J Ambient Intell Human Comput
https://doi.org/10.1007/s12652-022-03763-7

ORIGINAL RESEARCH

# Improving detection accuracy of politically motivated cyber-hate using heterogeneous stacked ensemble (HSE) approach

**Nanlir Sallau Mullah**[1,2] · **Wan Mohd Nazmee Wan Zainon**[1]

Received: 8 July 2021 / Accepted: 9 February 2022


**Abstract**
The surge in cyber-hate crimes is largely fuelled by the popularization of social media platforms. On that note, cyber-hate has become an increasing concern for most countries, especially those that are practising democracy. Studies on the influence of social media (SM) on political discourse have now become an important research area due to the rising trends of SM politics. It becomes necessary to address this problem using automated social intelligence. To tackle this concern, the researchers built a novel heterogeneous stacked ensemble (HSE) classifier for detecting politically motivated cyber-hate on Twitter. We constructed a heterogeneous stacked ensemble with eight baseline estimators. In the proposed methodology, the researchers employed TF-IDF for feature vectorisation. The researchers used Twitter API for data scraping to harvest tweets during a gubernatorial election in Nigeria for the training and evaluation of the stacked ensemble model. A total of 15,502 tweets were collected and after some preliminary cleaning, 5876 tweets were manually labelled as hate (1) or non-hate (0). The coded tweets contain 16.87% hate and 83.13% non-hate tweets. This article has three contributions – a critical review of literature on the detection of politically motivated cyber-hate, the building of a new dataset and the proposed stacked ensemble method. Two other public datasets (Kaggle and HASOC) were used to test the performance of our method. The F1-score metric was employed for comparison. Our method is better by 12% on the Kaggle and 4% on the HASOC datasets. We are working on more data for deep learning experiments.

**Keywords** Text categorization · Stacking ensemble · Machine learning · Hate speech · Social media platforms · Political discourse

## 1 Introduction

Social Media Platforms (SMPs) play vital roles in our everyday activities and schedules Hegazi et al. (2021) and political discourse is no exception (Visvizi et al. 2021). However, this comes with some side effects on our democracy. Democracy at varying developmental stages and transitions have proven prone to violence (Birch et al. 2020). In the past few decades, politically motivated cyber-hate has become an increasing concern for most countries across the globe (Ezeibe 2015). Electoral violence is considered a type of political unrest that is mainly characterized by its timing and intent (Fjelde 2020). Cyber-hate and cyberbullying are common electoral violence perpetrated during electioneering on different SMPs (Adum et al. 2019).

The election campaigns in most cases are the "trigger" event that usually leads to incidents of hate speech and other online abuses (Rosenzweig 2015). In some cases, the incidents may degenerate into physical violence among supporters of different political parties. Some of the most

✉ Nanlir Sallau Mullah
mullakns@gmail.com

✉ Wan Mohd Nazmee Wan Zainon
nazmee@usm.my

[1] School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia

[2] Federal College of Education Pankshin, PMB1027 Pankshin, Plateau State, Nigeria

patronised SMPs globally include Twitter, Facebook, Instagram and YouTube. Politicians, electorates and political parties usually make use of SM for different purposes such as election campaigns and political discourse among others (Goldwasser 2021).

Advancements in the applications of machine learning algorithms (MLAs) in texts classifications have paved the way for automated detection of SM abuses in tweets related to elections (Laaksonen et al. 2020). With these rising cases of cyber-hates and other discriminatory messages, the need to harness the power of MLAs effectively and efficiently is of significance. The arts domain has conducted pieces of research for hate speech identification for many decades, nonetheless the problem still lingers. All hopes are on computing power and machine learning algorithms to resolve this problem.

To monitor these SM abuses manually is greatly inefficient, error-prone and tasking. Therefore, using artificial intelligence (AI) such as ML is the best option available. Applying MLA to detect hate speech on Twitter during an election will be timely and can stem any physical violence that may occur due to cyber-hate spread. In this study, the researchers built a robust heterogeneous stacked ensemble approach (HSE) for cyber-hate detection on Twitter. As a new field of research in the computing domain, most researchers concentrated on forecasting elections results, such as Chauhan et al. (2021) and little or no research on the content of texts in circulation during the 'trigger' event, election. Our contribution is meant to fill this research gap.

This paper added three contributions to the domain of SM analysis and opinion mining. First, to review related literature on politically motivated hate speech detection using MLAs. Furthermore, to build a novel HSE model for detecting politically motivated cyber hates on Twitter. Lastly, to collect and annotate a new dataset for training the HSE algorithm for cyber-hate and other discriminatory abuses' detection during elections on Twitter.

The remaining sections of this paper are organised as follows: Section 2 covers related works about the detection of politically motivated cyber-hate on Twitter using the text mining approach. In Section 3, 4 and 5, we discussed the data scraping, justification for adopting binary classification and the proposed methodology architecture. Section 6 covers experimentation, while Section 7 deals with the result, discussion and comparison with state-of-the-art. The last section covers the conclusion and future works.

## 2 Related works

This section of the article introduces a summary of critical analysis of previous works. The section analysis and synthesis other researchers' works that are closely related to this study. Many researchers have conducted studies on cyber-hate detection on SM and more are ongoing because the problem persists (Mwadime et al. 2020). Therefore, there is a need to improve the model's detection accuracy through an ensemble method.

### 2.1 Critical analysis of the closely related works

There are limited works on the detection of hate speech in political debate on SM (Guellil et al. 2020). To the best of our knowledge, we found five closely related articles – (Aggrawal 2018; Gorrell et al. 2018; Guellil et al. 2020; Ratkiewicz et al. 2011; Stambolieva 2017). We critically analysed these five articles to avoid duplication of efforts in solving this problem.

Guellil et al. (2020) investigated Arabic and Arabizi hate texts on Youtube that targeted politicians. This study leveraged both canonical ML and deep learning (DL) approaches for their simulation experiments. The ML algorithms used are Support vector machine (SVM), Logistic Regression (LR), Gaussian Naïve Bayes (GNB), SGD Classifier (SGD) and Random Forest (RF). The DL algorithms used are multilayer perceptron (MLP), convolutional neural network (CNN), bi-directional long- or short-term memory (Bi-LSTM) and long- or short-term memory (LSTM). FastText and Word2vec were implemented using Continuous Bag of Word (CBOW) and Skip Gram (SG). 5,000 comments were collected from Youtube and were annotated or labelled as hate or no hate. The dataset was balanced and an accuracy of 91.0% was achieved using the DL models. The main contribution of this work was the building of a new dataset using Arabic and Arabizi comments for political hate on SM.

