Machine Learning and Deep Learning Based Traffic Classification and Prediction in Software Defined Networking

Ayşe Rumeysa Mohammed, Shady A. Mohammed, and Shervin Shirmohammadi
Distributed and Collaborative Virtual Environments Research Lab (DISCOVER Lab)
University of Ottawa, Ottawa, Canada
{amus037 | smoha191}@uottawa.ca
shervin@discover.uottawa.ca

***Abstract*—The Internet is constantly growing in size and becoming more complex. The field of networking is thus continuously progressing to cope with this monumental growth of network traffic. While approaches such as Software Defined Networking (SDN) can provide a centralized control mechanism for network traffic measurement, control, and prediction, still the amount of data received by the SDN controller is huge. To process that data, it has recently been suggested to use Machine Learning (ML). In this paper, we review existing proposal for using ML in an SDN context for traffic measurement (specifically, classification) and traffic prediction. We will especially focus on approaches that use Deep learning (DL) in traffic prediction, which seems to have been mostly untapped by existing surveys. Furthermore, we discuss remaining challenges and suggest future research directions.**

***Keywords*—Network measurement, software defined networking, machine learning, deep learning, traffic classification, traffic prediction.**

I. Introduction

The fast development of the internet and communication devices has created bigger and more complicated network structures, adapting and developing bigger hubs, routers, switches, etc. This complexity in networks has introduced an overflow of vast amounts of traffic data and contributed to the challenges in network management and traffic optimization, including traffic measurement (e.g. traffic classification) and traffic prediction.

In parallel, we are seeing two promising solutions to help manage networks more efficiently: SDN and Machine Learning. SDN provides a centralized access and control mechanism to all networking devices, where the SDN controller can not only monitor and measure all sorts of network parameters and metrics, but can also make a more informed and efficient decision about resources allocation and routing, since it has a global view of everything in the network. However, the amount of data an SDN controller receives could be overwhelming. While the SDN controller itself can be made scalable, for example by running it in a cloud, still efficient algorithms are needed to extract the required measurements and information from the received data. Here is where Machine Learning can help. Many of the traffic classification and traffic prediction issues can be performed efficiency by various ML algorithms, improving the system performance while maintaining relative simplicity in design.

In this survey, we review existing approaches for traffic classification and traffic prediction which use ML in an SDN context. We especially focus on ML's subcategory of Deep Learning (DL), which has not been covered in details by existing surveys. Therefore, our contribution is covering DL methods for traffic prediction, which is mostly not covered in the existing surveys, while we also cover some newer works in ML and DL for both traffic classification and traffic prediction that existing surveys have not covered. Finally, we investigate open research issues and suggest possible future research avenues.

The remainder of the paper is organized as follows. In Section II, we briefly explain the background knowledge. In Section III, we summarize the related work. In Section IV, we introduce the survey. Section V discusses the challenges and recommends future research directions, and finally Section VI concludes our work.

II. Background

In this section, first we explain the ML and DL algorithms mentioned in this survey. Thereafter, we present background knowledge on SDN.

A. Traditional Machine Learning Algorithms

ML is a data analysis method that learns from data to spot patterns within it and make decisions based on the information collected. It generally involves preprocessing, training and testing phases. The preprocessing includes actions such as data preparation, filtering, imputation, and tuning for specific purposes. Once the data is preprocessed, ML methods are implemented to train the data. Then the system makes decisions based on the input received from the training phase. ML algorithms can be studied under supervised or unsupervised learning where the former is given labeled training data and the latter works with unlabeled training data trying to extract information through clustering according to the resemblance within the observation points. The following are the ML and

DL algorithms used in this survey. Note that all but the last one are supervised:

(i) *Nearest Centroid (NC).* It computes the centroid for each labeled class. It calculates the distance between the observation points and the centroid. Then it assigns the data points to the class whose centroid has the minimum distance to the observation.

(ii) *Naive Bayes (NB).* It is a simple probabilistic classifier based on implementation of Bayes' theorem. It is used when the data dimensionality is high since it assumes the data features are independent from each other.

(iii) *Decision Tree (DT).* It is yet another simple algorithm. It performs a decision classifier through a tree-like model with leaf nodes which correspond to the class label, and the path from the tree roots to the leaf are associated with the classification rules.

