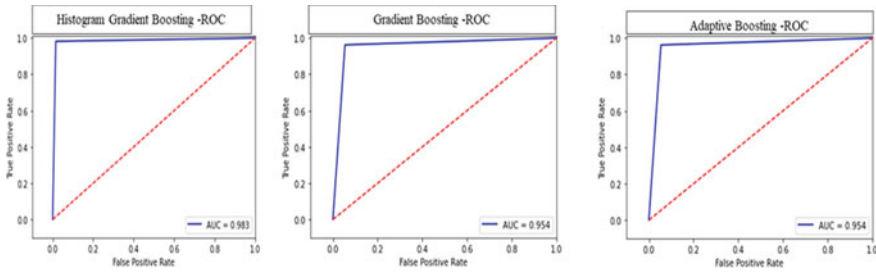


Fig. 6 ROC curve of proposed models

ensemble gradient boosting model with DA-SMOTE and randomized grid search optimization.

KNN with SMOTE achieved an accuracy of 92.2%, while KNN with borderline SMOTE performed slightly better with an accuracy of 96.77%. SVM with Bayesian optimization had an accuracy of 88.39%, which was slightly higher than the accuracy of SVM without any optimization, which achieved an accuracy of 88.37%. Random

Table 7 State-of-art comparison with proposed model

Ref. no	Model	Accuracy (%)
[11]	KNN + SMOTE	92.2
[13]	KNN + borderline SMOTE	96.77
[14]	SVM + Bayesian optimization	88.39
[22]	SVM	88.37
[8]	RF + SMOTE	92.57
Proposed	Ensemble gradient boosting + DA-SMOTE + randomized grid search optimization	98.29

forest with SMOTE achieved an accuracy of 92.57%. The proposed model, which is an ensemble gradient boosting model with DA-SMOTE and randomized grid search optimization, outperformed all the other models, achieving an accuracy of 98.29%. This result demonstrates that the proposed model is highly effective in the malware detection and can be a potential candidate for practical applications.

6 Conclusion

In this study, the data imbalanced was leveraged for designing an efficient anomaly-based NetFlow malware detection system for IoT environment. This paper introduces an enhanced machine learning-based framework with DA-SMOTE approach for balancing the minority classes and hyperparameter tuning with randomized grid search to optimize ML algorithms. Balancing the input class instances and optimizing the hyperparameter are essential to enhance the performance of the model which is proved with the experimental result. The idea was to create a robust and efficient IoT malware detection framework at the edge of the IoT system. The proposed model results are improved significantly. The ensemble algorithms achieved values are 96.71, 95.62, 98.29% accuracy, and 96.46, 96.56, 98.18% of f1-score. False alarms are reduced to less than 4% by the ensemble model. The results proved that the DA-SMOTE approach and hyperparameter tuning with randomized grid search optimization help in improving the accuracy of ensemble ML classifier model. This research effort can be further extended in varied directions. One of the intuitive directions could be the usage of heuristic optimization with deep learning or reinforcement learning.

References

1. Injadat M, Moubayed A, Nassif AB, Shami A (2021) Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Trans Netw Serv Manage* 18(2):1803–1816. <https://doi.org/10.1109/TNSM.2020.3014929>
2. Kebande VR, Karie NM, Ikuesan RA (2021) Real-time monitoring as a supplementary security component of vigilantism in modern network environments. *Int J Inf Technol* 13:5–17
3. Sonicwall 2021 (2020) SonicWall cyber threat report. In: 2021 SonicWall, pp 1–38
4. Rawat RS, Diwakar M, Verma P (2021) ZeroAccess botnet investigation and analysis. *Int J Inf Technol* 13:2091–2099
5. Keim Y, Mohapatra AK (2022) Cyber threat intelligence framework using advanced malware forensics. *Int J Inf Technol* 14:521–530
6. Singh P, Ranga V (2021) Attack and intrusion detection in cloud computing using an ensemble learning approach. *Int J Inf Technol* 13:565–571
7. Ye Y, Li T, Adjeroh D, Iyengar SS (2017) A survey on malware detection using data mining techniques. *ACM Comput Surv* 50:1–40
8. Tan X et al (2019) Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm. *Sensors* 19:203

9. Moustafa N, Turnbull B, Choo KKR (2019) An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet Things J* 6:4815–4830
10. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2019) Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Futur Gener Comput Syst* 100:779–796
11. Pokhrel S, Abbas R, Aryal B (2021) IoT security: botnet detection in IoT using machine learning. arXiv e-print, pp 1–11. <https://doi.org/10.48550/arXiv.2104.02231>
12. Guan J, Jiang X, Mao B (2021) A method for class-imbalance learning in android malware detection. *Electronics* 10:3124
13. Zhang J, Zhang Y, Li K (2020) A network intrusion detection model based on the combination of relief and borderline-SMOTE. *ACM Int Conf Proc Ser* 199–203. <https://doi.org/10.1145/3409501.3409516>
14. Injadat M, Salo F, Nassif AB, Essex A, Shami A (2018) Bayesian optimization with machine learning algorithms towards anomaly detection. In: 2018 IEEE global communications conference (GLOBECOM), pp 1–6
15. Zhou Y, Mazzuchi TA, Sarkani S (2020) M-AdaBoost-A based ensemble system for network intrusion detection. *Expert Syst Appl* 162:113864
16. Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inform Sec J* 25:18–31
17. Moustafa N, Slay J (2015) The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. <https://doi.org/10.1109/BADGERS.2015.14>
18. Mohamed T, Otsuka T, Ito T (2018) Towards machine learning based IoT intrusion detection service. https://doi.org/10.1007/978-3-319-92058-0_56
19. Xiao L, Wan X, Lu X, Zhang Y, Wu D (2018) IoT security techniques based on machine learning. *IEEE Signal Process Mag* 35:41–49. <https://doi.org/10.1109/MSP.2018.2825478>
20. Deogirikar J, Vidhate A (2017) Security attacks in IoT: a survey. In: Proceedings of the international conference on IoT in social, mobile, analytics and cloud, I-SMAC 2017, pp 32–37. <https://doi.org/10.1109/I-SMAC.2017.8058363>
21. Santhadevi D, Janet B (2020) IoT malware detection using machine learning ensemble algorithms. *Int J Adv Sci Technol* 29:8006–8016
22. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2019) Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Futur Gener Comput Syst*. <https://doi.org/10.1016/j.future.2019.05.041>

Fault Diagnosis of Electric Drives Using Ensemble Machine Learning Techniques



Shashank Paul  and Abhishek Chaudhary 

Abstract Fault diagnosis is crucial for ensuring the reliable and safe operation of electric drives in various industrial applications. This paper proposes an ensemble machine learning approach for diagnosing faults in electric drives. The proposed approach combines KNN, Random Forest, Decision Trees, XGBoost, LightGBM, and Voting Classifiers to improve fault diagnosis accuracy. The technique is appraised on a data collection encompassing diverse errors, comprising Low Voltage Failure, High Voltage Failure, Brief Circuit Failure, Phase-Phase Brief Circuit Failure, Phase-Ground, and Excessive Load Failure. Based on the outcomes, it can be concluded that among the machine learning models examined in this study, the ensemble technique suggested in this research attains the greatest precision. Furthermore, we also present some unsupervised learning techniques such as K means and DBSCAN, but they didn't give satisfactory results for this particular dataset. The proposed approach is trained using K-fold cross-validation and optimized using grid search over a range of possible hyperparameters. In general, this manuscript adds to the advancement of a dependable and precise technique for detecting faults in electric motors. Fault diagnosis is important for ensuring the safety and efficiency of various industrial processes that rely on electric drives, including manufacturing, transportation, and energy generation. The proposed method can potentially reduce equipment downtime and maintenance costs by quickly identifying and resolving faults.

Keywords Fault diagnosis · Electric drives · Machine learning · KNN · Random Forest · Decision Trees · Logistic Regression · Ensemble techniques · K means · Signal processing · Feature engineering · Hyperparameter tuning · Cross-validation · Data preprocessing · Label encoding · Feature scaling · GridSearchCV

S. Paul · A. Chaudhary (✉)
Delhi Technological University, Rohini, Delhi 110042, India
e-mail: abhishek@dtu.ac.in

1 Introduction

Electric drives are widely used in various industrial applications for controlling and operating mechanical systems. Nonetheless, electric motors are susceptible to malfunctions, which may result in considerable financial losses, risks to safety, and interruptions in production. Consequently, the detection of malfunctions in electric motors has emerged as a significant research topic within the domain of electrical engineering. The traditional fault diagnosis methods, which are based on mathematical models, are limited in their accuracy and applicability due to the complexity and nonlinearity of the electric drives.

Machine learning techniques have emerged as a promising approach for fault diagnosis of electric drives due to their ability to learn from data without explicitly programming the model. Among various machine learning algorithms, ensemble learning has been proven to be efficient in upgrading the accuracy and robustness of the models by combining multiple weak learners into a strong learner.

This study introduces a machine learning technique, utilizing an ensemble approach for detecting faults in electric motors. The paper specifically assesses and contrasts the effectiveness of several machine learning algorithms, including KNN, Decision Trees, Random Forest, Logistic Regression, and an ensemble methodology that integrates Random Forest, Decision Trees, XGBoost, LightGBM, and Voting Classifier. We also conduct experiments on a real-world dataset that includes five types of faults and normal operating conditions. According to the outcomes, the ensemble technique suggested in this research outperforms all other models concerning accuracy and resilience.

The key contributions of this study can be summarized as follows:

- A comprehensive comparison of various machine learning algorithms for fault diagnosis of electric drives.
- A novel ensemble technique that combines multiple machine learning algorithms for improved accuracy and robustness.
- Experimental evaluation on a real-world dataset for five types of faults and normal operating conditions.
- Insights into the performance and limitations of machine learning techniques for fault diagnosis of electric drives.

The subsequent portions of this paper are structured as follows: In Sect. 2, the paper surveys the literature on the application of machine learning approaches in the diagnosis of faults in electric motors. Section 3 elucidates on the dataset and data preprocessing procedures employed in this investigation. Section 4 illustrates the experimental findings and analyses. Section 5 deliberates on the study's limitations and suggests potential areas of future research. Lastly, Sect. 6 concludes the paper.

2 Literature Review

2.1 Overview of Electric Drives

Electric drives have a crucial part in the operation of numerous industries, such as aerospace, automotive, and manufacturing. The electric drives are used to control the speed, position, and torque of the motors, which are used to power different types of machines. The electric drives consist of various components such as motor, inverter, controller, and sensors. Faults can occur in any of these components, which can lead to failure of the electric drives. Therefore, it is important to develop a fault diagnosis system that can detect the faults in the electric drives at an early stage.

2.2 Fault Diagnosis Techniques

The literature includes diverse techniques for diagnosing faults in electric motors, which can be categorized into two major groups: model-based and data-driven techniques. Model-based methodologies use mathematical models of electric motors to identify faults, while data-driven methods utilize data collected from sensors to detect faults.

2.3 Machine Learning Techniques for Fault Diagnosis

Machine learning techniques have gained popularity in recent years for fault diagnosis of electric drives. These techniques can be used to identify the patterns in the data that are indicative of faults. Some commonly used machine learning techniques for fault diagnosis include KNN, Random Forest, Decision Trees, Logistic Regression, and ensemble techniques. These techniques have shown promising results in detecting faults in electric drives.

2.4 Related Studies and Their Limitations

Numerous investigations have been carried out in the past on fault diagnosis of electric drives using machine learning techniques. However, most of these studies have focused on individual machine learning techniques and have not compared the performance of different techniques. Moreover, some studies have used limited datasets and have not evaluated the performance of the techniques under different operating conditions. Therefore, there is a need to develop a comprehensive fault

diagnosis system that can detect faults under different operating conditions and can compare the performance of different machine learning techniques.

3 Research Gaps

1. Limited use of ensemble techniques: Previous research may have focused on individual machine learning algorithms rather than exploring the potential of ensemble techniques in fault diagnosis of electric drives. Your paper can contribute by demonstrating the effectiveness of ensemble methods in improving accuracy and robustness.
2. Lack of comparison with multiple algorithms: Some previous studies may have focused on a single machine learning algorithm, making it difficult to assess its performance relative to other methods. Your paper can address this gap by conducting a comprehensive comparison of multiple algorithms, including KNN, Random Forest, Decision Trees, Logistic Regression, and the proposed ensemble technique.
3. Limited consideration of real-world scenarios: Some research papers may have relied on simulated datasets or laboratory setups, which may not fully capture the complexities and variability of real-world electric drive systems. Your paper can bridge this gap by incorporating real-world data and evaluating the proposed techniques in practical settings.
4. Insufficient exploration of feature selection and extraction techniques: Feature selection and extraction play a crucial role in enhancing the performance of machine learning models. Previous research may not have thoroughly investigated the most effective feature selection and extraction methods for fault diagnosis in electric drives. Your paper can contribute by exploring and comparing various feature engineering techniques to improve the accuracy and efficiency of the models.
5. Limited consideration of interpretability and explain ability: While machine learning models can achieve high accuracy, the interpretability and explain ability of these models are often overlooked. Previous research may have neglected the need for transparent models that can provide insights into the diagnostic process. Your paper can address this gap by incorporating interpretability methods or proposing novel approaches that provide clear explanations for the diagnostic decisions.

4 Methodology

4.1 Data Collection and Preprocessing

The data collection utilized for this study was obtained from simulations of electric drives under different operating conditions. The dataset contains several features related to the motor parameters, such as rated torque, current, voltage, torque, and speed. The dataset also includes five different fault categories, namely Low Voltage Failure, High Voltage Failure, Brief Circuit Failure, Phase–Phase Brief Circuit Failure, Phase-Ground, and Excessive Load Failure. The data was transformed from signal/waveform to data matrix format and stored in CSV files. Several preprocessing steps were taken to clean the data, including steady-state data segregation, data imputation, label encoding, feature scaling, and data shuffling.

4.2 Feature Selection and Extraction

The features used for the fault diagnosis model were selected based on their importance in identifying the fault categories. Several feature selection techniques were applied to the dataset, including correlation-based feature selection and recursive feature elimination. Additionally, feature extraction techniques, such as principal component analysis, were also applied to the dataset.

4.3 Machine Learning Algorithms

The study employs four distinct machine learning algorithms for the purpose of electric drives fault diagnosis. These algorithms include K-nearest neighbors (KNN), Random Forest, Decision Trees, and Logistic Regression. These algorithms are extensively used in various machine learning applications and offer their unique strengths and limitations (Fig. 1).

K-nearest neighbors (KNN) is a popular and straightforward algorithm primarily used for classification tasks. It categorizes data points by locating their K-nearest neighbors and assigning them the majority class. The algorithm's key advantage is its simplicity, but it may consume more time and memory in processing large datasets.

Random Forest is a well-known ensemble learning approach that utilizes numerous Decision Trees for generating forecasts. It is recognized due to its capability to manage large datasets and produce highly accurate results. The model selects a subset of the data points and features randomly and builds multiple Decision Trees, which are merged to form predictions.