Aggrawal (2018) employed Rule-Based Naive Bayes (RNB) and collaboration between LDA with Naïve Bayes (LDANB) to investigate hate speech in the US political scene. The researchers collected forty thousand tweets that relate to the then-president Donald Trump. With the following hashtags "#antitrump", "#notMyPresident" and "#dumbtrump". Another 1,00,000 tweets were collected with keywords "trump" and "Donald trump". For contextual feature-based, cosine similarity, which was based on document similarity between two documents were used. And the adjective based approach was based on part of speech that was applied.

In 2017, a study by Amnesty International used the 2012 Kaggle dataset to train their sentiment analyser which was labelled as hate speech or no hate speech (Stambolieva

2017). Within six months, 2.85% out of 900, 2233 tweets sent to women MPs were identified as abusive. The research also found out that 64% of the tweets are true-positive. NB was used as the classifier and was able to achieve an accuracy of 34%. The researcher recommended the use of n-gram to train NB to improve the detection accuracy.

Gorrell et al. (2018) in their research, collected 1.4 million tweets between 2015 and 2017 before UK general election. This data was used to study the harassment aimed at politicians. The outcomes revealed that abuse rose dramatically in 2017 as compared to 2015. A dictionary-based technique was employed to identify hateful slurs in the tweets.

In an attempt to detect and track political abuse on Twitter during the 2010 U.S. midterm elections, Ratkiewicz et al. (2011) researched that effect. The following features were used to identify memes in the tweets: #hashtags, mentions, URLs, and Phrases. Google-based Profile of Mood States (GPOMS) sentiment analysis modified method was used for the analysis of the memes. This research also made use of an automatic binary classifier called Truthy. This Truthy labelled the data as truthy or legitimate. Finally, SVM and AdaBoost were trained by the labelled data for the automatic detection of the memes. An accuracy of 96.4% and 95.6% were achieved for AdaBoost and SVM respectively.

Most of these works focused on the western region of the world and only one focus on the middle east. All the closely related works do not make use of any ensemble approach for the simulation works. Therefore, to fill this methodological gap, we intend to apply a stacking ensemble for our experiment. We also collected data from Africa (Nigeria) to explore the new region. We go for the HSE technique to leverage the advantages contained in the different heterogeneous baseline models which have been tested in the past works for text classification tasks.

## 3 Data scraping

For this study, the Twitter platform was used as the source of data collection. A message post on Twitter is called a tweet, these tweets are the data of interest in this research. We collected the data that relate to political discourse in Nigeria. We decided to use Nigeria data Twittersphere because it is highly polarised during every election. This makes Nigeria political discourse on Twitter a fertile ground for research.

We scraped tweets officially using Twitter application programming interfaces (APIs). For the collection of tweets that relate to political discourse in Nigeria, we used hashtags that are commonly employed to reference or link a discussion on Twitter within the election period. Hashtags is simply a word or phrase preceded by a hash sign (#). Hashtags are generally used on SM to link or join a discussion. Hashtags help us to pull all discussions about a particular topic.

A gubernatorial election was conducted on September 19, 2020, in Edo State, Nigeria. Tweets with hashtags regarding this election were the focus of the data collection. We employed the following trending hashtags in our data scraping on Twitter: #EdoElection, EdoDecides, #APC #PDP #Obaseki, #Godwin, #Osagie #Ize #Iyamu. We used Twitter API for data scraping. A total of 15,502 tweets were harvested based on the hashtags stated above. We also removed any duplicate tweets found. After the sorting, we were left with a total of 5876 tweets. We also carry out some preliminary cleaning of the tweets so that annotators can read with ease.

To validate the work of the three annotators, it is necessary to compute the inter-annotator agreement (Krippendorff 2011). To do this, we used Krippendorff's alpha coefficient which was also used in (Burnap et al. 2015). This is a well-established statistical measure of agreement among independent annotators where a score of 0 implies no agreement and a score of 1 shows complete agreement. It is commonly employed by experts in the domain of content analysis Gwet (2015), therefore, is very suitable for our study. The instruction for the annotation was simple and clear. Most of the annotators agreed with each other in most instances and the Krippendorff alpha was computed as 0.847. This is also true that Nigerians are generally blunt and straightforward in their communications. This value is considered acceptable.

We applied the simple 'hard voting' principle to get the final label for each class. This means the majority label is final. Below is the summary of the labelled dataset and their respective class distribution as shown in Table 1:

## 4 Justification for adopting binary classification approach

We decided to use the binary classification technique to classify a tweet as hate speech or no hate speech for the following reasons:

- To narrow the focus for a better training process for our proposed model. For example, making it a multi-classification approach like hate, offensive and neither, will slightly complicate the learning process. The primary aim of the research is simple, if a tweet is marked as

**Table 1** labelled dataset and class distribution

| Label | Tweets | Percentage (%) |
|---|---|---|
| Hate (0) | 991 | 16.87 |
| Non-Hate (1) | 4885 | 83.13 |
| **Total** | **5876** | **100** |

hate speech during an election, should not be allowed into the public domain.
- To maximise the detection process in the two-class distribution. This model will focus on the two classes and improve the detection of tweets as hate speech or no hate.
- All offensive tweets which could lead to physical violence is a potential threat, hence considered here as hate speech. This will certainly give us peaceful elections if the model is deployed on SM during the election the world over.

## 5 Proposed methodology

Offensive text detection generally is modelled as a text classification problem. Every text classification task must conform to these four basic pipelines – from data collection/scraping, data pre-processing, feature extraction, classifier choice and training to model evaluation. The conceptual difference is always obvious in the choice of classifier and the arrangement to maximise the model's performance. The proposed methodology architecture is depicted in Fig. 1.

Figure 1, the proposed methodology has the following sections: Data source, data pre-processing, feature extraction, stacking ensemble and model evaluation. The novelty is in the stacked ensemble for hate speech detection. Let us briefly explain each section of the methodology in the following subsections.