(iv) *Random Forest Tree (RF).* It is an extension of DT that aggregates few DTs and fixes the overfitting problem by randomly selecting a subset of data features.

(v) *Support Vector Machine (SVM).* It is a binary classification and pattern recognition technique which maps the data points in n-dimensional space and plots the hyper plane that separates them into different clusters.

(vi) *Multi-Class Support Vector Machine (MCSVM).* In order to segregate the data into more than two classes, SVM is applied as a series of binary problems. However this is computationally expensive. Therefore new methodologies are developed to mitigate this issue [1].

(vii) *Laplacian Support Vector Machine (LapSVM).* It is an extension of SVM which regularizes the SVM by a Laplacian graph [2].

(viii) *Adaptive Boosting (AdaBoost).* It is a boosting technique that builds more accurate algorithms by creating a hybrid classifier out of weak classifiers.

(ix) *Gradient Adaptive Boosting (G-AdaBoost).* It functions in three steps: optimization of a loss function, predictions from a weak learner, and minimization of the loss function via the hybrid model of the weak learners.

(x) *M5Rules*. It can be found under Weka software. It makes decisions for prediction problems by combining decision trees and linear regression.

(xi) *Linear Regression.* When used for prediction and forecasting purposes, it tries to fit a model to data points based on independent variables.

(xii) *Polynomial Regression.* It shifts a linear regression model into a curve to better fit the observation points.

(xiii) *K-means.* Also referred to as *k-means clustering*. Unlike other methods mentioned, k-means is an unsupervised learning algorithm. It divides the data into k different clusters in which each data point is assigned to a cluster with the nearest mean value.

B. Deep Learning Algorithms

Deep Learning uses multiple layered neural networks which are biologically-inspired computing systems with input, hidden and output layers consisting of interconnected neuron-like nodes. The nodes contain activation functions. Information is fed through the input layer. The pattern recognition process is done in the hidden layer via activation functions and the answer is presented in the output layer. Each layer takes the output of the previous layer(s) as input and applies non-linear transformation to extract useful features for classification.

(i) *Convolutional Neural Network (CNN).* It is a type of NN that is build around three ideas: convolutional layers, weight sharing, and pooling. Convolutional layers and weight sharing function as filters that detect localized features in the data and decrease the number of data parameters whereas pooling further reduces the feature size while keeping the invariance of the data.

(ii) *Autoencoders (AE).* It is an unsupervised learning algorithm that encodes the data through dimensionality reduction. It trains the network by reconstructing its input. Its variations are sparse, denoising, contractive, convolutional, stacked.

(iii) *Recurrent Neural Network (RNN).* It is a network with loops that preserves its input due to its internal memory. Just like a human behavior, when it makes a decision, it takes into consideration the current information it has and previous experience gained through loops. Its most popular implementation is Long Short-Term Memory (LSTM). It backpropagates the errors through layers to learn in a recurrent manner.

C. Software Defined Networking

Managing computer network devices such as routers, switches, middle-boxes are challenging and complex. In order to perform a small change in a network's high-level policy, network operators need to configure each network device manually using low-level and most of the time vendor specific commands. Moreover, current networks are vertically integrated; i.e., the control and data planes are bundled together. These difficulties have led the network to be more rigid and to resist any new innovative concepts. For instance, the transition from IPv4 to IPv6 is taking more than two decades and still most of it is incomplete [3] [4]. These problems have led the researchers to focus their efforts to develop solutions that suit network evolution and mitigate scalability limitations.

SDN is an innovative way of network management and configuration. It divides the network into two main planes: control and data planes [5]. Control plane is the centralized network logic that dictates the overall network behavior. On the other hand, routers and switches in the data plane have become simple forwarding devices that learn packets' routing paths directly from the controller via a well-defined interface (API) [6]. This paradigm facilitates creating and introducing new abstractions in networking and managing existing networks.

III. Related Work

In a comprehensive review, Xie et al. [7] investigated ML algorithms employed in SDN particularly for traffic classification, routing optimization, QoS & QoE prediction, resource management and security. While Yan and Yuan [8] covered

thoroughly the traffic classification methods in SDN, Sultana et al. [9] solely focused on ML techniques for detecting network intrusions in SDN and explained the tools created in an SDN environment for this purpose.