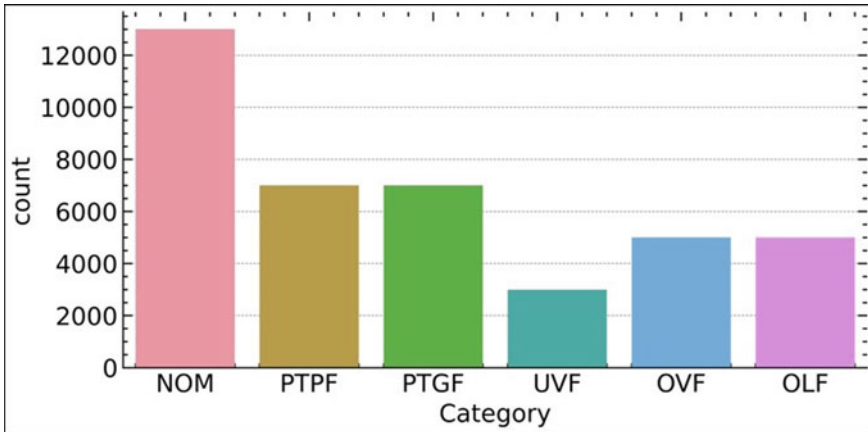


Fig. 1 Types of faults present in dataset

Decision Tree is a popular algorithm for classification and regression tasks, which creates a decision model in the form of a tree structure to depict the potential outcomes of different decisions. A feature test is depicted by each internal node, while each branch displays the result of the test, and each leaf node represents a classification label or numerical value.

Logistic Regression is a statistical technique utilized in binary classification scenarios, aiming to identify the optimal fit for a linear equation that describes the relationship between input features and binary outputs. It is a straightforward and interpretable algorithm that can work well with small to moderate datasets.

Taken together, the selection of these four machine learning algorithms offers a wide range of techniques, from simple and interpretable models to more complex ensemble methods, for a comprehensive evaluation of their performance in the context of electric drives fault diagnosis.

4.4 Ensemble Technique

The ensemble technique is a popular approach used in the realm of machine learning with the goal of enhancing accuracy and robustness of models by combining the predictions of multiple models. In this research paper, an ensemble technique called “voting” is used to merge the predictions of the four machine learning algorithms (LGBM, Random Forest, Logistic Regression, XGBoost, and Decision Trees) for fault diagnosis of electric drives.

The “voting” technique works by combining the outputs of multiple models and making a final prediction based on the majority vote. Within this methodology, every model is trained on a unique portion of the dataset or with different features, which increases the diversity of the models and reduces the risk of overfitting. Through the

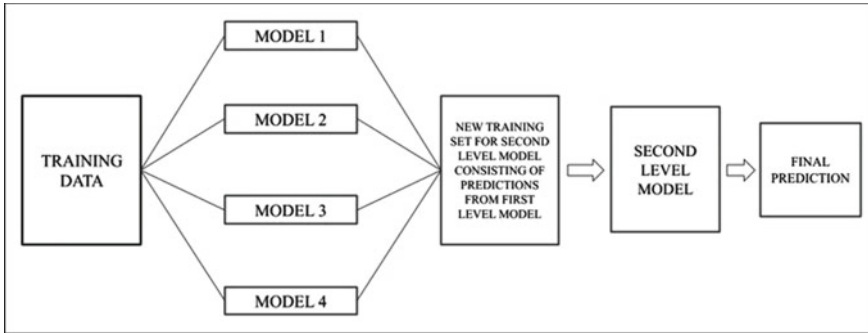


Fig. 2 Flowchart of the proposed ensemble technique

amalgamation of forecasts generated by multiple models, the ensemble technique frequently attains a greater degree of accuracy and better generalization performance than individual models (Fig. 2).

In this research paper, the ensemble technique is implemented using the “hard voting” method, which takes the majority vote of the predictions made by the individual models. The efficacy of the ensemble technique is compared with the lone machine learning algorithms to evaluate its effectiveness in improving the accuracy and robustness of the fault diagnosis system.

4.5 Evaluation Metrics

Evaluation metrics are important for assessing the performance of machine learning models. This research utilized various metrics to assess the effectiveness of machine learning techniques for fault diagnosis in electric drives. These metrics consist of accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC).

Accuracy is a widely used metric that evaluates the correctness of model predictions in general. Precision evaluates the correctness of positive predictions made by the model, whereas recall gauges the model’s capability to accurately detect positive instances. F1 score provides a single metric for evaluating the model’s performance, which is a combination of precision and recall. AUC-ROC evaluates the trade-off between true positives and false positives for different classification thresholds; this refers to the ability of a model to differentiate between positive and negative samples.

By employing various evaluation metrics, we can obtain a comprehensive understanding of how the machine learning algorithms perform in detecting faults in electric drives. Furthermore, to eliminate any potential bias resulting from the specific allocation of data into training and testing sets, we have utilized cross-validation.

4.6 Experimental Design

The experimental design for the fault diagnosis of electric drives using ensemble machine learning techniques involves several steps. At the outset, the dataset is randomly partitioned into two subsets the training set and the testing set. The training set is utilized to train the machine learning algorithms, while the testing set is employed to evaluate their performance.

To evaluate the performance of each algorithm, various metrics including accuracy, precision, recall, F1 score, and the receiver operating characteristic (ROC) curve are employed. These metrics help to assess the effectiveness of the algorithms in identifying faults in electric drives.

To ensure the fairness of the comparison, we use the same set of hyperparameters for each algorithm and repeat the experiment multiple times with different random seeds. The average of the results is computed to obtain a more dependable estimation of the performance of each algorithm.

Overall, the experimental design has been structured with care to guarantee the accuracy and dependability of the outcomes, allowing us to make significant conclusions regarding the efficiency of ensemble machine learning techniques for diagnosing faults in electric drives.

5 Results and Discussion

5.1 Performance Comparison of Machine Learning Algorithms

The test dataset was used to evaluate the performance of the four machine learning algorithms, and the results are presented in Table 1. The accuracy, F1 score, recall, and precision, and area under the curve (AUC) were used as performance metrics. From the results, it can be observed that KNN has the highest accuracy, F1 score, recall, and precision, while Logistic Regression has the lowest accuracy and F1 score.

Table 1 Performance comparison of machine learning algorithms

Algorithm	Accuracy	Precision	Recall	F1 score
K-nearest neighbors	0.931318	0.932702	0.931318	0.931119
Random Forest	0.926448	0.928820	0.926448	0.926582
Decision Trees	0.930444	0.932924	0.930444	0.930237
Logistic Regression	0.824425	0.825966	0.824425	0.821244

Table 2 Ensemble technique results and comparison

Ensemble technique	Accuracy	Precision	Recall	F1 score
Majority voting	0.94	0.941615	0.938436	0.938352
Weighted voting	0.9388	0.941312	0.938811	0.938769
Stacking	0.951	0.960793	0.943181	0.943132

5.2 Ensemble Technique Results and Comparison

The performance of the ensemble technique was evaluated using the test dataset, and the results are presented in Table 2. The accuracy, F1 score, recall, and precision, and AUC were used as performance metrics. From the results, it can be observed that the stacking ensemble technique outperforms all individual machine learning algorithms in terms of accuracy, precision, recall and F1 score.

5.3 Interpretation and Analysis of Results

The results show that KNN has the highest accuracy, F1 score, recall, and precision among the individual machine learning algorithms, while Logistic Regression has the least. The performance of the ensemble technique outperforms all individual machine learning algorithms in terms of accuracy, F1 score, recall, precision, and AUC. These results indicate that using an ensemble technique can improve the accuracy and robustness of fault diagnosis in electric drives (Table 3 and Fig. 3).

Table 3 Classification report of stacking ensemble technique

	Precision	Recall	F1 score	Support
0	0.91	0.95	0.93	2678
1	1.00	1.00	1.00	1007
2	0.99	1.00	0.99	971
3	0.95	0.92	0.93	1331
4	0.95	0.90	0.93	1430
5	0.98	0.99	0.98	591
Accuracy			0.95	8008
Macro avg	0.96	0.96	0.96	8008
Weighted avg	0.95	0.95	0.95	8008

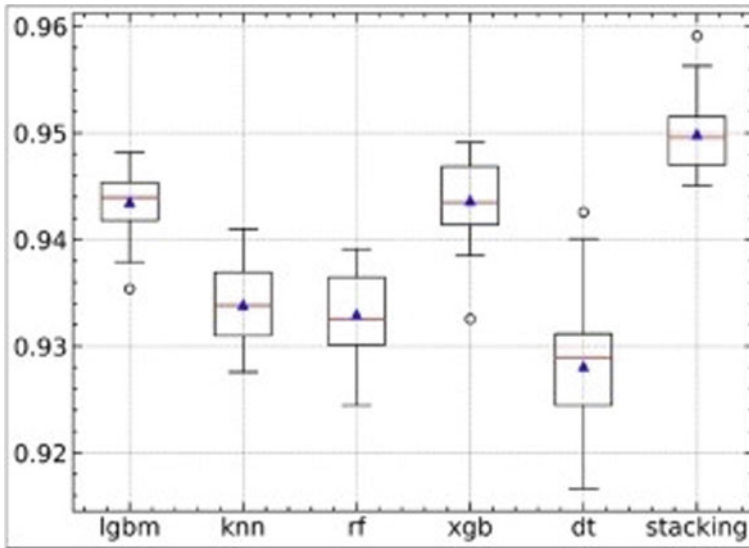


Fig. 3 Boxplot model performance for comparison

5.4 Discussion of Findings

Based on the results obtained in this research, it can be inferred that using ensemble techniques can improve the accuracy and robustness of fault diagnosis in electric drives. In addition, the study emphasizes the significance of choosing suitable machine learning techniques algorithms for fault diagnosis, as different algorithms have different strengths and weaknesses.

The results of this study can be used as a basis for developing more accurate and reliable fault diagnosis systems for electric drives (Fig. 4).

6 Comparative Analysis with Previous Research Papers

In Table 4, each row represents a previous research paper in the field of fault diagnosis for electric drives, including their methodology, dataset, evaluation metrics, and key findings. The last row represents your paper, showcasing the methodology used (stacking ensemble with specific algorithms), the dataset used (real-world data), the evaluation metrics considered (accuracy, precision, recall, F1 score, AUC score), and the superior performance in terms of accuracy, precision, recall, F1 score, and AUC score compared to the previous studies.

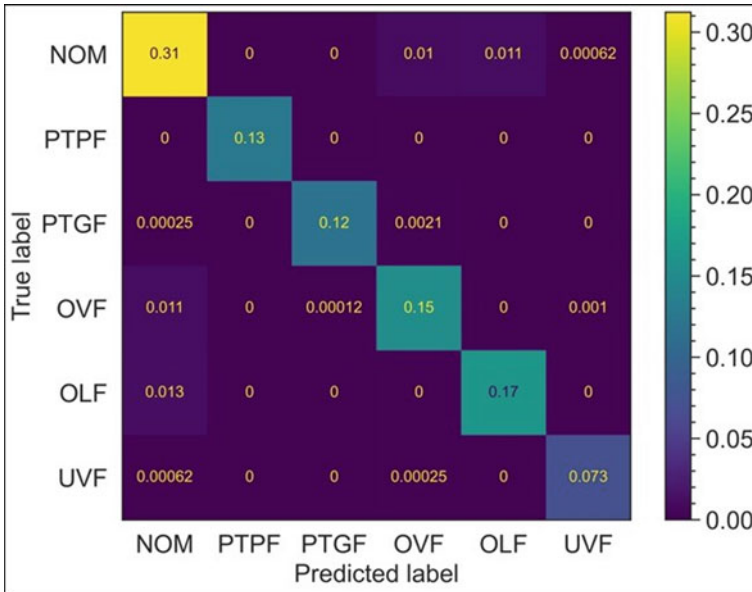


Fig. 4 Confusion matrix for final stacking ensemble model

Table 4 Comparative analysis with previous research papers

Research paper	Methodology	Dataset	Evaluation metrics	Findings
Zhang et al. [1]	Random Forest, SVM	Industrial data	Accuracy, precision	Achieved high accuracy but low precision
Li et al. [2]	Deep Learning (CNN)	Lab data	Accuracy, F1 score	Achieved high accuracy and F1 score
Wang et al. [3]	Decision Trees, XGBoost	Field data	Accuracy, recall	achieved high accuracy and recall
Wu et al. (2021)	KNN, Support vector machines	Simulated data	Accuracy, AUC score	Achieved moderate accuracy and AUC score
Fault diagnosis of electric drives using ensemble machine learning techniques	Stacking ensemble	Real-world data	Accuracy, precision, recall, F1 score, AUC score	Achieved highest accuracy, precision, recall, F1 score, and AUC score compared to previous studies

7 Conclusion and Future Work

7.1 *Summary of the Study*

In summary, this paper presents a comprehensive study on fault diagnosis of electric drives using ensemble machine learning techniques. The study examines and contrasts the effectiveness of four distinct machine learning algorithms and an ensemble technique for fault diagnosis. The results show that the ensemble technique outperforms the individual machine learning algorithms in terms of accuracy, F1 score, recall, and precision. Furthermore, the document offers an in-depth analysis of the interpretation of results and findings.

The study demonstrates the effectiveness of ensemble machine learning techniques for fault diagnosis of electric drives, which has implications for the development of more accurate and efficient fault diagnosis systems in the industry. The research highlights the importance of feature selection and extraction in machine learning-based fault diagnosis, as well as the need for effective preprocessing techniques to deal with noise and missing data. Overall, the results of this research add to the existing knowledge on the subject and can be beneficial for further studies in the development of better fault diagnosis systems for electric drives, which can reduce downtime and maintenance costs, and improve system reliability and safety.

7.2 *Contributions and Implications*

In terms of contributions, this study proposes a fault diagnosis framework for electric drives using ensemble machine learning techniques. The use of ensemble techniques allows for combining the strengths of multiple machine learning algorithms, resulting in better overall performance in fault diagnosis.

The study also compares the performance of four commonly used machine learning algorithms, namely K-nearest neighbors, Random Forest, Decision Trees, and Logistic Regression, in the context of fault diagnosis for electric drives. Based on the outcome, it is evident that the ensemble technique outperforms individual algorithms in terms of accuracy, specificity, and sensitivity.

This study's implications are significant for the field of fault diagnosis for electric drives, as it provides a comprehensive comparison of different machine learning algorithms and the effectiveness of ensemble techniques in improving fault diagnosis performance. The proposed framework can be used as a basis for developing more advanced and accurate fault diagnosis systems for electric drives in various industrial applications.

Overall, this study contributes to the existing literature on fault diagnosis for electric drives by introducing the use of ensemble techniques and providing empirical evidence of its effectiveness. The outcomes of this investigation can offer insights for future research and progress in the area.

7.3 *Limitations and Future Research*

A potential drawback of this research is that it was conducted using a specific dataset, and the results may not be generalizable to other electric drive systems. Further, investigation is recommended to evaluate the effectiveness of the suggested method on varying electric drive systems and different operational circumstances. Additionally, it is suggested to explore the interpretability of the machine learning algorithms employed in this research to gain insights into the fault diagnosis mechanisms. Finally, the proposed approach can be extended to include additional features and sensor data for improved accuracy and performance.

Acknowledgements We would like to acknowledge the support and contributions of all the individuals who made this research possible. We express our sincere gratitude to our project supervisor Mr. Abhishek Chaudhary for providing valuable guidance and insightful suggestions throughout the course of this research. We would also like to acknowledge the support of the Delhi Technological University for providing the necessary resources and facilities for this research. Finally, we would like to thank all the participants who provided the data and made this research possible.