### 5.1 Data source and justification for using the Twitter platform

There are many freely available datasets, but we could not adopt anyone because none is suitable for solving our problem, a Nigeria case. Two conditions are important in adopting a dataset – availability and relevancy (Mullah and Zainon 2021). There are numerous SMPs out there such as Facebook[1], YouTube[2], Instagram[3], but Twitter is the most favoured by researchers for the following reasons:

- Tweets are mostly publicly available, topically arranged and Twitter data can be accessed programmatically using Python code and Twitter API among others. Twitter allows tiered access to its data and is free to academic researchers. Twitter is also very flexible in terms of ways of collecting the data unlike other SMPs like Facebook or Instagram which are quite stricter.
- The availability of the diverse type of data with fewer restrictions. Diversity is possible because it is the most open type of SMP conversation given its non-reciprocal relationship system (Burnap and Williams 2016). Anonymous can join any discussion by using a hashtag and these properties made twitter a fertile ground for propagating hate speech.
- No restrictions on which topic to discuss, it is an open-ended platform. However, users are not allowed to publish discriminatory messages, though persist. And the length of the text should not be more than 280 characters.
- Real-time data can easily be collected from the Twitter[4] website and many researchers leverage this flexibility to obtain data for their pieces of research.
- Twitter is universally used and the second most patronised SMP after Facebook. This makes it suitable for research, for we can get information on any aspect of human activities anywhere that interests us at any time.
- Twitter has been adopted by many countries as an official medium for passing official announcements or messages to the public. For instance, the Federal Government of Nigeria has an official Twitter handle (@NigeriaGov).

### 5.2 Data pre-processing and exploration

Data pre-processing generally involves cleaning the dataset to be free of noise. We carried out the following pre-processing steps: we expand contracted text to their full words, remove stop words, remove all the special characters, remove all single characters, remove hyperlink or URL, substituting multiple spaces with single space, remove prefixed 'b', remove hashtag, @user, link of a tweet, remove cashtags (naira or dollar or euro or pounds signs). We also manually check if our dataset contains numbers such as '419'. The usage of this number has a special connotation in the Nigerian context, which simply means an unwholesome

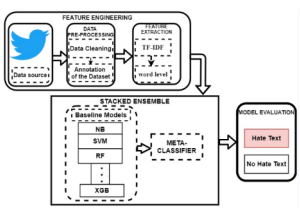


**Fig. 1** Proposed Methodological Flow

[1] https://www.facebook.com/.
[2] https://www.youtube.com/.
[3] https://www.instagram.com/.
[4] https://twitter.com/.

act (Mullah and Zainon 2021). After removing the stop words and other non-English words, we explore the dataset for some observations such as words counts and relationships as shown in Figs. 2 and 3.

From Fig. 2, the word 'election' was mentioned in the dataset more than any other one and the word 'result' was least in the first 25 words plotted. That means people discussed more using the word 'election' than other words in the dataset. Using Fig. 3, we can visualise some text relationships and structures. For instance, we can see that some obvious relationships in words such as 'primary school', 'breaking news' 'governorship election', 'local government', and 'polling unit', form common pairs. Some simple phrases such as 'accredited registered voters' and 'esan north east lga'. All the relationship displayed among the words is meaningful. It means our dataset is cleaned and sensible.

After the cleaning process and exploration, the data is ready for transformation into the computer-readable format, feature extraction.

### 5.3 Feature extraction

All machine learning libraries are built to handle numerical variables only. Therefore, categorical data cannot be processed directly by any learning algorithm in its original form for building a model. Therefore, we need to transform our dataset into a structured and numerical dataset to be used for ML training. This section of feature engineering in our methodology pipeline enables us to transform the textual data into numerical data acceptable as input by our MLAs into the decision vector space. This section deals with engineering the feature in the text's dataset for machine learning usage.

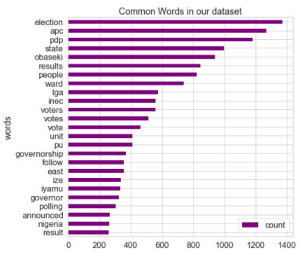


**Fig. 2** First 25 most used words in our dataset

**Fig. 3** Word relationship in the dataset

A feature of text data is the measurable properties or characteristics of the text data under consideration. Feature extraction can be seen as recreating a subset of the feature set with new features but maintaining the characteristics of the original dataset. Feature engineering of texts data is the classification of text data by the domain experts or feature extraction algorithm into a predefined class. Feature extraction is the most important aspect of the text classification task (Hussain et al. 2019). There are many feature extraction techniques in use today. The feature extraction process takes pre-processed or cleaned data as input, extract features and map the features into the decision vector space.

To feed the extracted feature into the text classifier, the text is first transformed into a feature vector space model. Each dimension of the vector model represents an extracted feature in the SM. Some examples of these feature extraction methods use in text-based classification tasks are Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Contextualised word representation, Word2Vec, Global Vectors for Word Representation (GloVe), and FastText (Kowsari et al. 2019). The TF-IDF is the most used feature extraction technique. For this research, we employed TF-IDF for the feature extraction of our Twitter dataset.

TF-IDF concept was first introduced in Salton and Yang (1973) and it is an acronym for Term Frequency (TF) Inverse Document Frequency (DF). TF-IDF weighting is applied in the domain of information retrieval and text mining to determine the weight (importance) of a term in a dataset. In its simplest form, it means how important a word is, in a given corpus or dataset. The relevancy or importance of each term or word x rises proportionately to the number of its occurrence in document (tweet) y. However, it is offset by the frequency of the term in the dataset. Each word or term has a value for TF and IDF. The weight of a term is the product of the score of TF and IDF. The larger the weight of a term, the rarer and more important the word is, and vice versa.

Given a collection of terms, x X that appear in a set of n documents y Y, with a total length of ny, the weighting $W_x$ (TF-IDF) is calculated thus (Yahav et al. 2019). This TF-IDF has two parts, TF and IDF. Let us first start with TF. The Term Frequency $(\mathrm{TF}_{\mathrm{x,y}})$of the term x in document y, can be computed by Eq. 1:

$$\mathrm{TF}_{\mathrm{x,y}} = nxy/ny \tag{1}$$

Where:

x represents a term in document (tweet) y.

nxy is the number of counts of term x in document y.

ny is the total number of terms in document y.

