applied sciences

Article

Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with Count Vectorizer

Turki Turki [1,*] and Sanjiban Sekhar Roy [2,*]

[1] Department of Computer Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India
* Correspondence: tturki@kau.edu.sa (T.T.); sanjibansroy@ieee.org (S.S.R.)

Abstract: A plethora of negative behavioural activities have recently been found in social media. Incidents such as trolling and hate speech on social media, especially on Twitter, have grown considerably. Therefore, detection of hate speech on Twitter has become an area of interest among many researchers. In this paper, we present a computational framework to (1) examine out the computational challenges behind hate speech detection and (2) generate high performance results. First, we extract features from Twitter data by utilizing a count vectorizer technique. Then, we provide the labeled dataset of constructed features to adopted ensemble methods, including Bagging, AdaBoost, and Random Forest. After training, we classify new tweet examples into one of the two categories, hate speech or non-hate speech. Experimental results show (1) that Random Forest has surpassed other methods by generating 95% using accuracy performance results and (2) word cloud displays the most prominent tweets that are responsible for hateful sentiments.

Keywords: hate speech; Twitter; vectorization; ensemble methods; word cloud

check for updates

Citation: Turki, T.; Roy, S.S. Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with Count Vectorizer. *Appl. Sci.* **2022**, *12*, 6611. https://doi.org/10.3390/app12136611

Academic Editors: Evgeny Nikulchev and Vladimir Borisovich Barakhnin

Received: 1 May 2022
Accepted: 24 June 2022
Published: 29 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The amount of web data that is available today is considerably larger than it was a few years back. The dramatic increase in the web data, especially in the form of social media such as Twitter and Facebook, has shifted the usability of internet to the next level. In addition to social media, the published contents are also available on the various websites, ecommerce companies, online communities, and various media of collaborative types. The rapid access to the web data on these different platforms has geminated a huge number of topics that are used to draw the attention of significant number of users, aiming to acquire knowledge from such a web information. Mining or extracting meaningful information from such spread and unstructured web data on social media is not an easy task. Social media data from Twitter, Facebook and Instagram were used to reveal a lot about the behaviour of users. This has generated huge interest among researchers, especially in automatic extraction, pre-processing, cognizing the sentiment and finally detecting the overall sentiment of the social media data. Natural language processing (NLP) combined with artificial intelligence and machine learning have been successful to some extent to address these challenges [1–3].

The detection of hateful speech in social media is a difficult task. The uncontrolled use of hateful speech can severely harm our society and certain groups. However, the major place for sharing hateful speech is social media, especially Twitter. Therefore, automatic detection of hateful speech in social network contributes immensely. The detection uses emoticons and hashtags. Understanding the sentiments of the user especially on Twitter or Facebook has been the central research idea and has been the hot area of NLP research in the recent times [4,5]. The first hurdle of hate speech detection is how to define hate speech. Among many social media platforms, hate speech on Twitter is very common and unfortunately widely practiced. Twitter is a defendable and legitimate source of data for analysing the hateful content, and therefore, it is important to find the sentiments of the tweets and to finally detect them in an automatic fashion by proposing machine-learning techniques. Researchers have found promising outcomes for classifying hate speech from the textual information in terms of tweets [6].

In the recent literature, machine-learning algorithms have been proposed to solve the hate speech detection problem and to successfully achieve good performance results. However, existing methods (1) are far from perfect mostly utilizing single classifiers; (2) lack detailed statistical analysis to extract textual insights; (3) miss the representation of hate speech from a visual perspective; and (4) use existing data without sophisticated data augmentation techniques. Therefore, in this work, we use a computational framework incorporating data augmentation techniques to promote better representation, which improves the performance of studied ensemble methods. Moreover, we offer a new way to visualize the hate speech and accomplished a comprehensive study on hate speech detections.

As Twitter is a popular social interaction platform for posting short messages, it accommodates various languages, and the length of the message is also very short, so it is a bit challenging to detect hate speech.

Contributions of this work have been mentioned below, as follows:

1. Our proposed framework can determine the unique features that were extracted from Twitter. The incorporation of count vectorizer library has proven to be a good choice for the feature extraction.
2. We provide word-cloud representation to identify (1) important words and (2) hateful words. Then, we provide statistical information about the tweets in terms of frequency and distribution. Word cloud is very significant in terms of exploratory textual analysis as this can represent the frequently appearing words in the case of tweets. It helps to understand the most salient themes or words that make speech hateful.
3. We construct a corpus of tweets, followed by pre-processing and cleaning of tweets. Then, we construct feature vectors via a text augmentation technique. Finally, the constructed feature vectors along with the labels are fed to ensemble methods for training and predicting hate speech of new tweets.
4. We perform a comprehensive comparative study among the proposed ensemble methods and provide detailed detection report using several performance measures such as accuracy, precision, recall, AUC, and F1 scores.

The rest of the paper is organized as follows. The related work is discussed in Section 2. Our proposed framework is presented in Section 3, followed by experimental evaluation, results, and discussion provided in Section 4. Finally, we conclude the work and point out future research directions in Section 5.