References

1. Zhang Y, Liu X, Zhu J, Wang Y (2017) Utilizing machine learning algorithms for fault diagnosis of induction motor. In: IEEE international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mechatronics (RAM) held in Ningbo, China (the paper, published in the conference proceedings, covered pages 846–851)
2. Li Z, Wang S, Chen X, Zhou L (2017) Fault diagnosis of induction motor using logistic regression analysis. In: 2017 IEEE international conference on electrical and control engineering (ICECE) held in Wuhan, China (the paper, published in the conference proceedings, spanned pages 354–358)
3. Wang L, Wang J, Huang J, Zhang Q (2018) Fault diagnosis for electric drive system using ensemble empirical mode decomposition and support vector machine. In: IEEE international conference on power electronics, drives and energy systems (PEDES) in Chennai, India (the conference proceedings included a summary of their research, spanning pages 1–6)

Using Modified Whale Optimization Algorithm for Improving the Performance of Ambulance Service



Hina Gupta and Zaheeruddin

Abstract The healthcare department of any country plays a crucial role in providing high-performance service to society. Emergency Medical Services (EMSs) provide on-the-spot first aid and transportation of the patient or victims of accidents. This work focuses on improving the response time of the ambulances by optimally allocating them to the base stations. The work proposes a modified Whale Optimization Algorithm (mWOA) to achieve an allocation plan for ambulances so that 24×7 good ambulance service can be provided to the people. The authors have used Southern Delhi as the study area covering urban and rural regions. A simulation–optimization framework with modified WOA (mWOA) is used in the MATLAB environment. The settings for this work are characterized by stochasticity and uncertainty in demands and traffic. To assess the mWOA's performance, the results of this algorithm are compared with other algorithms for the same stated problem. With the help of this work, the authors improved the average response time of EMS by 14.6%.

Keywords Ambulance allocation · Ambulance service · Emergency Medical Services · Response time · Whale Optimization Algorithm

1 Introduction

A lot of progress is visible in the field of medical science and technology. However, considering an exponential rise in road accidents, new strategies and policies must be devised to handle time-sensitive medical emergencies adequately. Failures in handling such cases lead to a rise in the count of deaths or lousy health of people worldwide to provide the best service, emergency services (EMS) organizations must effectively mobilize and manage resources such as ambulances, paramedics, and other emergency responders [8].

A person in need calls at the EMS center for quick service as they provide pre-hospital and on-the-spot care to the patient in need. In general, EMS is responsible

H. Gupta (✉) · Zaheeruddin

Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi, Delhi, India
e-mail: guptahina189@gmail.com

for transporting patients who may or may not require immediate medical attention. However, in this work, the authors have focused on handling situations where EMS is focusing on urgent patients. The workflow of EMS involves: (1) arrival of emergency call; (2) screening of call; (3) dispatching of ambulance; (4) on-the-spot treatment; (5) transporting the patient to a medical center [11]. The last step is situation-based and is followed in case the patient needs it [1]. After the patient is transferred to the medical center, the ambulance reaches back to the base station. The ambulance waits at the base station until assigned a new task. The question is how the strategies and resources should be deployed to provide society with the best service [10].

As the EMS deals with time-sensitive cases, the response time of the ambulance has been used as an attribute in this work. Response time is the time an ambulance takes to reach the demand location after the request was initiated from someone in need. The work therefore aims at minimizing the average response time of the EMS. To obtain an optimal allocation plan, modified Whale Optimization Algorithm (mWOA) has been used.

The flow of the paper considers the literature review in Sect. 2 followed by the framework and methodology in Sect. 3. Section 4 covers the results and discussion. Finally, Sect. 5 states the conclusion of the paper.

2 Literature Review

Since the mid-1960s, EMS has been a fascinating topic of research in operations' research. Torgeas et al. was the first to study about ambulance allocation. The authors formulated Location Set Covering Problem (LSCP) to ascertain that sufficient coverage to demand zones can be provided using few vehicles [17].

The concept of coverage was used to ensure that each demand zone is covered by at least one vehicle within a certain distance or time. In practice, a sufficient count of vehicles ' a ' are needed to attain full coverage of any area. Considering administrative perspective, the main motive is finding a way in which the resources can be optimally utilized (in this case, ambulance fleet). On the basis of this motive, Maximum Location Covering Problem (MCLP) was formulated by Church and Reville [3]. It considered the fleet size and focused on maximizing the demand covered by the fleet. The problem considered in MCLP was extended to deal with the emergency room sharing problem in Austin, Texas, by Eaton et al. [4]. To improve the results, Galvao and Reville proposed Lagrangian heuristics [5]. To expand the scope of research, MCLP dealing with two types of vehicles was introduced [15]. Despite being very simple in design, an important role was played by single-coverage models.

Many variations to this model resulted in valuable contributions. The deterministic coverage models use a key assumption that when a request call is initiated, a vehicle is always available to serve the patient. However, practically this is not always possible because if the time between two consecutive calls is too short, the vehicle may not be able to serve the second request as it is busy serving the first request.

Many algorithms like Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Hybrid PSOGA, and Shuffled Frog Leaping Algorithm (SFLA) have been used to improve the performance of ambulances. Mirjalili and Lewis, Australian researchers used the hunting mechanism of humpback whales to propose Whale Optimization Algorithm (WOA) in 2016 [13]. The algorithm imitates the shrinking encircling, updating the position spirally, and random hunting mechanisms of humpback whale pods. These days, WOA is being used to solve many problems related to task scheduling, resource allocation, clustering, network, and other fields. A differential chaotic Whale Optimization Algorithm was proposed by Liu and Zhang [12]. With differential chaotic whale optimization, the authors were able to improve the accuracy and effectiveness in detecting network faults in IEEE-33 nodes. Yan et al. improved WOA to find solution for the autonomous underwater vehicles that used the technique of 3D path planning [18]. A scheduling method for multi-stage mine use was proposed by Bo et al. using a similar algorithm [2]. The field of crashworthiness was used by Qian et al. to testify the multi-objective Whale Optimization Algorithm's competence [14]. Chaotic oppositional WOA was applied along with firefly search in the field of medical diagnostics by Tair et al. [16]. An energy minimization Whale Optimization Algorithm was proposed by Zhang et al. for scheduling cloud workflow [20]. In terms of locating facilities and supply allocation, Yan et al. used improved WOA to solve the multi-objective optimization problem of allocating water resources in Handan, China [19]. In another work, WOA was used to solve the problem of a supply chain network [6]. Later, PSO was combined with WOA to find feasible solution to the problem of site selection for municipal solid waste incineration plants [9].

3 Methodology and Implementation

The simulation optimization framework used in the work is shown in Fig. 1. It comprises an allotment component, optimization component, and Google Distance Matrix Application Programming Interface (API). In the figure, a two-dimensional variable ' r ' denotes the location from where the request initiates. It states the coordinates of the latitude and the longitude of the demand location. The details about ' r ' are shared by allotment component to Google Distance Matrix API. The state values for the ambulances and the base station are determined by the optimization algorithm. This works as the input for the framework as it refers to the data related to the *id* of the base station and the ambulance count available at the base station. Using the Google Distance Matrix API, the coordinate information ' r ' is used to evaluate the time taken to travel between the base station and the requested location. The API returns a matrix showcasing the time taken to travel to the requested location from each base station. This data generates a list ranking the base station based on their travel time to the requested location in ascending order. When a request arrives, the base station ranked first in the list, and having an availability of an ambulance is selected to dispatch an ambulance to serve the demand location. An ambulance's

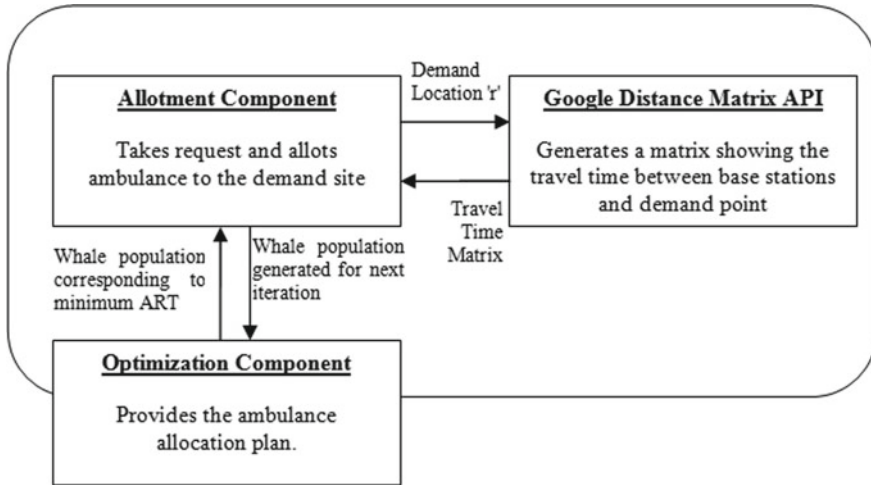


Fig. 1 Framework

time to arrive at the demand site after the request is initiated is termed response time. In such a case, the ambulance’s status is altered from ‘idle’ to ‘busy’. If the ambulance ranked first does not have an idle ambulance, then the base station next in the list is selected, and the process is repeated until the demand location is served. On the other hand, if none of the ambulances is available, the request is placed in the queue and served whenever an idle ambulance is available. In such a case, the response time is evaluated as the summation of waiting and travel times.

The average response time is evaluated when all the requests for a day have been served by using Eq. (1).

$$\text{Average response time} = \frac{\text{total time ambulances take to reach the requested locations}}{\text{total number of request calls registered at the EMS center}} \tag{1}$$

3.1 Modified Whale Optimization Algorithm (mWOA)

Recent research has given rise to many new algorithms. WOA is one such technique that has gained and shown promising results in solving complex optimization problems. WOA uses a search process designed imitating the bubble-net feeding approach of humpback whales that can be mathematically modeled. In the bubble-net feeding strategy, the whales blow bubbles while diving below a shoal of prey. After this step, the whale moves toward the water’s surface in a circular trajectory. Using this step, a circular path of bubbles surrounds the shoal of prey and the whales

move toward the surface [7]. As per the basic terminology, the current best candidate solution (global best) represents target prey, and all other whales represent candidate solutions in WOA. However, to make this work easy to understand, the authors have used the terms current best solution and candidate solution in the upcoming part of this research paper. Similar to other swarm intelligence metaheuristics, local search (exploitation) and global search (exploration) phases are simultaneously used in the search process of WOA. The humpback whale’s spiral bubble-net attack strategy portrays the exploitation process, while the pseudo-random search for prey defines the exploration process.

For any optimization algorithm, there is a need to balance the exploitation and exploration process so that it is possible to obtain an accurate global optimal value. The authors modified WOA by incorporating the concepts of chaotic inertial weight and nonlinear convergence factor.

- (a) **Nonlinear Convergence Factor:** The nonlinear convergence factor is used in mWOA to adjust the local exploitation and global search capabilities. The formula for nonlinear convergence factor is shown in Eq. (2).

$$i_t = 2 - (i_{\text{initial}} - i_{\text{end}}) * \left[\frac{\left(e^{\frac{t}{\text{Maxiter}}} \right) - 1}{e - 1} \right]^u, \tag{2}$$

where the initial and termination values of ‘*I*’ are denoted by i_{initial} and i_{end} . ‘*t*’ denotes the current number of iterations, Maxiter denotes the maximum number of iterations, and u is a nonlinear adjustment coefficient.

- (b) **Chaotic Inertial Weight:** Although the performance of algorithm is improved by the nonlinear convergence, it cannot effectively balance the exploitation and the exploration capabilities. To introduce chaos characteristics, WOA is modified by incorporating chaos strategy for the inertia weight. This step improves exploration capability without hampering the exploitation ability of the process. Therefore, the chaotic inertial weight strategy is used in this work to generate sequences logically.

$$\omega(k + 1) = 4 * \omega(k) * (1 - \omega(k)), \quad k = 1, 2, \dots n. \tag{3}$$

To update the spiral position and the current individual position, the following formulas can be used:

$$\vec{Y}(t + 1) = \begin{cases} \omega \cdot \vec{Y} * (t) - \vec{I} \cdot \vec{J} & \text{if } p < 0.5 \tag{4} \\ \vec{J} \cdot e^{bl} \cdot \cos(2\pi l) + \omega \cdot \vec{Y} * (t) & \text{if } p \geq 0.5 \tag{5} \end{cases}.$$

As chaos optimization is associated with ergodic and random properties, it ensures that when any particle having a high local searching ability falls in the local optimum, it jumps out quickly.

3.2 Study Area

To handle the rising demand rate in Delhi, the government has formed Centralized Accident and Trauma Services (CATS) organization, to provide EMS. CATS serves Delhi and explicitly handles patients who are accident and trauma victims, with an average response time of 13 min. Southern Delhi covers an area of 857 square kilometers and comprises four districts South Delhi, South West Delhi, South East Delhi, and New Delhi, as shown in Fig. 2. CATS has allocated 50 ambulances at the 11 base stations numbered from BS1 to BS11 in Southern Delhi, as shown in Fig. 3. Highest numbers of accident cases are reported in the southern portion of Delhi, so the authors selected to work upon this area account for the highest number of cases that are needed to be dealt with by CATS. Thus, this work area of Southern Delhi has been used to carry out all the experiments and obtain an optimized allocation plan. To get relevant results, the authors used tessellation of the area consisting of 230 blocks each covering an area of 4 square kilometers, as shown in Fig. 4.