Now, let compute IDF. Let Y represents all documents in the corpus, $\mathrm{Y}_x$ represents all documents that contain x. To

compute the inverse document frequency $\mathrm{IDF}_x$ of the term, x is shown in Eq. 2.

$$\mathrm{IDF}_x = Y/\mathrm{Y}_x \tag{2}$$

The Eq. 2 becomes $\log(Y/\mathrm{Y}_x)$. The log introduced is to reduce or dampen the effect of $Y/\mathrm{Y}_x$ which may be extremely large.

That is;

$$IDF_x = \log\left(Y/\mathrm{Y}_x\right) \tag{3}$$

The weight ($W_x$) or (TF-IDF) of a term x is the product of $\mathrm{TF}_{\mathrm{x,y}}$ and $IDF_x$, see Eq. 4:

$$W_x = \mathrm{TF}_{\mathrm{x,y}} * IDF_x \tag{4}$$

## 5.4 Ensemble techniques for hate speech detection

In the text classification task, the main essence of building models is to categorise texts into a predefined class label. This is very common in social computing research, and it helps researchers to understand the social interactions, emotions, beliefs and the like among participants in SMPs. The choice of which ML algorithm to use is a daunting task. There are many ML algorithms out there, therefore, to choose the most optimal is a huge task. Making the choice is difficult because each algorithm has its share of pitfalls and strengths. Some are obvious while others are latent.

In recent years, ensemble methods are receiving considerable attention due to the quest for a better model's accuracy (Divina et al. 2018). Ensemble techniques leverage the advantages of weaker learning algorithms to create an enhanced and powerful final classifier. The ensemble model has proven to be better than any single learning algorithm which was one of the baseline models.

Classification and regression are all ideals for ensemble approaches that are both for nominal variable prediction and numeric variable prediction problems respectively. Ensemble models generally reduce bias and variance to boost the model's detection accuracy. The evolving ensemble methods have been seen by many researchers as a significant milestone for enhancing ML performances (He et al. 2018). The commonly used ensemble approaches are boosting (Schapire 1990), stacking (Wolpert 1992) and bagging (Breiman 1996) for classification tasks. The brief explanation of each ensemble type is summarised in the following subsections.

### 5.4.1 Boosting ensemble

Boosting ensemble is a learning technique that utilises the error made by the previous baseline model to improve its accuracy sequentially. Just like any other ensemble technique, it combines several weaker baseline algorithms to form a final powerful model, hence enhancing the general performance of the ensemble model (Rong et al. 2020). The baseline models in boosting ensemble are arranged in sequence, in a way that each model learns from the predecessor and at the end create a better model. There are different types of boosting - gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost.

Boosted model is much similar to the bagged model, with little conceptual modification. Instead of considering all models output equally, different weights are assigned, and weighted voting is applied to determine the final output (Divina et al. 2018). For regression, the weighted average is used. More details and mathematical descriptions of this method can be found in (Schapire 1990).

### 5.4.2 Stacking ensemble

Stacking ensemble is also called stacked generalization. The stacking ensemble method can be described as a 2-stage approach (Fatemifar et al. 2020). The first is where the baseline algorithms are trained, then the output of the first stage serves as the input of the second stage. Stacked models are usually built by integrating and implementing all the baseline algorithms in parallel and one algorithm called the meta-learner or combiner algorithm. The meta-classifier is trained using the output of the baseline models to make its final prediction. Any suitable linear algorithm can be used as the combiner. More details and mathematical descriptions of this method can be found in (Wolpert 1992).

Since the creation of this technique, an improvement has been made. The improvement was made through the incorporation of cross-validation to solve the problem of overfitting observed with the technique. The algorithmic description is given in Algorithm 1.

### 5.4.3 Bagging ensemble

Bagging ensemble is short for bootstrap aggregating and was first proposed in (Breiman 1996). It mixes bootstrapping and aggregation to generate a customized and better-bagged model. It utilises decision trees algorithms to create different weaker baseline models and these baseline models will be aggregated to form a final bagging classifier. The benefit of a bagging ensemble classifier is that the model can help in reducing the variance in the baseline models and in doing so, it can also fine-tune the baseline models to the expected result, hence enhancing the final bagged model's accuracy (Dou et al. 2020). The variance reduction helps in improving the model accuracy and consequently eliminating overfitting, which is a common problem in most models. In the bagging scheme, two or more models are generated, the output of these baseline models are considered equally,

then voting is applied to identify the majority output decision (Divina et al. 2018). But if the problem is a regression task, the averaging method is invoked for the final decision. More details and mathematical descriptions of this method can be found in (Breiman 1996).

Real-life problems usually suffer from imbalanced class distribution problems, for instance, the hate speech dataset. To handle the imbalanced datasets problem, ensemble techniques have been employed with great success in diverse applications. Ensemble learning approaches have proven to be efficient tools for building classifiers and have also shown to have outperformed any standalone or single classifier. Of course, the traditional ML algorithms may not give us the much-desired result, therefore, the methods for overcoming this shortcoming must be applied, ensemble approach.

For this research, we desire to fill the gap of the non-usage of ensemble approach for politically motivated hate speech detection. We will be experimenting with stacking ensemble methods. We made use of an improved stacking ensemble that uses the K-fold cross-validation technique for the research experiment. The procedure for this method is shown in algorithm 1 (Wolpert 1992).

**Algorithm 1** Stacking Ensemble with K-fold cross-validation.

| S.N | Stacking Algorithm (Cross-Validation (CV)) |
|---|---|
| 1 | **Input**: Training dataset G = $(X_i, y_i)_{i=1}$ ($X_i$ R, $y_i$ Y) |
| 2 | **Output**: Stacking Classifier, S |
| 3 | Apply cv to train the baseline models and use the output to train meta-classifier in the second-level |
| 4 | Split G randomly into K-fold subsets G = ($G_1$, $G_2$, $G_3$, … $G_k$) |
| 5 | **for** K ←1 to K **do** |
| 6 | Step1.1: train the baseline models |
| 7 | **for***i* ←1 to I **do** |
| 8 | Learn a classifier $h_{k1}$ from G/$G_k$ |
| 9 | **end for** |
| 10 | Step1.2: generate training dataset for meta-classifier in the second level |
| 11 | **for** $X_i$ $D_k$**do** |
| 12 | Get a record {$X_i$', yi}, where$X'_i$ = {$s_{k1}(X_i)$,$s_{k2}(X_i)$, …, $s_{kT}(X_i)$} |
| 13 | **end for** |
| 14 | **end for** |
| 15 | Step 2: train the meta-classifier |
| 16 | Learn a new classifier s' from the collection of {$X'_1$, $y_i$} |
| 17 | Step 3: Re-learn first-level classifiers |
| 18 | **for***i* <--1 to I **do** |
| 19 | learn a classifier $s_t$ based on G |
| 20 | **end for** |
| 21 | **return** S(X) = s'($s_1$(X)), $s_2$(X), …, $s_T$(X)) |