In this survey, we focus on DL methods for traffic prediction, which is not covered in any of the above surveys. In addition, we also cover some newer works in ML and DL for both traffic prediction and classification that the above surveys have not covered. In Table I, we provide an overview of the existing surveys and their coverage.

TABLE I
RELATED SURVEYS

Reference	Targeted Problems	Year
[7]	traffic classification, routing optimization, QoS&QoE prediction, resource management and security	2018
[8]	traffic classification	2018
[9]	detecting network intrusions	2018
Our paper	traffic classification & prediction	2019

IV. TRAFFIC CLASSIFICATION AND PREDICTION IN SDN WITH ML AND DL

A. Traffic Classification

Traffic classification is crucial in optimizing internet access and user experience. Since the available bandwidth is limited, by classifying traffic we make the best use of the bandwidth and internet service providers can manage the resources by prioritizing the flow of packets.

Traffic classification can be achieved by identifying the network applications or group of applications. One of the basic approaches is port-based. However it is not practiced anymore due to unsatisfactory classification results since modern applications run on dynamic ports. The alternative to port-based is payload-based approach which is often referred to as deep packet inspection (DPI). DPI identifies the application by inspecting the content of the packet and yields better classification results. Nonetheless, it introduces several challenges. First, it consumes resources since the packets are treated as stacks and identifying a pattern within a packet is computationally expensive. Second, it cannot recognize encrypted traffic, which is quite prevalent these days. Hence, flow-based approaches using ML and DL are used to overcome the limitations of classification. ML methods try to detect patterns within the applications based on the selected feature sets. They can classify the encrypted traffic and work with a lower computational cost. Table II, shows a summary of the surveyed papers for traffic classification.

Xiao et al., [10] presented a low cost learning method to catch elephant flows in real time. The proposed strategy includes two-stage elephant flow detection. At first, suspicious elephant flows are distinguished from mice flows. At the second stage, after using a feature selection module, a correlation-based filter creates the optimum features in the dataset. Then elephant flows are used for improving the classification accuracy while decision trees classify them as real elephant flows or suspicious flows.

Suárez-Varela and Barlet-Ros [11] proposed a monitoring system consisting of flow-level measurement reports with labels classifying flows at the application layer. They utilized flow sampling to manage the processing overhead in SDN controllers and to arrange the required memory in switches for flow measurements by applying hybrid classification and ML techniques. Then they classified both encrypted and unencrypted traffic. For unencrypted traffic, they simply monitored every flow and then applied classification techniques by application protocol whereas for the web and encrypted traffic DPI techniques are implemented to determine which applications generate each flow.

Abubakar and Pranggono [12] designed a scalable flow-based intrusion detection system model to monitor traffic and detect attacks in the proposed star topology with hosts and servers connected to the OpenFlow OVS-switch. They applied neural networks to detect all possible anomalies and attacks. Whereas Naseer et al., [15] analyzed the usage of deep learning algorithms, specifically CNN, AE, and RNN for detecting anomaly-based intrusions. They trained these models with the NSLKDD training data set and then tested them on NSLKDDTest+ and NSLKDDTest21. In addition, extreme learning machine, nearest neighbor, DT, RF, SVM, NB, and quadratic discriminant analysis classification techniques were implemented for intrusion detection as well. Once the intrusion detection was achieved, the performances of DL and ML approaches were assessed by using receiver operating characteristics, area under curve, precision-recall curve, mean average precision and accuracy of classification.

Amiri et al., [13] proposed a fair bandwidth utilization framework that aims to improve QoS for game traffic. Firstly, the incoming traffic was classified real-time based on its application classes by the traffic classifier using an ML technique. Then the application classes were used as inputs to the routing policy generator module to determine the distribution of traffic flows for better bandwidth allocation while prioritizing the traffic for game users.

Fan and Liu [14] studied supervised and unsupervised ML algorithms. They mainly focused on SVM and k-means clustering for traffic classification. Furthermore, they investigated how model tuning and feature selection affect the performance of the classification techniques.