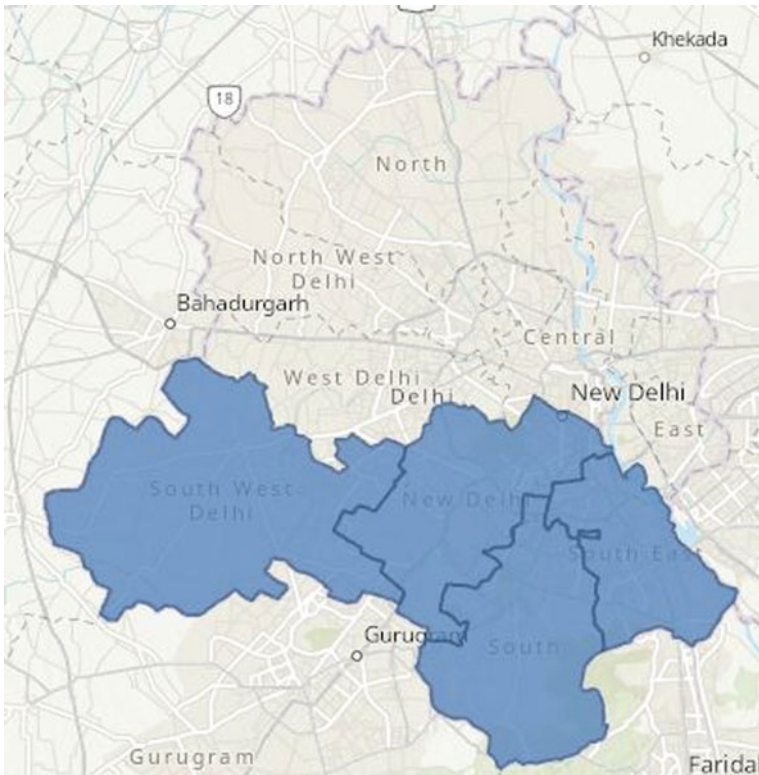


Fig. 2 Southern portion of Delhi

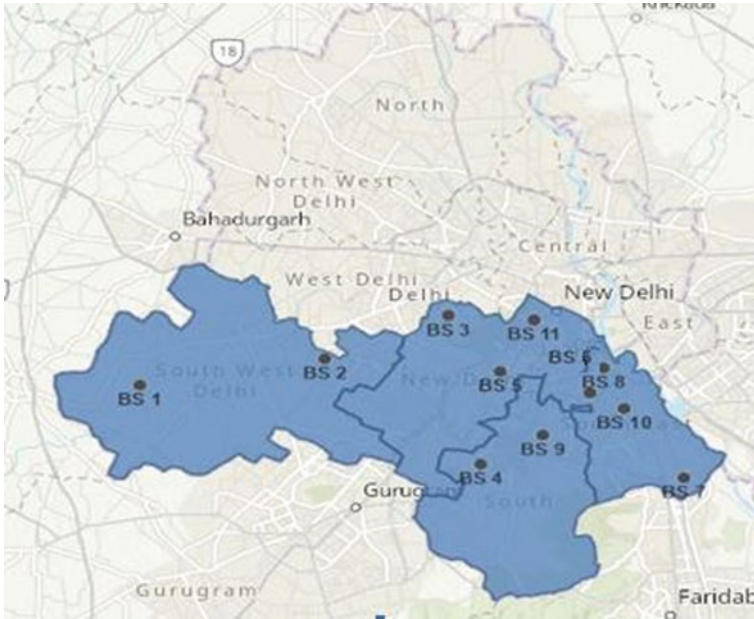


Fig. 3 Base stations of Southern Delhi

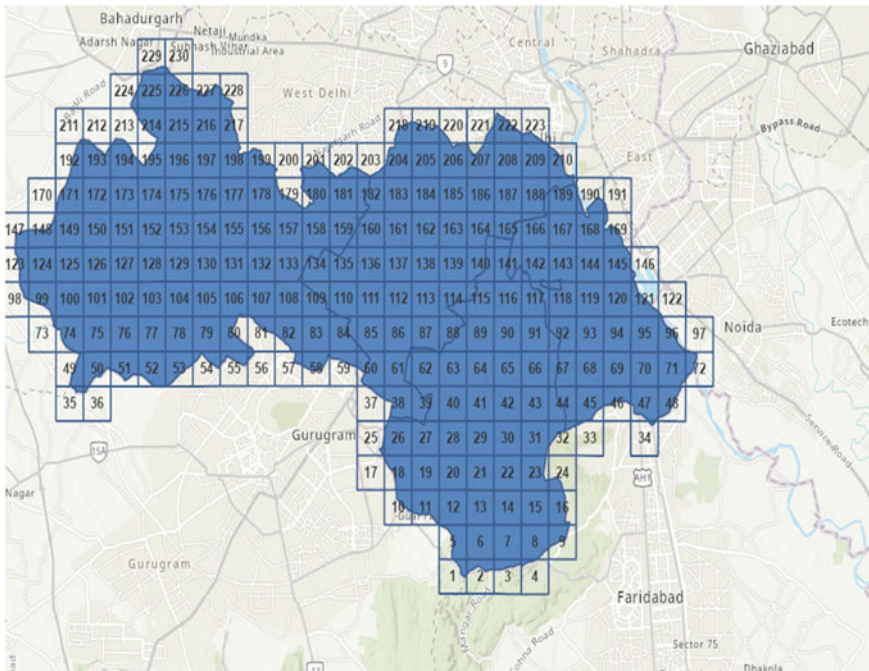


Fig. 4 Tessellations of Southern Delhi

In each framework, random requests are generated. The generation of requests is done so that at least one request is generated from each of the 230 blocks. To conduct experiments, the population size and iteration count were set to 100, and 1000 respectively. The simulation experiment was conducted 20 times with 1500 requests generated. The data was recorded for each of the 20 runs. Different algorithms have been compared using the recorded data, and the performances have been evaluated. The proposed simulation algorithm was implemented on MATLABR2015a.

4 Results and Discussion

The simulation optimization framework was executed twenty times. Different algorithms like GA, PSO, HPSOGA, SFLA, and mWOA were used as the optimization components. The comparisons of all algorithms have been done taking the rate of convergence, fitness function value, and constancy repeatability into consideration.

- (a) **Fitness Function Value:** This comparison considers the final objective function result obtained by each algorithm. Among all the values, PSO achieves the maximum value of 11.68 min, followed by GA, HPSOGA, and SFLA with fitness function values of 11.504, 11.495, and 11.412 min, respectively. The least fitness function value (minimum response time) is attained as 11.1 min by the mWOA fulfilling the motive of this work. The graph shown in Fig. 5 depicts that the least value for fitness function is provided by mWOA.
- (b) **Convergence Rate:** This value states the iteration number at which, and henceforth the algorithm starts providing the same result. The graph shown in Fig. 6 depicts the convergence rate of each algorithm. The plot has considered the best run and iteration of every algorithm. The results demonstrate that mWOA

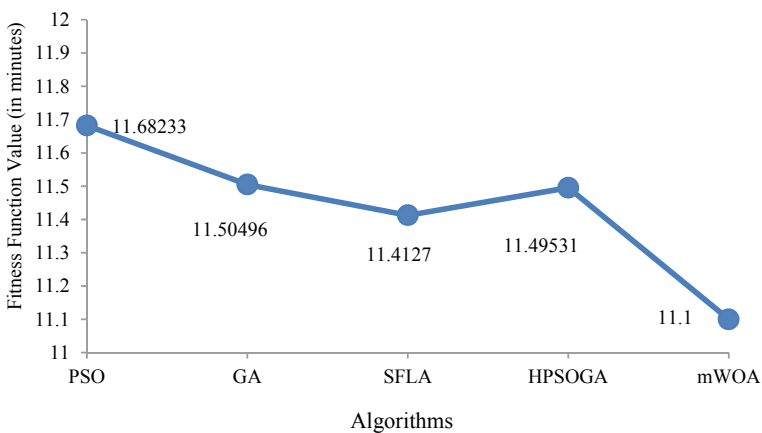


Fig. 5 Fitness function value graph

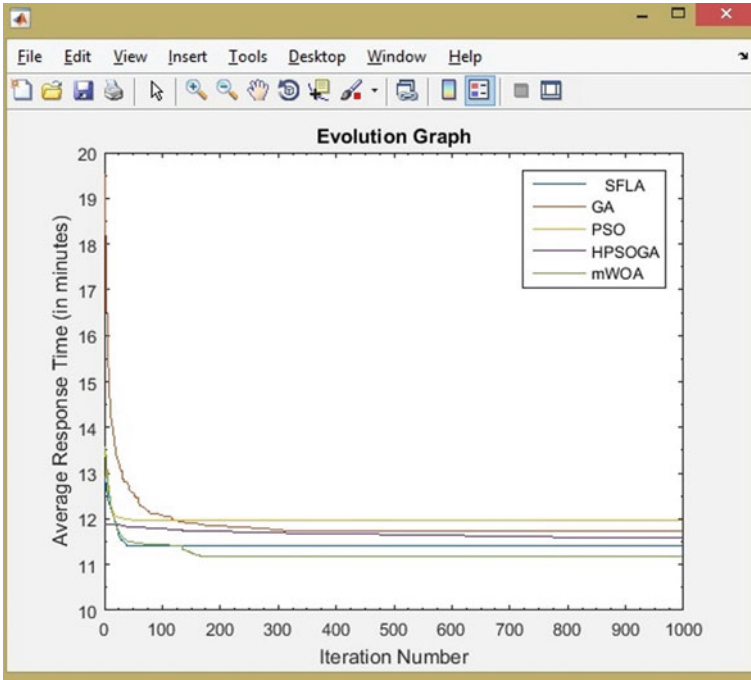


Fig. 6 Convergence rate (evolution) graph

converges at a moderate rate compared to PSO and SFLA, which converge quickly, and GA and HPSOGA, which converge slowly. The plan of ambulance allocation obtained by each algorithm is shown in Fig. 7.

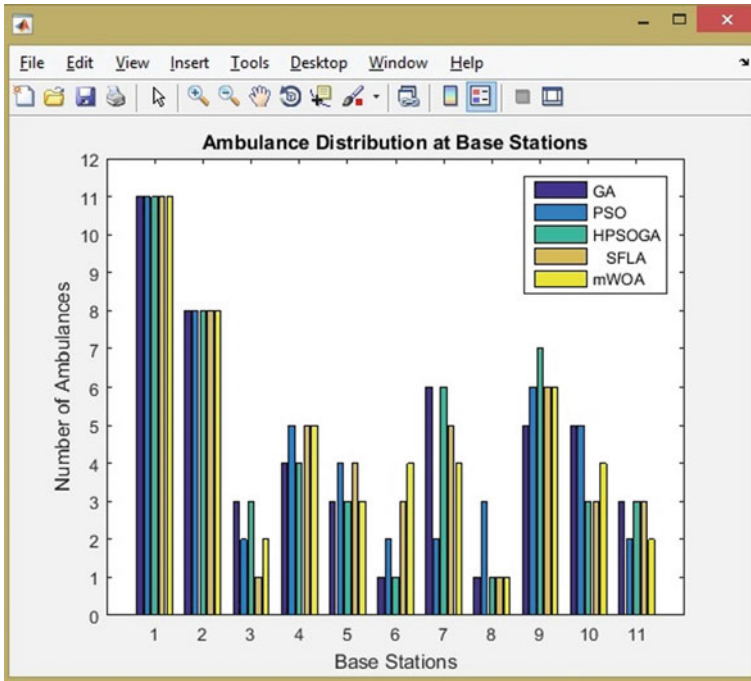


Fig. 7 Allocation plan of ambulances obtained by each algorithm

5 Conclusion

To improve the service efficiency of EMS, a new algorithm mWOA was introduced in this paper. The work aimed at strategically allocating ambulances. The area of Southern Delhi was selected as the study area. The problem focused on finding an allocation plan for 50 ambulances to be distributed among 11 base stations to reduce the average response time of CATS ambulance service. The basic WOA was modified using chaotic inertial weight and nonlinear convergence factors. Simulation experiments were performed to find and compare the results of different algorithms. Results justify that the allocation plan provided by mWOA will provide ambulance service with an average time of 11.1 min to respond to any demand spot in Southern Delhi. This value is 14.6% less than the current average response time of 13 min.

The authors focus on extending the problem and consider different types of vehicles available for handling patients in the future. The work can also be extended by changing the problem form single-objective problem to multi-objective problem.

References

1. Bijani M, Abedi S, Karimi S, Tehranineshat B (2021) Major challenges and barriers in clinical decision-making as perceived by emergency medical services personnel: a qualitative content analysis. *BMC Emerg Med* 21(1):1–12
2. Bo L, Li Z, Liu Y, Yue Y, Zhang Z, Wang Y (2022) Research on multi-level scheduling of mine water reuse based on improved whale optimization algorithm. *Sensors* 22(14):5164
3. Church R, ReVelle C (1972) The maximal covering location problem. *Papers Reg Sci Assoc* 32:101–118
4. Eaton DJ, Daskin MS, Simmons D, Bulloch B, Jansma G (1985) Determining emergency medical service vehicle deployment in Austin, Texas. *Interfaces* 15(1):96–108. <https://doi.org/10.1287/inte.15.1.96>
5. Galvão RD, ReVelle C (1996) A Lagrangean heuristic for the maximal covering location problem. *Eur J Oper Res* 88(1):114–123
6. Ghahremani-Nahra J, Kian R, Sabet E (2019) A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm. *Expert Syst Appl* 116:454–471
7. Goldbogen JA, Friedlaender AS, Calambokidis J, McKenna MF, Simon M, Nowacek DP (2013) Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *Bioscience* 63(2):90–100
8. Jaffe E, Sonkin R, Strugo R, Zerath E (2021) Evolution of emergency medical calls during a pandemic—an emergency medical service during the COVID-19 outbreak. *Am J Emerg Med* 43:260–266
9. Jiang S, Li Z, Gao C (2022) Study on site selection of municipal solid waste incineration plant based on swarm optimization algorithm. *Waste Manage Res* 40(2):205–217
10. Karpova Y, Villa F, Vallada E, Vecina MÁ (2023) Heuristic algorithms based on the isochron analysis for dynamic relocation of medical emergency vehicles. *Expert Syst Appl* 212:118773
11. Khennak I, Drias H, Khelfa C, Drias Y, Bourouhou NH, Zafoune I (2023) Multi-objective Harris hawks optimization for optimal emergency vehicle dispatching during a pandemic. In: 14th international conference on soft computing and pattern recognition (SoCPaR), pp 852–861
12. Liu L, Zhang R (2022) Multistrategy improved whale optimization algorithm and its application. *Comput Intell Neurosci* 2022:1–16
13. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
14. Qian L, Yu L, Huang Y, Jiang P, Gu X (2023) Improved whale optimization algorithm and its application in vehicle structural crashworthiness. *Int J Crashworth* 28(2):202–216
15. Schilling D, Elzinga DJ, Cohon J, Church R, ReVelle C (1979) The team/fleet models for simultaneous facility and equipment siting. *Transp Sci* 13(2):163–175
16. Tair M, Bacanin N, Zivkovic M, Venkatachalam K (2022) A chaotic oppositional whale optimisation algorithm with firefly search for medical diagnostics. *Comput Mater Continua* 72:959–982
17. Toregas C, Swain R, ReVelle C, Bergman L (1971) The location of emergency service facilities. *Oper Res* 19(6):1363–1373
18. Yan Z, Zhang J, Zeng J, Tang J (2022) Three-dimensional path planning for autonomous underwater vehicles based on a whale optimization algorithm. *Ocean Eng* 250:111070
19. Yan Z, Sha J, Liu B, Tian W, Lu J (2018) An ameliorative whale optimization algorithm for multi-objective optimal allocation of water resources in Handan, China. *Water* 10(1):87
20. Zhang L, Wang L, Xiao M, Wen Z, Peng C (2022) EM_WOA: A budget-constrained energy consumption optimization approach for workflow scheduling in clouds. *Peer-to-Peer Netw Appl* 15(2):973–987

An Integrated Model for Acceptance of QR Code Mobile Payment: A Comparative Study Between Male and Female



Priyanka Yadav, Anshul Jain, and Khyati Kochhar

Abstract To understand the functions and features of mobile payment apps and QR code technology, as well as how end users will respond to these modes, it is essential to study their behavior. Thus, this study pinpoints the variables affecting consumers' behavioral intentions toward mobile-based QR code payment. The study's conceptual framework is based on the TRA, TPB, and TAM models. The data for the study have been gathered using a structured questionnaire, and the responses of 440 respondents are considered suitable for further analysis. The confirmatory factor analysis (CFA) has been used for measuring the reliability and validity of the model, and analysis has been done using the structural equation modeling (SEM) technique in AMOS 24. The factors considered in this study are PU, PEOU, CO, SN, and TR. It is concluded that PU, PEOU, CO, SN, and TR significantly influence the users' attitudes, and attitude significantly influences the behavioral intention of the users. Moreover, it has been analyzed that male and female users' attitudes differ in the case of TR and SN. In addition, some future research directions are suggested, addressing the intention to use a QR code payment system based on the findings and considering the constraints mentioned above.