Ensemble approaches are good for lowering the variance in models, hence enhancing the accuracy of the classifiers. On the other hand, the cross-validation introduced in the stacking ensemble is to help in reducing the overfitting in the baseline models. The researchers used stratified 10-fold cross-validation to test the proposed method on binary classification problems and the results are superb. We tested the robustness of our method on three datasets. And stratified K-foldCV was used to address the problem of skewed class distribution to help the model learn every class's property equally. That means every fold contains the same class distribution as hate and non-hate as in the original data class distribution.

### 5.5 Model performance evaluation

Model evaluation is one of the critical steps in building learning models (Brownlee 2019). To evaluate simply means to measure or test how a trained model is performing with the help of some evaluation metrics. There are different ways to achieve this. One thing that is common to all is that the model will be exposed to the data which the model has not seen during the training process to evaluate its effectiveness.

The ability to understand exactly what each evaluation metric represents for any classification algorithm is crucial to understanding the model performance. Performance evaluation is a common technique employed in most research domains to evaluate the effectiveness of a particular method. Different evaluation metrics are available depending on the research and method used.

Performance metrics are logical-mathematical constructs generated through the difference between the ground truth values and the predicted values by the classifier. Mostly used examples include Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) for a regression problem. Other evaluation metrics include recall, precision, accuracy, F-score, Matthews Correlation Coefficient (MCC) Receiver Operating Characteristic (ROC) and area under the curve (AUC) for a classification task. Recall, precision, accuracy, and F-measure are mostly employed for classification (Zhang and Luo 2018). ROC and AUC are also good for evaluating models however, the metric is rarely used. We will use recall, precision, accuracy, F1-score, MCC, ROC and AUC for our evaluation. The most important of these metrics in our case is MCC and F-score, because of the skewed class distribution in our dataset (Davidson et al. 2017; Jurman et al. 2012).

To compute the classification metrics, the following are required: true-positive (TP), false-positive (FP), false-negative (FN) and true-negative (TN), of which all are obtained through a confusion matrix. The summary of the interpretation of the metrics is given in Table 2.

The following evaluation metrics: precision, sensitivity, F1-score, accuracy and specificity can be computed as

shown in Eq. 5, Eq. 6, Eq. 7, Eq. 8 and Eq. 9 respectively as follows:

**Precision ($P_r$)**:

$$P_r = \frac{TP}{TP+FP} \tag{5}$$

**Sensitivity or Recall ($R_c$)**:

$$R_c = \frac{TP}{TP+FN} \tag{6}$$

**F1-score (F1)**:

$$F_1 = 2 * \frac{P_r * R_c}{P_r + R_c} \tag{7}$$

**Accuracy (A)**:

$$A = \frac{TP+TN}{TP+FP+FN+TN} \tag{8}$$

**Specificity ($S_t$)** This is also called the true-negative rate. It refers to how a particular model identifies true-negative samples as negatives in the predicted result. Equation 9 shows the specificity relationship in terms of TN and FP.

$$S_t = \frac{TN}{TN+FP} \tag{9}$$

#### 5.5.1 Matthews correlation coefficient (MCC)

MCC is seen as a more comprehensive evaluation metric for a dataset with skewed class distribution, especially in binary classification tasks (Jurman et al. 2012). This evaluation measure comprehensively captures confusion metrics where other metrics are based. MCC value lies between [-1, 1], where 1 is the ideal situation or perfect classification in which both FN and FP are zero (0). When the MCC is -1, it means an extreme misclassification case. That is both TP and TN value is zero (0). When the value of MCC is zero, it indicates that the model is classifying randomly, that is it cannot differentiate between the two classes. Mathematically, MCC is computed using Egn 10.

MCC =.

$$\frac{(TP*TN)-(FP*FN)}{(TP+FP)(TP+FN)(TN+FP)(TN+FN)} \tag{10}$$

Many researchers have used this evaluation metric for testing the performances of some models. These studies include (Feng et al. 2018) and (Rao and Pais 2020).

**Table 2** Evaluation metrics

| Evaluation metric | Interpretation |
|---|---|
| TP | The classifier predicts the results as positive and final confirmation shows it is correct predictions. |
| FP | The classifier predicted the result as positive, but the correct value is negative. |
| TN | The algorithm correctly predicted negative value and confirmed it as negative |
| FN | It was predicted as negative, but the actual value is positive |

#### 5.5.2 ROC and AUC

ROC curve as an evaluation metric has been extensively applied in medical research to evaluate classification accuracy (Wang and Cai 2021). This graph is obtained by plotting the TP rate against the FP rate to show the performance of a classifier at different classifications thresholds. ROC curves are normally employed to evaluate the performance of a binary classification model. The coordinate points (1,0), for TP rate and FP rate respectively, is the ideal situation. However, this is not realistic in most real-life scenarios. These values can be computed and plotted by invoking the Scikit-learn library (Pedregosa et al. 2011).

AUC on the other hand is the whole area under the ROC curve[5]. The larger the area, the better the performance of the model. Usually, there is an AUC threshold of 0.5. No model is expected to perform below this threshold. Any model that the AUC is equal to 0.5, means it is confusing and cannot differentiate between the two classes. Every model strives to attain an AUC of 1, best performance. But AUC of 1 is rarely achieved in most cases.

## 6 Experimentation

The primary goal of our proposed method is to train a classifier to classify a political tweet correctly and efficiently as hate speech or non-hate. For this research, we use data collected from the Twitter platform during a gubernatorial election in Nigeria. We applied a heterogeneous stacking ensemble (HSE) technique for the experiment along with stratified K-fold cross-validation. The essence of using the ensemble and stratified K-fold technique include – to enhance the performance and improving the robustness of the detection model. Furthermore, to build a model capable of handling the imbalanced class distribution common in most real-life cyber-hate datasets.