Gangadhar and Sterbenz [16] utilized ML algorithms to provide flexibility for traffic by extending the range of capabilities of the controller in SDNs. They proposed a system that gives the controller the ability to make decisions in real-time on suspicious packets, and once the decision is made, the controller drops the malicious flow. The decision making algorithms, which were DT, SVM, and NB, were trained using MIT KDD 1999 dataset.

He et al., [17] studied ML classifiers such as K Nearest Neighbors, SVM, DT (RF and ET), AdaBoost, NB, and Multi-Layer Perceptron (MLP) to examine the impact and the effectiveness of protocol types and flow features on clas-

TABLE II
TRAFFIC CLASSIFICATION IN SDN

Ref.	Objective	Network Topology	Simulator	ML/DL Technique	Data Features
[10]	To introduce cost-sensitive learning method to define a real-time elephant flow detection strategy and the subsequent metric in flow detection	Not mentioned	Mininet	Decision Trees	flow features (src_ip, src_port, dst_ip, dst_port, protocol)
[11]	To accurately classify encrypted and unencrypted traffic	Not mentioned	Not mentioned	Decision Tree	src port, dst port, src ip, dst ip, IP protocol and size of the first few packets (max. 6 packets)
[12]	To detect intrusion for SDN	Star topology	Mininet	Neural Networks	(i) duration: length of the connection; (ii) protocol_type; (iii) service: network service on the dest.; (iv) src_bytes: number of data bytes from src. to dest.; (v) dst_bytes: number of bytes from the dest. to src.; (vi) count: number of connections to the same host as the current connection in the past 2 sec; (vii) srv_count: number of connections to the same service as the current connection in the past 2 sec
[13]	To solve the bandwidth allocation problem in cloud computing data center networks	Tree-based topology (fat-tree)	Mininet	Nearest Centroid, Nave Bayes, Multi-Class Support Vector Machine	total_packets; total_volume; pktl; biat; duration; active; idle; sflow_packets; sflow_bytes; fpsh_cnt; total_hlen
[14]	To characterize internet traffic by using payload- independent traffic statistics	Not mentioned	Not mentioned	Support Vector Machine and K-Means	30 features
[15]	To investigate the suitability of deep learning approaches for anomaly-based intrusion detection	Not mentioned	Not mentioned	AE (Sparse, Denoising, Contractive, Convolutional), LSTM, CNN	41 features from NSLKDD dataset
[16]	To extending the use of ML in order to improve traffic tolerance for SDNs	Not mentioned	Mininet	SVM, DT, and NB	41 features from MIT Intrusion Detection Evaluation Dataset
[17]	To classify network flows at very high line rates while simultaneously preserving user privacy	Not mentioned	Not mentioned	Decision Trees, Random Forest Tree, Extended Tree, AdaBoost Gradient and AdaBoost	(i) flag: normal/error; (ii) logged in; (iii) count: sum of connections to the same host as the current connection in the past two seconds; (iv) error rate: % of connections that have REJ errors; (v) diff srv rate: % of connections to different services among the same-host connections; (vi) srv diff host rate: % of connections to different hosts among same-service connections; (vii) dst host count: sum of connections to the same host in the previous 100 connections
[18]	To present a management framework to perform anomaly detection, classification, and mitigation jointly	Topology of the Federal University of Rio Grande do Sul campus network	Mininet	K-means, SVM	Not mentioned
[19]	To classify traffic in a QoS-aware manner for SDNs	Not mentioned	N\A (real internet data)	Laplacian SVM, K-Means	10 features
[20]	To study 2 ML algorithms for traffic classification	Not mentioned	Not mentioned	Multilayer Perceptron, Stacked Autoencoder and CNN	Not mentioned

sification performance. They proposed a traffic classification scheme which dynamically chooses and implements the best performing ML classifiers at run-time.

Da Silva et al., [18] presented a framework which detects attack-based traffic, categorizes traffic anomalies by using ML algorithms, e.g. K-means and SVM. After the categorization, they performed particular actions based on the information collected. This mitigation methodology includes actions such as preventing the traffic of suspicious flows.