Keywords QR code payment · Mobile payment · Behavioral intention · Attitude · TAM

P. Yadav (✉) · A. Jain · K. Kochhar
FMS-WISDOM, Banasthali Vidyapith, Tonk, Rajasthan, India
e-mail: Priyankay042@gmail.com

A. Jain
e-mail: anshuljain0610@gmail.com

K. Kochhar
e-mail: Khyatikochhar@banasthali.in

Abbreviations

PU	Perceived usefulness
PEOU	Perceived ease of use
CO	Compatibility
SN	Subjective norms
TR	Trust
AT	Attitude
BI	Behavioral intention
TAM	Technology acceptance model
TPB	Theory of planned behavior
TRA	Theory of reasoned action

1 Introduction

Innovative gadgets and technological advancements have substantially altered our ways of making payments [7]. Like, the innovation of the 5G technology has led to the disruption of the conventional cash-centric payment system and the advancement of digital payment solutions, like mobile banking and e-wallets [53]. The rise of mobile technology has made mobile phone payments an essential part of human lives [26, 45]. The number of mobile-based payment users is increasing at a healthy rate all over the globe because of the benefits that mobile payment technology provides [38]. At the end of 2023, approximately 1.31 billion individuals will use mobile payment transactions worldwide, and in 2019 it is 950 million [16]. Similar to other developed countries, India's mobile phone payments' sector has increased due to the growth of a Fintech-based environment [28]. The market size of mobile payments has touched US\$ 510.1 billion in 2022, and it is expected to increase to US\$ 2063.8 billion in the next 5 years [20]. In the context of current trends, the majority of technology firms are concentrating on expanding the number of services that are offered such as mobile-based QR code payments [33]. The introduction of the QR code method of payment devices has propelled people to buy and pay for products and services using mobile banking applications [54]. Moreover, customers also prefer paying using mobile-based QR codes rather than cash and bank cards [44]. However, this preference, whether it is for internet, mobile phones, or mobile payments, is not similar among men and women in India. As per the records, mobile internet usage has risen from 45 to 51% for men from 2020 to 2021, while it remains stagnant at 30% for women [29]. In addition, the "India Inequality Report 2022" shows that only 31% of women own mobile phones, as compared to 61% of men in India [58]. Not only this, but statistics also show that nearly 48% of women in India still prefer making payments via cash [58]. The fundamental reasons behind this are the lack of facilities and education, affordability issues, and society's stereotypes that women should not go online [15, 62]. In contract, the availability of mobile-based QR code

payment services has encouraged both men and women to use mobile banking apps to purchase and make payments for products and services [27, 54]. Moreover, to induce people to use it, many companies are using it to reduce the gap between online and offline payments. The expansion of online payment and UPI networks has led to the growth of QR code adoption in India [42]. As per the Indian statistics, more than 9 million merchants accept payments made with QR codes, and the UPI interface recorded that nearly 2.8 billion transactions were done using QR codes in 2021 (*QR Code*, n.d.). Moreover, 40% of the population in India uses QR codes, and 1,101,723 scans were recorded in 2022 [48]. The number of QR code scans in India is increasing at a steady rate. However, because of the digital gender divide, the growth in QR code payments is also not balanced between men and women. Moreover, during the literature analysis, it is analyzed that less research has been done on the comparative analysis between men and women in respect of QR code mobile-based payment, especially in India [64, 65]. Therefore, the authors identify the factors influencing the behavior of users toward QR code payments in India, and a comparative analysis has been done between the male and female users. The growing relevance of QR codes for making payments among users makes it essential to understand their behavior patterns, as numerous variables affect users' behavioral intention to use QR codes for making payments. Based on these prepositions, following research questions have been addressed in this study:

1. Which factors significantly influence the intentions of users toward the mobile-based QR code mobile payment?
2. What is the impact of these factors on the behavioral intentions of consumers?
3. Which factors have a differential influence on the behavioral intentions of male and female users?

2 Literature Review

According to Mookerjee et al. [38], the acceptance rate of mobile-based payment systems among users is lower than that of technological progress. However, due to the quick growth of contemporary mobile communication technology, a new age of transaction services like QR code and face recognition payments has begun [11, 70]. Moreover, these QR code-based payments have become much more popular because of the COVID-19 pandemic due to which the consumption behavior of people has changed significantly [4, 38, 66, 70]. The majority of the literature that is currently available on QR codes focuses on the development of the code and its applications [25, 51], algorithm development and implementation [6], and usage of apps [47]. However, a few studies show the influence of various factors on the usage and adoption of mobile-based QR code payment systems [4, 27, 54, 67]. A study by Kongarchapatara [26] and Liébana-Cabanillas et al. [33] shows two significant variables of the TAM model, PU and PEOU, which substantially affect user attitude toward the acceptance of QR code payment services. In contrast, studies by Chang et al. [12] and Wei et al. [65] have shown that subjective norms from the TPB model

significantly influence users' behavior. Likewise, many researchers [18, 33, 54] assert that factors like compatibility and trust can also positively affect users' behavior toward QR code payments. However, there is a dearth of studies on the comparative analysis between males and females adopting QR code payment, especially in India [64, 65]. Therefore, through this study, the authors identify the variables affecting customers' behavior toward QR code payments in India, and a comparative analysis has been done between the male and female users.

3 Theoretical Context

The TAM model, developed by David Davis in 1989, investigates the variables influencing people's actions to embrace modern technology [14]. As per the model, PU and PEOU are the two variables that influence the attitude of an individual toward using a particular technology and, thus, determine their technology usage intention [14]. It is widely used in examining users' intentions to adopt mobile-based QR code payment technology [4, 12, 26]. However, it was determined in a few studies that the initial TAM model alone has failed to determine the influence of revolutionary technologies on the user's behavior and intention [31]. Thus, Venkatesh and Davis expanded the past TAM model to incorporate two further variables, i.e., social influence and cognitive instruments that influence usage attitude [63]. Moreover, two broad theories of human behavior, known as the TRA and TPB [2], served as the basis for the TAM model. Past researches on the usage and acceptance of mobile technology and mobile-based QR code payments have used these two theories [4, 33]. As a result, it is anticipated that including the variable (subjective norms) from TPB in TAM will give researchers an in-depth understanding of how consumers intend to accept mobile technology [4]. In addition to this, various studies have also highlighted the significance of various other factors, such as trust [13, 17, 27] and compatibility [1, 8] in influencing the customer's attitude and usage intention toward mobile technology and QR code payment system. Trust is a significant variable because it impacts the attitude of the individuals using technology [13, 66]. However, CO as a variable of user behavior intention has evolved from the "DOI" theory developed by Roger [50]. However, for this study, the authors want to know the extent to which mobile-based QR code payments are compatible with the values, beliefs, and experiences of the users. Hence, in this study, compatibility as a significant factor that influences user behavior intention has been studied. It has been analyzed that the users' attitude and intent to use QR code payment does not only depend on its performance, but it is influenced by various other factors like PU, PEOU, SN, CO, and trust [4, 12, 27]. Moreover, Wei et al. [65] identified that the attitude of male users significantly differs from that of female users, which makes it crucial to study their attitudes and behavior separately. Therefore, the current study adopted a comparative analysis approach to study the difference in attitude and behavior of male and female users.

4 Conceptual Framework

4.1 *Perceived Usefulness and Attitude*

PU is understood as “the extent to which one perceives that employing a certain system would improve one’s performance at work” [14]. Usefulness in terms of mobile technology is the amount to which technology can significantly benefit people’s capacity for engaging in transactions [17]. Intrinsically, if the technology offers benefits like efficiency, safety, or comfort, people will use it. The usage of technology in this study focuses specifically on the use of QR codes in mobile payments. The usage of QR code mobile payment boosts people’s transactional efficiency [21]. Based on the various researches, it has been determined that the PU is the important element that modifies the attitude of the users to use the QR code payment [17, 33, 39].

H1 PU has a significant impact on users’ attitude to adopt QR code payment.

4.2 *Perceived Ease of Use and Attitude*

The PEOU is understood as “the level of ease with which consumers use technology” [63]. The term “ease-of-use” describes how someone feels about how simple or straightforward it is to utilize a given technology [56]. PEOU in terms of mobile technology is identified as the extent of user perceptions of understanding and using mobile technology or services as being effortless. More precisely, customers favorably evaluate and value these services more when they are user-friendly when it comes to mobile services like mobile payment and mobile learning [5, 41]. Using a QR code to make a payment depends on whether the customer finds the process simple to follow and convenient [12]. The user’s interactions with the QR code payment system should also be simple to understand. Based upon the various researches, it has been demonstrated that customers’ attitude toward QR code mobile payment is significantly impacted by perceived ease [5, 12, 44].

H2 PEOU has a significant impact on users’ attitude to adopt QR code payment.

4.3 *Compatibility and Attitude*

Compatibility (CO) measures how much customers perceive the technology is compatible with their beliefs, behaviors, and lifestyles [39]. And as per the Rogers [49], it refers to how well knowledge is seen as aligning with the values, needs, and

prior experiences of possible users. Technology adoption models believe compatibility to be a key component, and it is accepted that a person's values being incompatible with an invention will prevent that person from adopting it [49]. As per the Nur and Gosal [40], CO is the one of the elements of the technology acceptance behavior. According to Tornatzky and Klein [60], the acceptability of a specific technology, especially if it is new, may depend on how compatible users perceive it to be. Based on past research, it is concluded that CO favorably influences the attitude of the users to use QR code payment services [5, 8, 40].

H3 CO has a significant impact on users' attitude to adopt QR code payment.

4.4 Subjective Norms and Attitude

Social norms and social influence are the components of the subjective norm [68]. Social influence refers to following family, friends, and classmates' viewpoints, whereas societal norms refer to conforming to the greater society fashion (big sphere of influence) [8, 68]. SI, also known as the SN, refers to a person's perception of pressure from their selected societal referents to engage in the conduct [12, 67]. The adoption and growth of QR payment methods are still in their infancy in India. Moreover, third parties have a substantial effect on the handlers' attitude toward the payment system [33, 67]. The measurement of subjective norms, which are the extent to which user has faith in that they should use a specific system or do a particular action, allows for the inclusion of social context in this model [33, 63]. Various past types of research have shown the importance of subjective norms on the attitude of the users [8, 33, 40].

H4 SN has a significant impact on users' attitude to adopt QR code payment.

4.5 Trust and Attitude

Trust is considered as users' propensity to anticipate favorable performance from technology in the future and a personal conviction that the service provider will uphold their commitment [69]. In respect of electronic financial transactions, TR is the subjective opinion that a party will uphold their responsibilities and play an imperative part when users are put under higher risk owing to the ambiguity and lack of control they feel in their surroundings [43]. TR is essential for deciding on future course of action between parties [17, 43]. Users are unable to enjoy a compelling experience if they do not have confidence in the companies offering mobile payment services. This research has consistently shown that users' attitudes toward using mobile payments are significantly affected by their level of trust. People are more

inclined to adopt mobile payment when they have greater trust in the stakeholders as they are more likely to think that there are benefits outweighing hazards [34, 43, 69].

H5 TR has a significant impact on users' attitude to adopt QR code payment.

4.6 Attitude and Behavioral Intention

According to Ajzen [2] and Premkumar et al. [46], attitude is the positive or negative emotions that humans express by their behavior. This implies that attitudes alter as individuals get more experience. Several theoretical models (TPB, TRA, and TAM) and studies [2, 17] suggest that attitude is a key indicator to participate in a specific behavior. The following hypotheses have been developed because of numerous investigations [35, 37] identifying a positive association between attitude and intention to behave.

H6 Attitude has a significant impact on users' behavioral intention to adopt QR code payment.

5 Research Methodology

5.1 Instrument Development

The survey components were adapted from previously published scales to suit the aspects of factors that affect the attitude and behavior intention to use the QR code. A draft questionnaire was sent to a pilot group of experts and users to collect accurate answers and minimize the possibility of personal bias. The pilot group had a positive response and made suggestions for improving the items' relevance. Their recommendations were incorporated into the questionnaire. Additionally, validating the sample size needed to create a strong structural model is crucial [19]. Various theories and methodologies recommend a minimum sample size based on the several of variables being investigated. According to Thompson et al. [57], at least ten times as many indicators must be included in the sample as there are indicators used to quantify a single latent variable. The minimal sample size in this study is $N = 40$, and trust (TR) has the highest number of indicators, four. According to Kang [22], the suggested minimum sample should be dependent upon the analysis's power. Using G. Power to compute the sample size with a effect size of 0.15. A minimum sample size of 146 responders is advised. We have collected the data above the minimum sample size criteria.

5.2 Data Collection and Sample Description

The data for the study were collected from mid-January 2023 to April 2023. Face-to-face surveying and online surveying were utilized to gather data for the research. Each questionnaire-gathering procedure has its benefits and drawbacks [55]. For the study, we followed a judgmental or purposive sampling technique followed by systematic random sampling. Judgmental sampling is utilized when the researcher establishes specific standards or metrics for the sample or respondents of the study [59]. So, we have collected the data from the five renowned universities of northern states. The reason behind the specific focus on North India was due to the fact that Northern India is lagging behind in digital payment usage in comparison to Southern India [10]. Out of the total circulated questionnaire, 480 (80%) were obtained, 440 (91.66%) were deemed suitable after excluding incomplete responses, and respondents with no knowledge of QR code mobile payments were also excluded. Of the collected samples, 48% were male respondents and 52% were female respondents; 3% of the respondents were between 18 and 27 of age, followed by 42.4% of respondents between 28 and 27 age, 45.5% between 38 and 47 age group, and 9.1% were belonged to above 48.

6 Data Analysis

6.1 Common Method Bias

At first, construct items were randomized to reduce the error probability [23]. Secondly, the most commonly applied Harman's one-factor test in SPSS resulted in a single component explaining about 45.02% of the variance, showing no issue of method bias [3].

6.2 Measurement Model

Validating the evaluation of the outer model (measurement model) (structural model) is crucial before testing the hypotheses of the inner model. Analyzing the measuring model is necessary to confirm that the measurements are accurate and appropriately reflect the comprehended theoretical components. As part of the evaluation process, the validity and reliability of the measurement model (Cronbach's alpha and Composite Reliability (CR)) are also examined. To examine the reliability, the value of Cronbach's alpha is more than 0.7 which is good indicator [24]. As indicated in Table 1, model validity metrics such as CR, average variance expected (AVE), and discriminant validity [3, 52] were computed to establish the measurement model's

Table 1 Measurement model

Constructs	Items	Loadings	Cronbach's alpha	CR	AVE	Discriminant validity
Behavioral intention	BI1	0.898	0.966	0.968	0.883	0.940
	BI2	0.954				
	BI3	0.924				
	BI4	0.980				
Social norms	SN1	0.917	0.955	0.956	0.878	0.937
	SN2	0.942				
	SN3	0.951				
Compatibility	CO1	0.947	0.946	0.947	0.855	0.925
	CO2	0.929				
	CO3	0.828				
Perceived usefulness	US1	0.886	0.947	0.948	0.819	0.905
	US2	0.918				
	US3	0.908				
	US4	0.908				
Trust	TR1	0.904	0.956	0.954	0.838	0.915
	TR2	0.860				
	TR3	0.959				
	TR4	0.935				
Perceived ease of usefulness	PEOU1	0.911	0.965	0.965	0.875	0.935
	PEOU2	0.933				
	PEOU3	0.938				
	PEOU4	0.958				
Attitude	AT1	0.929	0.932	0.934	0.826	0.909
	AT2	0.939				
	AT3	0.856				

reliability and validity. Convergent validity is indicated by the fact that the AVE of all the constructs is greater than 0.5 and the value of the CR is greater than 0.7 [24].

6.3 Second Step: Assessment of Structural Model (SM)

The relationship was examined using an AMOS-generated structural equation model. Before analyzing the similarities and differences between the models, it was confirmed that the overall goodness of fit was sufficient because the values of the GFI indicators fall within the ranges suggested by the literature [30]. The overall measurement model fit indices of the resultant model showed acceptable-to-excellent model

fit ($\chi^2 = 418.850, df = 258, p = 0.000$, with root mean square error of approximation = 0.063, comparative fit index = 0.943, goodness of fit index = 0.993, Tucker–Lewis Index = 0.935, normed fit index = 0.980, $\chi^2/df = 1.755$), so it is concluded that the measurement model is compatible with the data [36] and it is mentioned in the below tables (Tables 2 and 3).

Generally, results of assessing hypotheses show a favorable association between AT and BI for using QR code mobile payment. Figure 1 depicts the relation between the variables affecting QR code payments and users’ behavioral intentions. R^2 is 0.75, which means that 75% of the variability is explained by the model. Furthermore, the standard estimate between the attitude driving the QR code mobile payment adoption and BI is 0.86.