We first selected ten (10) commonly use ML algorithms as baseline models. These are the MLAs used in the articles reviewed in this study. These include Random Forest (RF), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), Decision Tree (DT), Logistic Regression (LR), Gaussian Naive Bayes (GNB), k-nearest neighbour (KNN), Gradient Boosting Classifier (GBC), XGBoost (XGB) and AdaBoost (Adab). After the first run of the experiment with the ten estimators, we obtained some interesting results as detailed in the following section. All simulations are conducted using skit-learn IDE (Pedregosa et al. 2011).

[5] https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html#sphx-glr-auto-examples-model-selection-plot-roc-py.

## 7 Results and discussion

This section is meant to display and discuss the results of our proposed heterogeneous ensemble experiment for detecting politically motivated hate speech on Twitter. The result of the first runs shows that KNN and GNB did not perform well on our dataset as illustrated in Fig. 4. The AUC for GNB and KNN is 0.59 and 0.62 respectively. This means that these models are best described as being confused despite the training, hence not suitable for use as parts of our baseline classifiers. The two algorithms hardly differentiate between hate texts and non-hate texts after the training.

The two algorithms were removed from the baseline models because they add up too little or nothing to the proposed ensemble. This will help us to conserve time and processing power. We now re-run the experiment using eight baselines' models and re-evaluate using ROC and AUC as in Fig. 5.

Viewing Fig. 5, the AUC of the baseline models ranges from 0.90 to 0.96, and these results are reasonable to go with. We use these eight MLAs to build our stacking ensemble. The main aim of this research is to improve the hate speech detection on political discourse on Twitter.

Generally, the TF-IDF technique has been widely applied for feature extraction in text categorisation with excellent results (Zhu et al. 2019). Therefore, the researchers adopted TF-IDF for feature extraction in the proposed method. Stratified 10-Fold cross-validation was used to implement the proposed ensemble technique because of the imbalanced nature of our dataset. We first tested our HSE technique on our new dataset and evaluation using ROC and AUC as depicted in Fig. 6. Each fold performed well, with an average AUC of 0.95±0.02. The value ±0.02 means the AUC ranges between 0.93 and 0.97 The detail of the other classification report is found in Table 3; Fig. 7.

### 7.1 Comparative analysis

For purpose of comparison, we used the two most recent and closely related works by (Yadav et al. 2021) and (Kovács et al., 2021). In Yadav et al. (2021), the researchers used the Kaggle[6] dataset to categorise twitter texts as hate and non-hate texts. Kovács et al. (2021) used HASOC2019 (Mandl et al. 2019) to solve the problem of task_1, categorising text as the "Hate and Offensive" category (HOF) and the "Non-Hate and offensive" category (NOT). We tested our proposed method using the two datasets and compare the performances based on the metrics reported in these articles. Figures 8 and 9 compare the performances between HSE with Yadav et al. (2021) and Mandl et al. (2019) respectively.

Considering Fig. 8, our method outperformed the method used in Yadav et al. (2021) for solving the same problem. Our method is better by 12% using the F1-score. The F1-score evaluation metric has been rated high by researchers for datasets with an imbalanced class distribution (Madichetty et al. 2021; Yadav et al. 2021) used F1-score to evaluate the model performance, therefore, we used the same for ease of comparison.

HASOC2019 dataset is provided with both train and test data. After the training, we tested our model using the test data as did in Mandl et al. (2019). We got a macro-F1 of 0.67. Our model performs better than Mandl's with 0.04 as depicted in Fig. 9.

To test the robustness of our method, we train the proposed ensemble on our new dataset. A more comprehensive evaluation metric was employed for the evaluation, the MCC metric. It is believed that MCC is one of the best evaluation metrics, especially in binary classification and in an imbalanced class distribution dataset (Jurman et al. 2012). The trained model on the new dataset performed well with an F1-score of 96% and an MCC of 0.92. MCC of 0.92 is an excellent performance. The detail of the classification report is displayed in Table 3.

### 7.2 Common machine learning error analysis

There are common errors that normally occur in every machine learning experiment. For example, false-positive and false-negative. False-positive means a model classified a tweet as 'no hate speech' while it is 'hate speech' using the ground truth. In the case of false-negative, a model classified a tweet as 'hate speech' while it is a 'no hate speech' tweet. False-positive and false-negative help us in vetting our model and certifying its effectiveness. Let use the output of our HSE experiment on the new dataset to explain further:

[6] https://www.kaggle.com/arkhoshghalb/twitter-sentiment-analysis-hatred-speech.

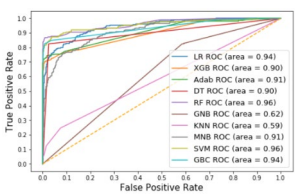


**Fig. 4** ROC Curves for Estimators

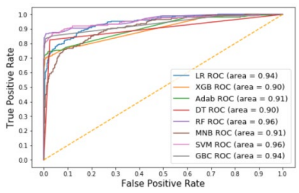


**Fig. 5** ROC Curves for the selected estimators

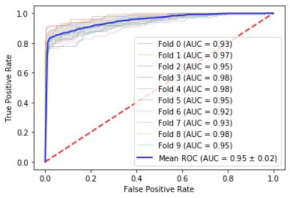


**Fig. 6** The HSE ROC and AUC for stratified 10-fold cross-validation on the new dataset

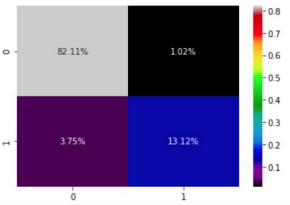


**Fig. 7** Confusion matrix of HSE on the new data

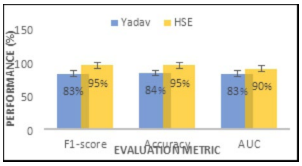


**Fig. 8** HSE vs. Yadav et al. (2021) methods performance comparison on Kaggle dataset

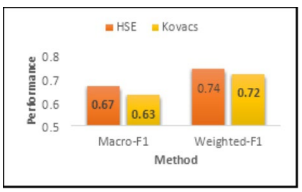


**Fig. 9** HSE vs. Mandl et al. (2019) methods performance comparison on HASOC2019 dataset

**Table 3** Classification report of HSE on our new dataset

| Label | Pr | Rc | F1 | A | MCC |
|---|---|---|---|---|---|
| No-hate (0) | 0.96 | 0.99 | 0.98 | | |
| Hate (1) | 0.95 | 0.80 | 0.87 | | |
| A | | | | 0.96 | |
| Macro avg. | 0.96 | 0.90 | 0.92 | | |
| Weighted avg. | 0.96 | 0.96 | 0.96 | | |
| MCC | | | | | 0.92 |