Wang et al., [19] proposed a scheme for SDN which categorizes traffic in real-time based on classes with different QoS conditions. In order to achieve higher accuracy in classification, DPI and semi-supervised ML; i.e., Laplacian SVM and k-means approaches, were used together.

Wang et al., [20] designed a scheme for aiding the SDN controller in real-time traffic monitoring and classification of encrypted traffic flows by implementing MLP, stacked AE, and CNN. They first preprocessed the data and the resulting meta data was fed back to create the proposed framework.

B. Traffic Prediction

The goal in traffic prediction is to forecast future congestion in the network using historical and/or real-time traffic data. In addition to traffic classification, traffic prediction also plays a significant role in analyzing the traffic flow to prevent traffic getting congested. The predictability of network congestion is desired in order to provide and maintain high quality network communication since based on the outcome of the analyzed traffic data, the SDN controller directs the flows to the less congested links. Table III enlists the surveyed papers discussed below.

In [21], a new deep learning algorithm was proposed to predict prospective traffic load and congestion in the network. At first, a basic deep belief architecture and the deep CNN was used in the training process. Then DL based prediction algorithm was coupled with a DL based channel assignment algorithm to intelligently route the traffic.

[22] focused on Bayesian ML algorithm to enable the switches in SDN to determine the controller creating the flow rules and anticipate the unmatched packets in the flow table.

Jain et. al, [23] followed a methodological approach for providing higher QoS in SDN. The approach had the following steps: exploring correlations within data using big data analytic methods, analyzing root causes behind the problems discovered, and finally based on the outcomes making predictions and analyzing the future trends in the network.

Mestres et. al, [24] studied neural networks to model the delays in the networks. They trained various neural network architectures under various scenarios with different essential network parts such as topology, network size, traffic intensity and routing in order to formulate instructions about training such neural networks.

V. Challenges and Future Research Directions

The major obstacles researchers face when training NN are centered around the data itself. The limited availability of labeled data decreases the accuracy in classification and limits the choices of algorithms since DL techniques often require large amount of data for training. Finding a way to combine unsupervised learning with supervised learning to teach NN how to learn with fewer data is a promising area of research. Furthermore, teaching NN to accumulate its knowledge will make it more effective and efficient in learning new things, and thus, less data will be required for training.

A. Traffic Classification

Network traffic involves encrypted/encapsulated flow packets which hide the features of the flows. Classifying such traffic requires advanced DL methods that can reveal hidden patterns.

The evolution in the networking architecture has brought flexibility and extensibility. However, the decoupling of data and control planes has also made the network more prone to security issues. For example, since the network is managed by a single controller, overloading it with malicious flows creates a challenging problem. To address this problem, DL algorithms can be used more often in detecting suspicious flows and anomaly based attacks.

B. Traffic Prediction

Traffic prediction is necessary in providing high quality communication over the network. Forecasting possible congestion will enable a solution to be offered before QoS/QoE drops. RNN can be applied for prediction analysis since it will use historical data to make better decisions and therefore achieve higher accuracy. Additionally, foreseeing a possible elephant flow occurrence at unusual times which can most probably be labeled as a flow-based intrusion, will provide a more secure network. In addition to that, the prediction of such elephant flows can also eradicate the risk of overburdening the controller in SDN.

Moreover, traffic prediction will enable determining the possible congestion on the links before they lower the QoS & QoE and route the traffic to the less congested links. Exploiting DL algorithms for this manner can make the routing process more intelligent and autonomous, therefore sophisticated enough for SDN. Hence, routing optimization with DL is a significant research problem.

VI. Conclusion

In this article, we surveyed the ML and DL methods used for classification and prediction in SDNs. First, we explained the algorithms and the SDN architecture. Next, we summarized the existing works. We then presented our survey and finally we addressed the challenges and future work that are dataset characteristics, data volume, methodology of applying DL, security related issues due to SDN structure, and flow encryption. Since employing ML and DL algorithms for classification and prediction in SDN is quite new, more problems might be identified in practice that cannot be predicted now.

Acknowledgment

We thank Sa'di Altamimi for his valuable contributions to this work.