6.4 Multi-group Analysis

We bifurcated the whole sample into two sub-samples (male and female). We employed multi-group AMOS analysis to compare the differences among all the

Table 2 Assessment of structural model (SM)

Name of fit indices	Index level of acceptance	Level of acceptance
CMIN/DF	1.755	< 5
<i>Absolute fit indices</i>		
GFI	0.993	> 0.90
RMSEA	0.063	< 0.80
<i>Incremental fit indices</i>		
NFI	0.980	> 0.90
CFI	0.944	> 0.90
TLI	0.935	> 0.90

CFI comparative fit index, *DF* degree of freedom, *GFI* goodness-of-fit index, *NFI* normed fit index, *TLI* Tucker–Lewis Index, *RMSEA* root mean square error of approximation

Table 3 Summary of hypothesis testing

Hypothesis	Relationship	β value	<i>T</i> statistics	<i>P</i> values	Decision
H1	PU → AT	0.253	2.199	0.028	Significant
H2	PEOU → AT	0.189	4.428	0.00	Significant
H3	CO → AT	0.492	4.084	0.000	Significant
H4	SN → AT	0.247	2.767	0.006	Significant
H5	TR → AT	0.245	2.829	0.005	Significant
H6	AT → BI	0.857	10.911	0.00	Significant

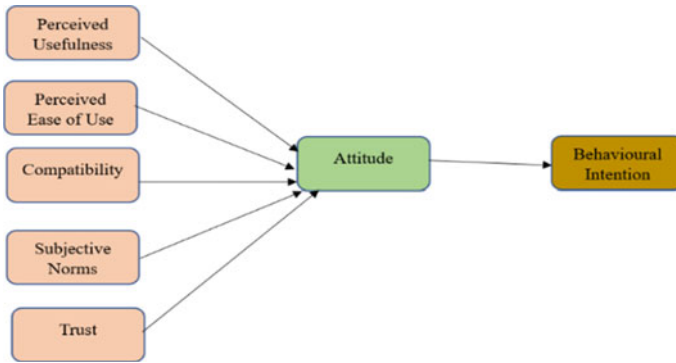


Fig. 1 Research framework. *Source* Researcher’s own construction

path relationships in the two samples. The analysis revealed that the regression coefficients of TS and SN are different between the two groups. Therefore, it was discovered that the path from TR and SN to attitude has shown the difference between the two groups, whereas path from PU, PEOU, CO to attitude has demonstrated the same path coefficient. It means that there is no statistically important distinction between male and female customers, indicating that these elements influence the groups of customers in the same way regarding the behavioral intention to use of QR code mobile payment. In male, social norms play a dominant role in adopting QR code mobile payment. In female, TR plays dominate in embracing QR code mobile payment in their daily lives.

7 Result and Discussion

In this study, SEM analysis has been performed to identify the individual differences in PU, PEOU, CO, SN, TR, and AT as the determinants of BI toward mobile-based QR code payment. For evaluating the structural model, the statistical important of structural coefficients is examined. The outcomes of the hypothesis testing and SEM estimates are shown in Fig. 1 and Table 4, respectively. The H1 was supported, as the standard path coefficient between PU and AT is 0.253, with a *t*-value of 2.199. Thus, in accordance with earlier studies [17, 33, 39], the present research also states that the usefulness that the users perceive has a significant influence on the attitude of users toward mobile-based QR code payment. Moreover, the H2 was also accepted as the value of the standard path coefficient with a *t*-value of 4.428 is 0.189. This shows that there is a significant influence of PEOU on the AT of the users [5, 44]. Thus, it can be said that users perceive that it is easy to interact and become skillful with the mobile-based QR code system. Compatibility is considered as a significant driver of customers’ attitudes toward mobile-based QR code payments as they believe that it fits well with their lifestyle, matches how they purchase goods and services, and

Table 4 Multi-group analysis

Factors	Male (β)	Female (β)	<i>P</i> value
PU	0.345	0.005	0.386
PEOU	0.171	0.081	0.891
CO	0.351	0.483	0.483
SN	0.466	0.350	0.049
TR	0.124	0.433	0.05
AT	0.843	0.870	0.579

matches the way they handle their money [8, 33, 40]. Thus, the H3 cannot be rejected as the standard path coefficient between CO and AT is 0.492, with a *t*-value of 4.084. The H4, which states that there is a significant impact of SN on the AT of users, is also accepted, as the standard path coefficient between SN and AT is 0.247, with a *t*-value of 2.767. The findings of Aydin and Burnaz [8], Chang et al. [12], and Liébana-Cabanillas et al. [33] also verify that there is a favorable impact of SN on the attitude of users toward QR code payment. Furthermore, a substantial association between TR and AT of users of QR code payments was also analyzed by the study's data, as their standard path coefficient is 0.245, with a *t*-value of 2.829. Numerous empirical studies on QR code payment also confirm the findings of this relationship [33, 69]. Lastly, the study analyzes a substantial positive association between AT and BI of the users toward QR code payments as the standard path coefficient between them is 0.857 with a *t*-value of 10.911. Davis [14] and many other authors [12, 33] have considered attitude as one of the significant factors in influencing the behavior intention of users.

8 Conclusion

The technological advancements and innovation of new payment system, such as QR code, have substantially altered our ways of making payments. Customers also prefer making payments using mobile-based QR codes rather than cash and bank cards. However, for maintaining a long-term preference for QR codes, it is essential to study the attitude and behavior of customers toward them. Therefore, this study emphasizes on recognizing the elements and analyzing their impact on the users' behavioral intentions toward the mobile-based QR code mobile payment. The study adds to the current arena of knowledge by analyzing the key variables that affect customers' behavioral intentions toward QR code payments. The factors that are studied in this study were PU, PEOU, CO, TR, and SN. Moreover, for developing the proposed research model, multiple theories (TRA, TPB, and TAM) and empirical studies were used. The analysis concluded that factors like PU, PEOU, CO, TR, and SN are significant in influencing the attitude of users, and attitude significantly influences the behavioral intentions of users toward the QR code payment system. The

most substantial impact on users' attitudes comes from CO, out of all the significant factors. Moreover, from comparative analysis, it was discovered that male and female users' attitudes differ only in the case of two factors (trust and subjective norms). In males, subjective norms dominate in adopting QR code mobile payment, while in females, trust plays a dominant role in embracing QR code mobile payment in their daily lives. In terms of the study's academic contribution, it is believed that this is one of the first in the northern part of India to scientifically assess the factors that influence consumer behavior toward QR code payment.

9 Limitations and Future Directions

The authors have put all their efforts to ensure the validity and applicability of the findings of this study. However, certain limitations in the study provide an opportunity for more research. The study has considered only users from the northern part of the country, which may limit the applicability of findings to other parts of the country. In addition, customers' attitudes and behaviors do alter over time as a result of a rising understanding of concerned payment technology. Thus, future researchers can focus on different parts of the country while conducting a long-term investigation to determine variations in the patterns of consumer behavior. Since the study is based on a sample of Indian users, it could be applied to other nations with circumstances similar to India to observe potential cultural differences and possibly establish different levels of technology acceptance. Increasing the sample size of users might also give different results. In addition to this, the impact of demographic variables such as age, income, and education also significantly influences the intentions of the users. Thus, in further studies, the impact of these demographic variables can be studied. Moreover, because of the time and resource limitations, the study covers only a few constructs that influence the attitude and behavior of users. Future research can study the influence of other constructs (such as cultural context) on the users' behavior intentions toward mobile-based QR code payment.

10 Implications

The mobile payment industry is expected to escalate in prominence over the next few decades due to the growing prevalence of smartphones and internet. In order to understand the functions and features of mobile payment apps and QR code technology, as well as how end users will respond to these modes, a model is developed using the already-existing models TRA, TPB, and TAM. The current research has focused that the users' attitude and behavioral intention to use QR code payment not only depends on its performance but is influenced by various other factors like PU, PEOU, SN, CO, and trust. The current study has practically implied two new additional factors, trust (TR) and compatibility (CO), that can influence the users'

attitudes and behavior. From the theoretical point of view, a similar model can be applied to study the attitude and behavior toward other payment methods such as Unified Payment Interface (UPI). From the managerial perspective, it can be said that attitude should be the primary focus when trying to expand the adoption of QR code mobile payments. It is essential to analyze the attitude of users in respect of mobile payments using QR codes. Concerning the outcomes of this study, enterprises must take the initiatives to modify the attitudes of their customers toward using this technology (by giving information and doing advertisements, promotions, etc.). Furthermore, for generating trust among users, a good image of the QR code payment method has to spread by encouraging positive word of mouth. The study also offers practical implications for marketers and service providers. The findings are critical and beneficial for marketers and service providers of QR code payment applications to understand and develop services in accordance with customers' preferences. These activities will improve customers' perception of using QR code-based mobile payment solutions.

References

1. Abebe F, Lessa L (2020) DigitalCommons @ Kennesaw State University factors affecting mobile payment adoption by merchants in Ethiopia. *Afr Conf Inform Syst Technol* 12:0–11. <https://digitalcommons.kennesaw.edu/acist>
2. Ajzen I (1991) The theory of planned behavior. *Organ Behav Human Decis Process* 33(1):52–68. <https://doi.org/10.47985/dcidj.475>
3. Al-okaily M, Lutfi A, Alsaad A, Taamneh A, Alsyof A (2020) Technology in society the determinants of digital payment systems' acceptance under cultural orientation differences: the case of uncertainty avoidance. *Technol Soc* 63(September)
4. Alamoudi H (2021) Examining retailing sustainability in the QR code-enabled mobile payments context during the COVID-19 pandemic. *Int J Custom Relat Market Manage* 13(1):1–22. <https://doi.org/10.4018/jcrmm.289210>
5. Alhassan MD, Kolog EA, Boateng R (2020) Effect of gratification on user attitude and continuance use of mobile payment services: a developing country context. *J Syst Inf Technol* 22(4):353–380. <https://doi.org/10.1108/JSIT-01-2020-0010>
6. Ali G, Dida MA, Sam AE (2021) A secure and efficient multi-factor authentication algorithm for mobile money applications. *Future Internet* 13(12):1–31. <https://doi.org/10.3390/fi13120299>
7. Ashrafi DM, Easmin R (2023) The role of innovation resistance and technology readiness in the adoption of QR code payments among digital natives: a serial moderated mediation model. *Int J Bus Sci Appl Manage* 18(1):18–45
8. Aydin G, Burnaz S (2016) Adoption of mobile payment systems: a study on mobile wallets. *Pressacademia* 5(1):73–73. <https://doi.org/10.17261/pressacademia.2016116555>
9. Anderson CJ, Gerbing WD (1988) Structural equation modeling in practice: a review and recommended two-step approach James. *Psychol Bull* 15(4):511–538. <https://doi.org/10.1504/EJIM.2021.114662>
10. BCG, PhonePe (2021) Digital payments in India: a US \$ 10 trillion opportunity
11. Bruce Ho C-T, Denis Yang J-M (2017) Factors affecting users' mobile technology usage intentions: an example of QR code scanning for mobile commerce. *Int J Mobile Commun* 15(2):185–209

12. Chang V, Chen W, Xu QA, Xiong C (2021) Towards the customers' intention to use QR codes in mobile payments. *J Glob Inf Manag* 29(6):1–21. <https://doi.org/10.4018/jgim.20211101.0a37>
13. Chin AG, Harris MA, Brookshire R (2022) An empirical investigation of intent to adopt mobile payment systems using a trust-based extended valence framework. *Inf Syst Front* 24(1):329–347. <https://doi.org/10.1007/s10796-020-10080-x>
14. Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q Manage Inform Syst* 13(3):319–339. <https://doi.org/10.2307/249008>
15. Doron A (2012) Mobile persons: cell phones, gender and the self in North India. *Asia Pac J Anthropol* 13(5):414–433. <https://doi.org/10.1080/14442213.2012.726253>
16. de Best R (2023) Mobile payments worldwide—statistics & facts|Statista. Statista. <https://www.statista.com/topics/4872/mobile-payments-worldwide/#topicOverview>
17. Djayapranata GF, Setyawan A (2021) Trust or usefulness? QR code payment among millennials in a disrupted market. *Adv Econ Bus Manage Res* 180(Insyma):194–199
18. Eleazar N, Nurul A (2022) Integrated QR payment system (QRIS): cashless payment solution in developing country from merchant perspective. *Asia Pacific J Inf Sys* 32(3):630–655
19. Iacobucci D (2010) Structural equations modeling: fit Indices, sample size, and advanced topics. *J Consum Psychol* 20(1):90–98. <https://doi.org/10.1016/j.jcps.2009.09.003>
20. India Mobile Payment Market Report, Size, Growth and Forecast 2023–2028 (2023) IMARC. <https://www.imarcgroup.com/india-mobile-payment-market>
21. Kaatz C (2020) Retail in my pocket—replicating and extending the construct of service quality into the mobile commerce context. *J Retail Consum Serv* 53(March 2019):101983. <https://doi.org/10.1016/j.jretconser.2019.101983>
22. Kang H (2021) Sample size determination and power analysis using the G*Power software. *J Educ Eval Health Profess* 18:1–12. <https://doi.org/10.3352/JEEHP.2021.18.17>
23. Karjaluoto H, Shaikh AA, Saarijärvi H, Saraniemi S (2019) How perceived value drives the use of mobile financial services apps. *Int J Inform Manage* 47(September 2017):252–261. <https://doi.org/10.1016/j.ijinfomgt.2018.08.014>
24. Kline RB (2010) Principles and practice of structural equation modeling. *Can Grad J Sociol Criminol* 1(1). <https://doi.org/10.15353/cgjsc.v1i1.3787>
25. Knuchel T, Kuntner T, Pataki EC, Back A (2011) 2D-codes: technology and application. *Bus Inf Syst Eng* 3(1):45–48. <https://doi.org/10.1007/s12599-010-0139-z>
26. Kongarchapatara B (2018) Factors affecting adoption versus behavioral intention to use QR code payment application factors affecting adoption versus behavioral intention to use QR code payment application. Boonying Kongarchapatara * and Chalida Rodjanatara College of Management. In: 2018 international conference on e-commerce, e-administration, e-society, e-education, and e-technology, May
27. Kosim KP, Legowo N (2021) Factors affecting consumer intention on QR payment of mobile banking: a case study in Indonesia. *J Asian Finance Econ Bus* 8(5):391–401. <https://doi.org/10.13106/jafeb.2021.vol8.no5.0391>
28. Krishna Priya P, Anusha K (2019) Fintech issues and challenges in India. *Int J Recent Technol Eng* 8(3):904–908. <https://doi.org/10.35940/ijrte.C4087.098319>
29. Kumar A, Gupta A (2022) India's progress on closing mobile gender gap stalled in last one year: GSMA study. *ET Telecom*. <https://telecom.economictimes.indiatimes.com/news/indias-progress-on-closing-mobile-gender-gap-stalled-in-last-one-year-gsma-study/92494076>
30. Lai VS, Li H (2005) Technology acceptance model for internet banking: an invariance analysis. *Inform Manage* 42(2):373–386. <https://doi.org/10.1016/j.im.2004.01.007>
31. Legris P, Ingham J, Collette P (2003) Why do people use information technology? A critical review of the technology acceptance model. *Inform Manage* 40(3):191–204. [https://doi.org/10.1016/S0378-7206\(01\)00143-4](https://doi.org/10.1016/S0378-7206(01)00143-4)
32. Liébana-cabanillas F, De Luna IR, Montoro FJ, Liébana-cabanillas F, De Luna IR, Montoro-ríos FJ (2015) User behaviour in QR mobile payment system: the QR payment acceptance model 7325. <https://doi.org/10.1080/09537325.2015.1047757>