A tweet identified as 0, meaning non-hate is 96% likely to be correct (i.e. precision). This means there is only a 4% false-positive likelihood in the non-hate class (0). We have a 1% false-negative chance in the positive class with a high recall of 99%. A tweet identified as hate speech is 95% likely to be correct. This implies that there is only a 5% false-negative chance in the hate speech class (1). There are 20% false-negative in the negative class with a recall of 80%. This looks good because the general performance of the model based on the weighted average of F-score is 96%.

Check out the confusion matrix in Fig. 7. Out of 83.13% no hate class, 82.11% were correctly classified by the model. For hate speech class having a class distribution of 16.87%, the model was able to identify 13.12%. The hate class, the minority class is more important to us. If you add up 3.75% and 13.12%, it is exactly 16.87%. 16.87% represent 100% of the minority class. Therefore, 13.12% translate to 77.77%. This means each time the model is 77.77% correct when it classifies a text as hate speech. On the other hand, it is 98.77% correct to detect a non-hate post. This is very reasonable for skewed data of this magnitude.

MCC in this experiment is 0.92 and F1-score is 96. F1-score value is influenced by $P_r$ and $R_c$ directly. But the value of MCC is not directly dependent on the $P_r$ and $R_c$, rather it depends directly on the values of the confusion matrix.

The skewed nature of the dataset was effectively managed by the stratified K-fold cross-validation implemented. The stratified K-fold cross-validation helps to reshuffle the dataset to enable the algorithm to learn all aspects in each class of the training data in equal proportion. That is, in each fold, the training data will contain an equal proportion of hate and no-hate dataset in the percentage ratio, 16.87% hate and 83.13% non-hate.

### 7.3 Dataset availability

The dataset used in this study is available and can be accessed from the corresponding author on reasonable request.

## 8 Conclusion and future works

The main aim of our proposed methodology is to improve the detection of politically motivated cyber-hate on the Twitter platform using a proposed heterogeneous stacked ensemble approach. Ensemble methods are meant to improve the detection accuracy of models in the general sense. However, our work is meant to prove that the HSE approach along with stratified K-fold cross-validation is robust enough for detecting politically motivated cyber-hate on Twitter in the presence of imbalanced class distribution. We have successfully proven this assertion by comparing our work with state-of-the-art results, Mandl et al. (2019) and (Yadav et al. 2021). Our method is better than the work of (Mandl et al. 2019) and (Yadav et al. 2021) with an F1-score difference of 12% and 4% respectively as seen in Figs. 8 and 9.

In our HSE method, $P_r$ is 96% and $R_c$ is 96%, which shows that false-positives and false-negatives are both 4%. In Yadav et al. (2021), $P_r$ was 83% and $R_c$ was 84%, which means the model has 17% false-positive and 16% false-negatives. The difference between our $R_c$ (16-5)% is 11%. That means that Yadav contained more numbers of false-negative, 11% higher than HSE. In the case of Mandl et al. (2019), $P_r$ and $R_c$ values were not reported.

This is ongoing research; we intend to collect more data during any election in Nigeria in the future. When the volume of our dataset improves, we will try deep learning and deep learning ensemble in the future. We also intend to use other different 'trigger' events, besides the election. Thirdly, we recommend that other SMPs such as Facebook, Instagram and WhatsApp can be exploited.

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/s12652-022-03763-7.

**Funding** No funds, grants, or other support was received.

**Additional information** Correspondence and requests for the dataset and any material should be addressed to MN.

## Declarations

**Conflict of interest** The authors declare no competing interests relevant to this article.

## References

Adum AN, Ojiakor OE, Nnatu S (2019) *Party Politics, Hate Speech and the Media: A Developing Society Perspective*. *5*(1), 45–54

Aggrawal N (2018) *Detection of Offensive Tweets: A Comparative Study Niyati*. *1*(1), 1–26

Birch S, Daxecker U, Höglund K (2020) Electoral violence: An introduction. J Peace Res 57(1):3–14. https://doi.org/10.1177/0022343319889657

Breiman L (1996) Bagging Predictors. Mach Learn 24(421):123–140. https://doi.org/10.1007/BF00058655

Brownlee J (2019) *Statistical Methods for Machine Learning Discover how to Transform Data into Knowledge with Python*

Burnap P, Williams ML (2016) Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Science 5(1):1–15. https://doi.org/10.1140/epjds/s13688-016-0072-6

Chauhan P, Sharma N, Sikka (2021) The emergence of social media data and sentiment analysis in election prediction. J Ambient Intell Humaniz Comput 12(2):2601–2627. https://doi.org/10.1007/s12652-020-02423-y

Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, 512–515

Divina F, Gilson A, Goméz-Vela F, Torres MG, Torres JF (2018) Stacking ensemble learning for short-term electricity consumption forecasting. Energies 11(4):1–31. https://doi.org/10.3390/en11040949

Dou J, Yunus AP, Bui DT, Merghadi A, Sahana M, Zhu Z, Chen CW, Han Z, Pham BT (2020) Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed. Japan Landslides 17(3):641–658. https://doi.org/10.1007/s10346-019-01286-5

Ezeibe CC (2015) Hate Speech and Electoral Violence in Nigeria. *Hhate Speech and Electoral Violence in Nigeria*, *July 2015*, 1–35

Fatemifar S, Awais M, Akbari A, Kittler J (2020) A Stacking Ensemble for Anomaly Based Client-Specific Face Spoofing Detection. *Proceedings - International Conference on Image Processing, ICIP*, *2020-Octob*(October), 1371–1375. https://doi.org/10.1109/ICIP40778.2020.9190814