TABLE III
TRAFFIC PREDICTION IN SDN

Ref.	Objective	Network Topology	Simulator	ML/DL Technique	Data Features
[21]	To design a novel deep learning-based prediction algorithm to forecast future traffic load and congestion in network	fixed or dynamic topology network	C++/WILL	Deep-CNN	traffic load, hop count, interference factor
[22]	To overcome latency and overheads in SDN	three-tier Clox	ns-3	Bayesian ML	(i) time of PACKET_IN control message sent from a switch to controller; (ii) queuing&processing delay at controller; (iii) time of PACKET_OUT control message sent from controller to switch; (iv) time for switch to process PACKET_OUT message; (iv) time for dest. to receive the packet
[23]	To improve the management of QoS in SDN by discovering new patterns, finding root cause analysis, and predicting traffic congestion	hierarchical model of core	Mininet and POX	Spearman's Algorithm, Linear Regression, M5Rules	Not mentioned
[24]	To model delays using neural networks	unidirectional ring, star, and scale-free networks	Omnet++	ANN, Polynomial Regression	NN hyper-parameters: (i) number of hidden layers; (ii) number of neurons per layer; (iii) the activation function; (iv) the learning rate and the regularization parameter

REFERENCES

[1] X.-Y. Yang, J. Liu, M.-Q. Zhang, and K. Niu, "A new multi-class svm algorithm based on one-class svm," in *International Conference on Computational Science*. Springer, 2007, pp. 677–684.

[2] L. Gómez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, "Semisupervised image classification with laplacian support vector machines," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 336–340, 2008.

[3] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, and S. Shenker, "Software-defined internet architecture: decoupling architecture from infrastructure," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. ACM, 2012, pp. 43–48.

[4] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Intelligent design enables architectural evolution," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. ACM, 2011, p. 3.

[5] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama *et al.*, "Onix: A distributed control platform for large-scale production networks." in *OSDI*, vol. 10, 2010, pp. 1–6.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[7] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, 2018.

[8] J. Yan and J. Yuan, "A survey of traffic classification in software defined networks," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2018, pp. 200–206.

[9] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, pp. 1–9, 2018.

[10] P. Xiao, W. Qu, H. Qi, Y. Xu, and Z. Li, "An efficient elephant flow detection with cost-sensitive in sdn," in *2015 1st International Conference on Industrial Networks and Intelligent Systems (INISCom)*. IEEE, 2015, pp. 24–28.

[11] J. Suárez-Varela and P. Barlet-Ros, "Sbar: Sdn flow-based monitoring and application recognition," in *Proceedings of the Symposium on SDN Research*. ACM, 2018, p. 22.

[12] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *2017 Seventh International Conference on Emerging Security Technologies (EST)*. IEEE, 2017, pp. 138–143.

[13] M. Amiri, H. Al Osman, and S. Shirmohammadi, "Game-aware and sdn-assisted bandwidth allocation for data center networks," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2018, pp. 86–91.

[14] Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2017, pp. 1–6.

[15] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48 231–48 246, 2018.

[16] S. Gangadhar and J. P. Sterbenz, "Machine learning aided traffic tolerance to improve resilience for software defined networks," in *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, 2017, pp. 1–7.

[17] L. He, C. Xu, and Y. Luo, "Vtc: Machine learning based traffic classification as a virtual network function," in *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. ACM, 2016, pp. 53–56.

[18] A. S. da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "Atlantic: A framework for anomaly traffic detection, classification, and mitigation in sdn," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 27–35.

[19] P. Wang, S.-C. Lin, and M. Luo, "A framework for qos-aware traffic classification using semi-supervised machine learning in sdns," in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 760–765.

[20] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway," *IEEE Access*, vol. 6, pp. 55 380–55 391, 2018.

[21] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in sdn-iot: A deep learning approach," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, 2018.

[22] A. Baz, "Bayesian machine learning algorithm for flow prediction in sdn switches," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2018, pp. 1–7.

[23] S. Jain, M. Khandelwal, A. Katkar, and J. Nygate, "Applying big data technologies to manage qos in an sdn," in *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 2016, pp. 302–306.

[24] A. Mestres, E. Alarcón, Y. Ji, and A. Cabellos-Aparicio, "Understanding the modeling of computer network delays using neural networks," in *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. ACM, 2018, pp. 46–52.