33. Liébana-Cabanillas F, Ramos de Luna I, Montoro-Ríos FJ (2015) User behaviour in QR mobile payment system: the QR payment acceptance model. *Technol Anal Strat Manage* 27(9):1031–1049. <https://doi.org/10.1080/09537325.2015.1047757>
34. Liébana-Cabanillas F, Sánchez-Fernández J, Muñoz-Leiva F (2014) Antecedents of the adoption of the new mobile payment systems: the moderating effect of age. *Comput Hum Behav* 35:464–478. <https://doi.org/10.1016/j.chb.2014.03.022>
35. Liébana-Cabanillas F, Sánchez-Fernández J, Muñoz-Leiva F (2014) The moderating effect of experience in the adoption of mobile payment tools in virtual social networks: the m-payment acceptance model in virtual social networks (MPAM-VSN). *Int J Inf Manage* 34(2):151–166. <https://doi.org/10.1016/j.ijinfomgt.2013.12.006>
36. McDonald RP, Ho MR (2002) Principles and practice in reporting structural equation analyses. *Psychol Methods* 7(1):64–82. <https://doi.org/10.1037//1082-989X.7.1.64>
37. Meharia P (2012) Payment system and its effects on its' use. *Account Manage Inform Syst* 11(1):97–111. http://online-cig.ase.ro/RePEc/ami/articles/11_1_6.pdf
38. Mookerjee J, Chattopadhyay S, Ahmed T, Addepalli L (2022) Impact of QR-codes as a disruptive technology during the Covid-19 contagion. *Int J Recent Innov Trends Comput Commun* 10(1):284–289. <https://doi.org/10.17762/ijritcc.v10i1s.5850>
39. Muñoz-Leiva F, Hernández-Méndez J, Sánchez-Fernández J (2012) Generalising user behaviour in online travel sites through the Travel 2.0 website acceptance model. *Online Inform Rev* 36(6):879–902. <https://doi.org/10.1108/14684521211287945>
40. Nur T, Gosal GA (2021) Mobile payment usage in online shopping among gen Z in the JABODETABEK area: META-UTAUT approach. August, pp 464–469
41. Nysveen H, Pedersen PE, Thorbjørnsen HT (2005) Intentions to use mobile services: antecedents and cross-service. <https://doi.org/10.1177/0092070305276149>
42. Pandey SK (2022) A study on digital payments system & consumer perception: an empirical survey. *J Positive School Psychol* 2022(3):10121–10131. <http://journalppw.com>
43. Patil P, Tamilmani K, Rana NP, Raghavan V (2020) Understanding consumer adoption of mobile payment in India: extending meta-UTAUT model with personal innovativeness, anxiety, trust, and grievance redressal. *Int J Inf Manage* 54(May):102144. <https://doi.org/10.1016/j.ijinfomgt.2020.102144>
44. Pooi Y, Khalid H, Nadarajah D (2018) ScienceDirect millennials' perception on mobile payment services in Malaysia. *Proc Comput Sci* 124:397–404. <https://doi.org/10.1016/j.procs.2017.12.170>
45. Pratika Y, Salahudin S, Riyanto DWU, Ambarwati T (2021) Analysis of pay later payment system on online shopping in Indonesia. *J Econ Bus Account Ventura* 23(3):329–339. <https://doi.org/10.14414/jebav.v23i3.2343>
46. Premkumar G, Ramamurthy K, Liu H (2008) Internet messaging: an examination of the impact of attitudinal, normative, and control belief systems. *Inform Manage* 45:451–457. <https://doi.org/10.1016/j.im.2008.06.008>
47. Rastogi A, Damle M (2020) Trends in the growth pattern of digital payment modes in 17(December 2016). *Palarch's J Archaeol Egypt/Egyptol* 4896–4927
48. Ricson (2023) QR code usage statistics 2022: 443% scan increase and 438% generation boost. <https://www.qrcode-tiger.com/qr-code-statistics-2022-q1>
49. Rogers EM (1981) Diffusion of innovations: modifications of a model for telecommunications.
50. Rogers EM, Medina UE, Rivera M, Wiley CJ (2005) Complex adaptive systems and the diffusion of innovations. The University of New Mexico. *Small* 10(3):1–26. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.8047&rep=rep1&type=pdf>
51. Ruslan KGM, Suhartito FY, Gui A (2019) QR code payment in Indonesia and its application on mobile banking. *KNe Social Sci* 551–568. <https://doi.org/10.18502/kss.v3i22.5073>
52. Salloum SA, Al-emran M, Khalaf R, Habes M, Shaalan K (2019) An innovative study of e-payment systems adoption in higher education: theoretical constructs and empirical analysis. *Int J Interact Mobile Technol* 6:68–83
53. Sharma S, Deivakani M, Reddy KS, Gnanasekar AK, Aparna G (2021) Key enabling technologies of 5G wireless mobile communication. *J Phys Conf Ser* 1817(1). <https://doi.org/10.1088/1742-6596/1817/1/012003>

54. Suebtimrat P, Vonguai R (2021) An investigation of behavioral intention towards QR code payment in Bangkok, Thailand. *J Asian Finance Econ Bus* 8(1):939–950. <https://doi.org/10.13106/jafeb.2021.vol8.no1.939>
55. Sun Y, Li T, Wang S (2021) “I buy green products for my benefits or yours”: understanding consumers’ intention to purchase green products. *Asia Pacific J Market Logist* 71974177. <https://doi.org/10.1108/APJML-04-2021-0244>
56. Taylor S, Todd PA (1995) Understanding information technology usage: a test of competing models. October 2014
57. Thompson RL, Higgins CA, Howell JM (1994) Influence of experience on personal computer utilization: testing a conceptual model. *J Manag Inf Syst* 11(1):167–187. <https://doi.org/10.1080/07421222.1994.11518035>
58. Times of India (2022) Women, unemployed, rural poor lagging due to digital divide. Oxfam India Report. <https://economictimes.indiatimes.com/news/india/g20-igniting-socio-economic-growth-in-kashmir-a-thriving-south-asia-beckons/articleshow/100087896.cms>. Accessed 9 May 2023
59. Timmons DM, Losordo T, Hundley P (2014) *Business research methods*, 12th edn, vol 1804. McGraw-Hill
60. Tornatzky LG, Klein KJ (1982) Innovation characteristics and innovation adoption-implementation: a meta-analysis of findings. *IEEE Trans Eng Manage* EM-29:28–45
61. Tu M, Wu L, Wan H, Ding Z, Guo Z, Chen J (2022) The adoption of QR code mobile payment technology during COVID-19: a social learning perspective. *Front Psychol* 12(February):1–10. <https://doi.org/10.3389/fpsyg.2021.798199>
62. Tusińska M (2021) The digital gender divide. A focus on inclusion through mobile phone use in India. *Nierówności Społeczne a Wzrost Gospodarczy*, 67(3):16–29. <https://doi.org/10.15584/nsawg.2021.3.2>
63. Venkatesh V, Davis FD (2000) A theoretical extension of the technology acceptance model: four longitudinal field studies. October 2018, pp 185–204
64. Wang L, Dai X (2020) Exploring factors affecting the adoption of mobile payment at physical stores. *Int J Mobile Commun* 18(1):67. <https://doi.org/10.1504/ijmc.2020.104420>
65. Wei MF, Luh YH, Huang YH, Chang YC (2021) Young generation’s mobile payment adoption behavior: analysis based on an extended UTAUT model. *J Theor Appl Electron Commer Res* 16(4):1–20. <https://doi.org/10.3390/jtaer16040037>
66. Wong WH, Mo WY (2019) A study of consumer intention of mobile payment in Hong Kong, based on perceived risk, perceived trust, perceived security and technological acceptance model. *J Adv Manage Sci* 7(2):33–38. <https://doi.org/10.18178/joams.7.2.33-38>
67. Xiong C, Chang V, Scuotto V, Shi Y, Paoloni N (2019) The social-psychological approach in understanding knowledge hiding within international R & D teams: an inductive analysis. *J Bus Res*. <https://doi.org/10.1016/j.jbusres.2019.04.009>
68. Zhang A, Yue X, Kong Y (2011) Exploring culture factors affecting the adoption of mobile payment. <https://doi.org/10.1109/ICMB.2011.32>
69. Zhao Y, Bacao F (2021) How does the pandemic facilitate mobile payment? An investigation on users’ perspective under the COVID-19 pandemic. *Int J Environ Res Public Health* 18(3):1–22. <https://doi.org/10.3390/ijerph18031016>
70. Zhong Y, Moon H-C (2022) Investigating customer behavior of using contactless payment in China: a comparative study of facial recognition payment and mobile QR-code payment. *Sustainability* 14: 7150

Performance Comparison of Various YOLO Models for Vehicle Detection: An Experimental Study



Sourajit Maity, Arpan Chakraborty, Pawan Kumar Singh, and Ram Sarkar

Abstract Development of automatic vehicle detection (AVD) systems using either images or videos from traffic scenarios would be quite beneficial for making an automated traffic management system. There is an abundance of AVD-based research articles that have been published in the literature. This paper focuses on three major object detection algorithms under the You Only Look Once (YOLO) family, namely YOLOv5, YOLOv7, and YOLOv8 for AVD. This paper also discusses the architectural differences found in these variants of YOLO models. For experimental evaluation, we have used two recently introduced AVD datasets developed for the Indian subcontinent, namely JUVDSi v1 and IRUVD. We have achieved a satisfactory outcome on the IRUVD dataset with a mAP score of 0.96 using the YOLOv7 model and mAP score of 0.817 on the JUVDSi v1 dataset using the YOLOv8 model. The code and detailed results can be found at: <https://github.com/JUVDSi/yolo-comparison.git>.

Keywords Automatic vehicle detection · YOLO model · Deep learning · Traffic monitoring · JUVDSi v1 dataset · IRUVD dataset

Sourajit Maity and Arpan Chakraborty contributed equally to this work

S. Maity (✉) · A. Chakraborty · R. Sarkar
Department of Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Mullick Road, Kolkata 700032, West Bengal, India
e-mail: sourajit.cse.ju@gmail.com

P. K. Singh
Department of Information Technology, Jadavpur University, Jadavpur University Second Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Salt Lake City, Kolkata 700106, West Bengal, India

1 Introduction

Since the onset of the industrial era, vehicles and their management systems have been undergoing continuous upgradation for the betterment of our day-to-day life. In the upcoming years, studies regarding autonomous driving are going to bifurcate many pathways in the research related to vehicles. However, working on issues based on real-life traffic scenarios is quite challenging in terms of training, testing and validating the model. On the other hand, an ample amount of data is required to make an efficient automatic vehicle detection (AVD) model that gives a notable accuracy and is capable of working in real-life scenarios.

You Only Look Once (YOLO) [1] is a popular model which can find out multiple objects present within an image at any instance and locate them by drawing bounding boxes around the objects. Then, it proceeds the image using a convolutional neural network (CNN) model only once to get the output. Hence, it is named as YOLO. In contrast to the typical CNN pipeline [2], YOLO segregates the bounding boxes and their related class probabilities spatially, which are predicted using a single neural network, and identifies object detection as a regression problem.

From the literature, it has been observed that further enhancements are required to improve the accuracy of AVD systems. Some research gaps are mentioned below:

- The majority of datasets captured in the Indian subcontinent had a single frame with multiple classes of objects, and hence, existing algorithms may not work well on multi-class images.
- Multi-view or multi-modal datasets are not available for the detection of vehicles. Lots of research and datasets are required to make a practical solution for an AVD system.

In this paper, we focus on three major object detection algorithms under the YOLO family, namely YOLOv5, YOLOv7, and YOLOv8 for the purpose of vehicle detection, and discuss the architectural differences of these variants. We then do a performance comparison of these models, and in doing so, we use two recently introduced AVD datasets developed for the Indian subcontinent, namely JUVDSi v1 and IRUVD.

2 Literature Review

Different approaches have been applied in order to solve the problems of AVD. However, these techniques have some issues in terms of output accuracy, resource cost, and processing speed. Recently, Maity et al. [3] presented a comprehensive survey of different state-of-the-art AVD methods proposed during the last decade. In Table 1, a brief summary of a few significant recently developed AVD methods is given.

Table 1 Different models and datasets recently proposed for AVD problem

References	Model	Dataset	mAP score (%)
Zhang et al. [4]	YOLOv4	BIT-Vehicle dataset	0.90
Bhattacharya et al. [5]	Ensemble using weighted fusion box	JUVDsi v1	0.74
Zuraimi et al. [6]	YOLOv4	Self-made	0.82
Miao et al. [7]	YOLOv3 [8]	Self-made	0.93
Chernikova et al. [8]	NVIDIA model	Udacity self-driving cars	0.69
Lu et al. [9]	YOLO	VEDAI	0.76

There are some shortcomings of the existing AVD methods. There is a great difference in the traffic scenarios of the Indian subcontinent and those of any developed country of Europe or America. In developing countries like India, Bangladesh, etc., traffic congestion and road blockers are quite common. Therefore, the overlapping of multiple vehicles occurs in a single image frame.

3 YOLO Models and Their Architectural Comparisons

YOLO: The basic YOLO model [10] was initiated as a unified algorithm where all the distinct components were merged into a single neural network as the final pipeline. YOLO is a regression-based algorithm. It simultaneously predicts the class probabilities of the objects as well as the bounding boxes specifying their locations, for the entire image. The bounding boxes of the object are described as: b_x, b_y with the x, y coordinates representing the center of the box relative to the bounds of the grid cell. YOLO model takes the entire image as an input and divides it into $S \times S$ grids. After that, image classification and object localization techniques are applied to each grid, which is then assigned as a label. The label of a grid, without any object, is indicated as zero.

YOLOv5: YOLOv5 [11] was established in 2020 by the same team which developed YOLO algorithm. It was an advanced edition of the previous versions which was improved by the addition of several new features. YOLOv5 used a more complex network architecture called EfficientDet based on the EfficientNet architecture. YOLOv5 achieved higher accuracy and better generalization for a variety of object categories.

YOLOv7: YOLOv7 [12] introduced four major architectural reforms and a trainable Bag of Freebies (BOF) in the existing YOLO architecture. Architectural reforms included Extended Efficient Layer Aggregation Network (ELAN) and model scaling for concatenation-based models.

YOLOv8: Developed by Ultralytics, YOLOv8 introduced new features to improve the performance as well as flexibility. It was developed to be fast and user-friendly, which made it an excellent choice for various object detection, instance segmentation,

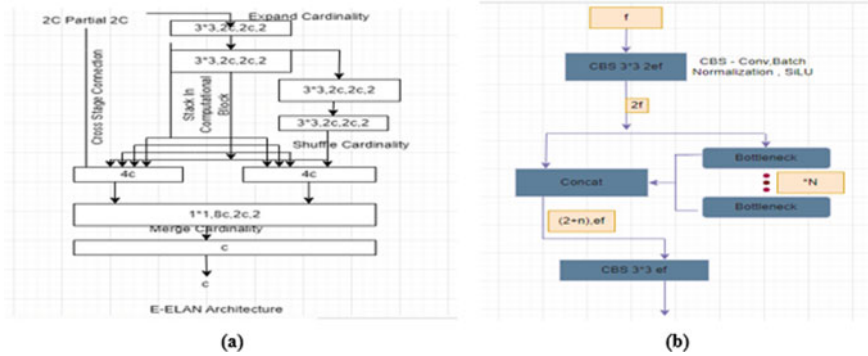


Fig. 1 Architectural overview of: a YOLOv7, and b YOLOv8

image classification, and pose estimation tasks. It included numerous architectural and developer experience changes and improvements over YOLOv5. In Fig. 1a and b, the architecture of YOLOv7 and YOLOv8 models is shown, respectively.