Feng F, Zhou Q, Shen Z, Yang X, Han L, Wang JQ (2018) The application of a novel neural network in the detection of phishing websites. J Ambient Intell Humaniz Comput 0(0):1–15. https://doi.org/10.1007/s12652-018-0786-3

Fjelde H (2020) Political party strength and electoral violence. J Peace Res 57(1):140–155. https://doi.org/10.1177/0022343319885177

Goldwasser D (2021) *MEAN: Multi-head Entity Aware Attention Network for Political Perspective Detection in News Media*. 66–75

Gorrell G, Greenwood MA, Roberts I, Maynard D, Bontcheva K (2018) Twits, twats and twaddle: Trends in online abuse towards UK politicians. *12th International AAAI Conference on Web and Social Media, ICWSM 2018*, 600–603

Guellil I, Adeel A, Azouaou F, Chennoufi S, Maafi H, Hamitouche T (2020) Detecting hate speech against politicians in Arabic community on social media. Int J Web Inform Syst 16(3):295–313. https://doi.org/10.1108/IJWIS-08-2019-0036

Gwet KL (2015) *On Krippendorff 's Alpha Coefficient*. *1971*, 1–16

He H, Zhang W, Zhang S (2018) A novel ensemble method for credit scoring: Adaption of different imbalance ratios. Expert Syst Appl 98:105–117. https://doi.org/10.1016/j.eswa.2018.01.012

Hegazi MO, Al-Dossari Y, Al-Yahy A, Al-Sumari A, Hilal A (2021) Preprocessing Arabic text on social media. Heliyon 7(2):e06191. https://doi.org/10.1016/j.heliyon.2021.e06191

Hussain S, Mufti MR, Sohail MK, Afzal H, Ahmad G, Khan AA (2019) A step towards the improvement in the performance of text classification. KSII Trans Internet Inf Syst 13(4):2162–2179. https://doi.org/ 10.3837/ tiis.2019.04.024

Jurman G, Riccadonna S, Furlanello C (2012) A comparison of MCC and CEN error measures in multi-class prediction. PLoS ONE 7(8):1–8. https://doi.org/10.1371/journal.pone.0041882

Kowsari K, Meimandi KJ, Heidarysafa M, Mendu S, Barnes L, Brown D (2019) Text classification algorithms: A survey. Inform (Switzerland) 10(4):1–68. https://doi.org/10.3390/info10040150

Krippendorff K (2011) Agreement and Information in the Reliability of Coding. Communication Methods and Measures 5(2):93–112

Laaksonen SM, Haapoja J, Kinnunen T, Nelimarkka M, Pöyhtäri R (2020) The Datafication of Hate: Expectations and Challenges in Automated Hate Speech Monitoring. Front Big Data 3, 1–16. https://doi.org/10.3389/fdata.2020.00003

Madichetty S, Muthukumarasamy S, Jayadev P (2021) Multi-modal classification of Twitter data during disasters for humanitarian response. Journal of Ambient Intelligence and Humanized Computing, 1–15

Mandl T, Modha S, Patel D, Majumder P, Dave M, Mandlia C, Patel A (2019) Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In Proceedings of the 11th Forum for Information Retrieval Evaluation, 14–17

Mullah NS, Zainon WMNW (2021) Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review. IEEE Access 9:88364–88376. https://doi.org/10.1109/ACCESS.2021.3089515

Mwadime G, Odeo M, Ngari B, Mutuvi S (2020) *Modeling Hate Speech Detection in Social Media Interactions Using Bert*. *VII*(Ii), 78–81

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grise O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D (2011) Scikit-learn. J Mach Learn Res 19(1):2825–2830. https://doi.org/10.1145/2786984.2786995

Rao RS, Pais AR (2020) Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach. J Ambient Intell Humaniz Comput 11(9):3853–3872. https://doi.org/10.1007/s12652-019-01637-z

Ratkiewicz J, Meiss M, Conover M, Gonçalves B, Flammini A, Menczer F (2011) Detecting and Tracking Political Abuse in Social Media. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 297

Rong G, Alu S, Li K, Su Y, Zhang J, Zhang Y, Li T (2020) Rainfall induced landslide susceptibility mapping based on bayesian optimized random forest and gradient boosting decision tree models—a case study of shuicheng county, china. Water (Switzerland) 12(11):1–22. https://doi.org/10.3390/w12113066

Rosenzweig S (2015) Dangerous Disconnect: How Politicians' misperceptions about voters lead to violence in kenya. *Seasupennedu*, 1–22. http://www.seas.upenn.edu/~eas285/Readings/Hammond_HowPeopleLearn.pdf

Salton G, Yang CS (1973) On the specification of term values in automatic indexing. J Doc 29(July):351–372

Schapire RE (1990) The Strength of Weak Learnability. Mach Learn 5(2):197–227. https://doi.org/10.1023/A:1022648800760

Stambolieva E (2017) *Methodology: Detecting Online Abuse against Women MPs on Twitter*. Amnesty International, 1–20

Visvizi A, Lytras MD, Aljohani N (2021) politics, governance and democracy. J Ambient Intell Humaniz Comput 12(4):4303–4304. https://doi.org/10.1007/s12652-021-03171-3. Big data research for politics: human centric big data research for policy making,

Wang D, Cai X (2021) Smooth ROC curve estimation via Bernstein polynomials. PLoS ONE 16(5):e0251959. https://doi.org/10.1371/journal.pone.0251959

Wolpert DH (1992) Stacked generalization. Neural Netw 5(2):241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Yadav N, Kudale O, Rao A, Gupta S, Shitole A (2021) Twitter Sentiment Analysis Using Supervised Machine Learning…*" In Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020*, *57*(March), 631–642. https://doi.org/10.1007/978-981-15-9509-7_51

Yahav I, Shehory O, Schwartz D (2019) Comments Mining With TF-IDF: The Inherent Bias and Its Removal. IEEE Trans Knowl Data Eng 31(3):437–450. https://doi.org/10.1109/TKDE.2018.2840127

Zhang Z, Luo L (2018) Hate speech detection: A solved problem? The challenging case of long tail on Twitter. Semantic Web 10(5):925–945. https://doi.org/10.3233/SW-180338

Zhu Z, Liang J, Li D, Yu H, Liu G (2019) Hot Topic Detection Based on a Refined TF-IDF Algorithm. IEEE Access 7:26996–27007. https://doi.org/10.1109/ACCESS.2019.2893980