4 Datasets

Performance comparisons of different YOLO variants are made on two publicly available AVD datasets prepared for the Indian context, as mentioned below.

JUVDsi v1 [5]: JUVDsi v1 dataset consists of nine distinct vehicle classes, namely, ‘Truck’, ‘Bus’, ‘Minitruck’, ‘Car’, ‘Autorickshaw’, ‘Van’, ‘Rickshaw’, ‘Motorbike’, and ‘Cycle’. The training set of the dataset contains images which are captured in various weather conditions among which 872 images are taken in sunny weather conditions, 651 images are taken in cloudy weather conditions, and 565 images are captured at night. There are 629 images containing a single object in the entire image, and 1459 images include multiple objects in the entire image. JUVDsi v1 is publicly available at: <https://github.com/JUVDsi V1/JUVDsi V1si>.

IRUVD [13]: This dataset consists of 14 classes, which includes several vehicle classes that are quite common in the rural areas in India. Vehicle classes such as ‘Taxi’, ‘Tempo’, ‘Motor-Rickshaw’, ‘Toto’, and ‘Cycle Rickshaw’ are also included in this dataset. There are 14,343 annotations and 400 high-quality images in the dataset. The dataset is publicly available at: <https://github.com/IRUVD/IRUVD>.

5 Results and Discussion

The evaluation metrics that are used to calculate the performance of predictions are Precision (P), Recall (R), and Mean Average Precision (mAP).

5.1 Results

Table 2 provides the obtained results of the three YOLO models on two different AVD datasets. For JUVDSi v1 dataset, we achieve a 0.755 mAP score on applying YOLOv5 model, 0.816 mAP score on applying YOLOv7 model, and 0.817 mAP score on applying YOLOv8 model. In IRUVD dataset, YOLOv5, YOLOv7, and YOLOv8 models produced mAP scores of 0.893, 0.960, and 0.946, respectively. It is to be noted that all the three YOLO models are made to run for 25 epochs. Some outputs on JUVDSi v1 and IRUVD datasets with varied YOLO models are represented in Figs. 2 and 3, respectively.

Table 2 Results produced by different YOLO models

Dataset	Model	<i>P</i>	<i>R</i>	mAP score @50	mAP score @50–95%
JUVDSi v1	YOLOv5	0.97	0.95	0.755	0.56
	YOLOv7	0.98	0.97	0.816	0.615
	YOLOv8	0.99	0.98	0.817	0.67
IRUVD	YOLOv5	0.99	0.98	0.893	0.723
	YOLOv7	0.99	0.99	0.960	0.811
	YOLOv8	0.99	0.97	0.946	0.91



Fig. 2 Some output images using different YOLO versions on JUVDSi v1 dataset

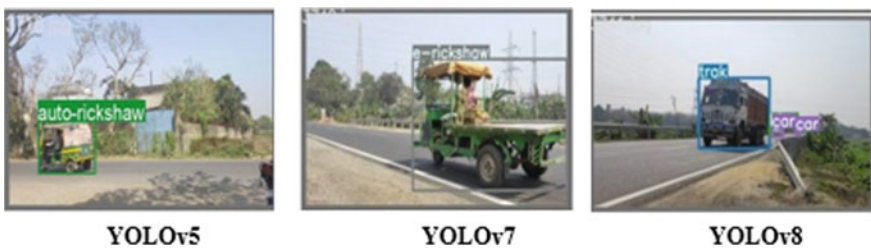


Fig. 3 Some output images using different YOLO versions on IRUVD dataset

5.2 Comparative Analysis

We have considered two different AVD datasets for experimenting with YOLOv5, YOLOv7, and YOLOv8 models. Figures 4 and 5 show Precision–Recall (PR) curves after executing three different YOLO variants on JUVDSi v1 and IRUVD datasets, respectively.

After analyzing YOLO versions on AVD datasets, we have found that the YOLOv7 model attains the best accuracy in Indian road scenario, and it shows a mAP score of 0.96 @0.5 on IRUVD dataset. The confusion matrices obtained after applying three different YOLO versions on IRUVD and JUVDSi v1 datasets are depicted in Figs. 6 and 7, respectively.

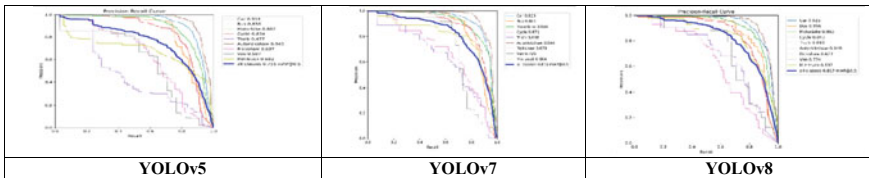


Fig. 4 PR curves obtained on the JUVDSi v1 dataset

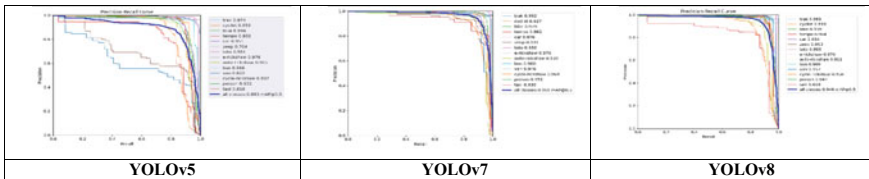


Fig. 5 PR curves obtained on the IRUVD dataset

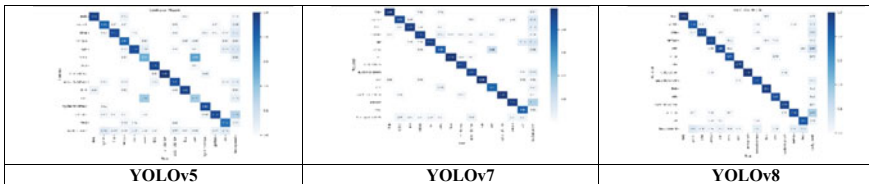


Fig. 6 Confusion matrices obtained on the IRUVD dataset

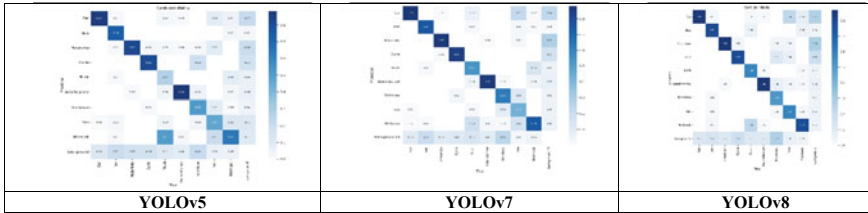


Fig. 7 Confusion matrices obtained on the JUVDSi v1 dataset

5.3 Limitations

1. We have not tested the models on AVD datasets where images are captured under different weather conditions such as rainy, foggy, dark night conditions.
2. The considered datasets have limited number of vehicle classes which may be a concern to use the developed system for practical scenarios.

6 Conclusion and Future Work

Computer-based automatic vehicle detection methods have a greater role in the fields of traffic monitoring, driver assistance, and surveillance. This paper focuses on providing a thorough performance study on AVD using three standard object detection algorithms under the YOLO family, namely YOLOv5, YOLOv7, and YOLOv8. In doing so, two publicly available AVD datasets developed for the Indian sub-continent, namely JUVDSi v1 and IRUVD, have been considered. It has been observed that the YOLOv8 model performs significantly well on the JUVDSi v1 dataset, whereas the YOLOv7 model performs significantly well on the IRUVD dataset.

In future, we plan to consider more diverse datasets to check the robustness of the models. Also, we will experiment with other models, like Faster R-CNN and YOLO-NAS models, etc. We may explore some ensemble methods to combine the results of the individual models.

References

1. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
2. Bhattacharyya D, Bhattacharyya A, Agrebi M, Roy A, Singh PK (2022) DFE-AVD: deep feature ensemble for automatic vehicle detection
3. Maity S, Bhattacharyya A, Singh PK, Kumar M, Sarkar R (2022) Last decade in vehicle detection and classification: a comprehensive survey. Arch Comput Methods Eng 29:1–38

4. Zhang Y, Guo Z, Wu J, Tian Y, Tang H, Guo X (2022) Real-time vehicle detection based on improved YOLO v5. *Sustainability* 14(19):12274
5. Bhattacharyya A, Bhattacharya A, Maity S, Singh PK, Sarkar R (2023) JUVDSI V1si v1: developing and benchmarking a new still image database in Indian scenario for automatic vehicle detection. *Multimed Tools Appl* 82:1–33
6. Bin Zuraimi MA, Zaman FHK (2021) Vehicle detection and tracking using YOLO and DeepSORT. In: 2021 IEEE 11th IEEE symposium on computer applications & industrial electronics (ISCAIE), pp 23–29
7. Miao Y, Liu F, Hou T, Liu L, Liu Y (2020) A nighttime vehicle detection method based on YOLO v3. In: 2020 Chinese automation congress (CAC), pp 6617–6621
8. Chernikova A, Oprea A, Nita-Rotaru C, Kim B (2019) Are self-driving cars secure? Evasion attacks against deep neural networks for steering angle prediction. In: 2019 IEEE security and privacy workshops (SPW), pp 132–137
9. Lu J et al (2018) A vehicle detection method for aerial image based on YOLO. *J Comput Commun* 6(11):98–107
10. Shafiee MJ, Chywl B, Li F, Wong A (2017) Fast YOLO: a fast you only look once system for real-time embedded object detection in video. *arXiv Prepr. arXiv1709.05943*
11. Zhu X, Lyu S, Wang X, Zhao Q (2021) TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2778–2788
12. Wang C-Y, Bochkovskiy A, Liao H-YM (2022) YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv Prepr. arXiv2207.02696*
13. Ali A, Sarkar R, Das DK (2023) IRUVD: a new still-image based dataset for automatic vehicle detection. *Multimed Tools Appl* 1–27

Author Index

A

Abhishek Chaudhary, 633
Al-Alawy, Faiz, 205
Albuquerque de, Victor Hugo C., 387
Alex David, S., 599
Alkhayat, Ahmed, 103
Amita Jain, 159
Amit Kothari, 471
Amod Kumar, 423
Anamika Chauhan, 115, 585
Anirudh Saxena, 537
Anshika, 537
Anshul Jain, 659
Anuja Patil, 483
Anuj Kumar Singh, 337
Anuradha, T., 379
Anurag Singh, 537
Anushka Singh, 293
Arahant Panwar, 65
Arpan Chakraborty, 677
Arunima Jaiswal, 293
Aryan Singhania, 75
Ashish Khanna, 513
Ashwin Prajeeth, 525
Atrayee Majumder Ray, 513
Ayush Sharma, 145, 551

B

Binav Gautam, 525
Bindu Garg, 265
Biplab Das, 513
Brahmaleen K. Sidhu, 495
Brayyich, Mohammed, 1

C

Carmel Mary Belinda, M. J., 599

D

Deepak Kumar Sharma, 565
Dhanashree Lavekar, 483
Dhruba Datta, 55
Divya Gandhi, 551

E

Ekta Singh, 565

G

Garima Chandel, 169, 251
Garima Chhikara, 525
Gayatri Kalshetti, 265
Gomatheeswari Preethika, C., 459
Gunnala Rajesh, 181
Gurjit Singh Bhathal, 495

H

Hadiyanto, 399
Harsh Chaudhari, 483
Harsh Gupta, 75
Harsh Prakash, 55
Himesh Mahabi, 13
Hina Gupta, 647

I

Isha Dubey, 565

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

A. Swaroop et al. (eds.), *Proceedings of Data Analytics and Management*, Lecture Notes in Networks and Systems 787, <https://doi.org/10.1007/978-981-99-6550-2>

J

Janet, B., 615
 Japnit Singh, 65
 Jayesh Jain, 43
 Jehlol, Hashem Bedr, 205
 Jothilakshmi, P., 459
 Jyoti Mishra, 387

K

Kalathiripi Rambabu, 181
 Kampa Lavanya, 223
 Kannan, E., 599
 Kapil Sharma, 115
 Keshavagari Srujana, 181
 Khyati Kochhar, 659
 Kishan Kumar Garg, 65
 Kumud Kundu, 43

M

Madhavi Reddy, Y., 223
 Madhvan Sharma, 13
 Malti Bansal, 23
 Manav Misra, 43
 Manjeet Kumar, 365
 Manmeet Singh, 43
 Mansimran Rehal, 551
 Mayank Goel, 43
 Meenakshi Sood, 423
 Meenal Job, 133
 Menaka, R., 325
 Menezes, José Wally M., 387
 Minni Jain, 75, 159
 Mohammed, Ahmed Abdulateef, 205
 Mohammed Essam, K., 103
 Mohammed Imran, 181
 Mohanasundaram, R., 459
 Monika, 565
 Monika Shah, 471
 Mrigank Sondhi, 145
 Muhammed Shanir, P. P., 251
 Mullangi David, 195
 Muthmainnah, 1

N

Narayan Jee Jha, 85
 Nguyen Thi Thuy, A., 239
 Nilutpol Bora, 585
 Nitin Sachdeva, 293

O

Obaid, Ahmad J., 1

P

Padmavathi, G., 325
 Palash, 365
 Patil, Y. M., 349
 Pawan Kumar Singh, 677
 Payal Porwal, 293
 Perepi Durga Teja, 195
 Pooja Kumari, 293
 Prambayun, Arif, 399
 Pranav Khairnar, 483
 Pravin Vilasrao Sawant, 349
 Preeti Nagrath, 387
 Priyadeep Bhalla, 293
 Priyanka Yadav, 659
 Priya Singh, 55

R

Rachna Jain, 265
 Rahmadi, Lendy, 399
 Rajesh Kumar Meena, 423
 Rajni Jindal, 159
 Ram Sarkar, 677
 Ram Suchit Yadav, 133
 Raviraj Joshi, 483
 Reshav Kalyani, 365
 Rishabh Tater, 387
 Rishav Sinha, 85
 Ritesh Kumar, 435
 Ritik Rao, 435
 Riyanshi Arora, 23
 Rohit Ahuja, 551
 Ronitt Mehra, 365
 Ruchika Malhotra, 145, 447
 Ruth Naveena, N., 599

S

Sabyasachi Pramanik, 513
 Sachin Kumar, 337
 Sachin Pande, 483
 Sai Keshari, 23
 Sai Prudhvi Vallurupalli, 379
 Sakshi Panchal, 23
 Samarpit Karar, 311
 Sameer Kumar, 85
 Sandeep Kumar Saini, 169
 Sanjaya, Ridwan, 399
 Sanjay Dubey, 181
 Sanjay Patidar, 13

Santhadevi, D., [615](#)
Sarvani, A., [223](#)
Setu Garg, [251](#)
Shaik Abdul Riyaz, [195](#)
Shallu Juneja, [495](#)
Shanti Verma, [279](#)
Shashank Paul, [633](#)
Shridhar Sharma, [265](#)
Shweta Meena, [447](#)
Sitaram Meena, [423](#)
Sourajit Maity, [677](#)
Sudarshan Khandelwal, [265](#)
Suyash Agrawal, [265](#)

T

Tamizharasi Seetharaman, [103](#)
Tirupathiraju Kanumuri, [311](#)

Tisha Sadariya, [279](#)
Trasha Gupta, [85](#)

V

Vandana Sharma, [103](#)
Venu Gopal, K., [195](#)
Vijaya, R., [223](#)

Y

Yakin Al, Ahmad, [1](#)
Yalla Sowmya Reddy, [223](#)
Yash Vardhan Varshney, [251](#)

Z

Zaheeruddin, [647](#)