2. Related Work

Recently, researchers have investigated the detection of offensive language based on FCNN classification model as in Hajibabaee et al. [3] as they obtained much better performance in terms of accuracy and F1 score. Machine-learning-based methods have also been proposed by Chia et al. [4] for identifying cyberbullying. They have collected a dataset that was created by Evaluation 2018 Task 3: Irony Detection in English Tweets to detect cyberbullying. Hate speech is one of the most commonly trolling techniques that are used for cyberbullying. This kind of aggression—when it happens—in which a social media platform such as Twitter becomes a complex phenomenon, and many collaborative research fields are trying to solve this critical problem [6]. In the literature, many other works also have tried to address these issues, such as cyberbullying [6,7], trolling [8], extremism [9], hate speech [10,11] and racism [12,13]. There are different approaches that could be applicable for feature engineering, including N-gram to generate the vectors from dictionary of hate-related words [14–16]. Another important part of hate speech detection is vectorization, and this can be achieved with the use of TFID that decides the importance of the words with weight attached to each word, and the other one is count vectorizer. In addition, a bag-of-words approach of natural language processing is used for information retrieval, and it is usually used for document representation [17,18].

In Table 1, we provide a summary of several studies about hate speech detection by various authors. The columns in this table demonstrate the proposed models by the authors, representation of features, outcomes, limitations or gap, and dataset. It can be shown from Table 1 the methods, limitations as well as the used performance measures, including F1 score, accuracy, precision and recall when reporting performance results.

Table 1. Studies of hate speech detection.

Authors	Model/Models	Representation of Feature	Outcomes	Limitations or Gap	Dataset
Araque and Iglesias [1]	TF-IDF and SIMilarity-based sentiment projectiON (SIMON)	TFIDF	F1-scores	Could have tried combination method	Pro-Neu, Pro-Anti, Magazines, SemEval-2019, Davidson
Heidari et al. [3]	Feed Forward Neural Network (FFNN) and Logistic Regression (LR)	Word embedding	Accuracy, F1 score	No of evaluation metrics could have been more	Cresci 2017
Chia et al. [4]	Various classifiers have been used	TFIDF	F score	Larger variety of preprocessing methods will be adopted by the authors	Dataset created by Semantic Evaluation 2018 Task 3: Irony Detection in English Tweets
Shekhar et al. [6]	LSTM	Context words, First & last words etc.	F score	Multiple languages could have been considered	WordNet
Agarwal and Chowdary [15]	ensemble learning-based adaptive model f	Word vector	F1, Precision and recall	Lack of fine-grained and free data set	COVID-19 and US elections
Sadiq et al. [16]	CNN-LSTM and CNN-BiLSTM	unigram and bigram encode with TF–IDF	F1, Accuracy, Precision, Recall	Not considered different features	Cyber-Trolls dataset was created
Alammary [18]	Modified TF-IDF model	TFPOS-IDF	Accuracy, Precision, Recall	Word embedding method not carried out	Made own data set based on Bloom's Taxonomy
Sharma et al. [19]	*MoH* + M-Bert; *MoH* + MuRIL	Not mentioned	Precision, Recall and F1	Error causing factors could be futher enhanced	TRAC-I; HOT and HS Data
Roy et al. [20]	DCNN, LSTM	tf-idf	precision, recall and F1-score	Insufficient data set	Kaggle.com
Al-Garadi et al. [21]	Naïve Bayes, Support vector machine (SVM), Random Forest, KNN	Feature engineering	F1, Precision, recall, AUC, ROC	Cyberbullying concept could have been explored	geo-tagged tweets
Mohapatra et al. [22]	Support vector machine (SVM), naïve Bayes (NB) and Random Forest (RF)	TF-IDF	F1, Precision, recall, accuracy	Different languages could have been considered	From various social media pages

3. Materials and Methods

In the following section, we have discussed how we have prepared the data set, i.e., the corpus of tweets and how we have carried out the pre-processing. In the proposed method section, we have written the proposed classification algorithms.

3.1. Dataset Preparation and Preprocessing

Table 2 provides a description of used datasets. In general, a text augmentation technique is different from an image augmentation technique. Changing the order of the words in a sentence might change the meaning of the sentence; therefore, the text augmentation is slightly different from other augmentation techniques. As the Twitter data [19,22] that we have considered from Kaggle was not sufficient, we have increased the dataset size using nlpaug tool. This nlpaug [23] method uses word-embedding techniques and various augmenter strategies such as insertion and substitutions to augment the data on a character level, word level and sentence level. To augment (i.e., increase the sample size of) tweets, we perform a character level augmentation (using KeyboardAug [24], OcrAug, and RandomAug [25] methods), word level augmentation (AntonymAug [25], ContextualWordEmbsAug, SpellingAug SplitAug, SynonymAug, TfIdfAug, WordEmbsAug and BackTranslationAug and ReservedAug), sentence level augmentation (using Contextual-WordEmbsForSentenceAug, AbstSummAug, and LambadaAug [26]). Figure 1 shows the steps in our framework; it includes 5 major steps [27–34]. The first step is to incorporate the Twitter data upon which a comprehensive pre-processing method has been carried out, afterwards extraction of features from the resulting pre-processed tweets has been accomplished. Finally, proposed ensemble methods have been introduced to predict if any unknown test tweet is hateful or nonhateful [29], which means the final decisional outcome is produced based on the ensemble classifiers' output [35–42].

Table 2. Description of datasets.

Dataset	No. of Instances	No. of Classes	Source
Al-garadi et al. [21]	2.5 million geo-tagged tweets	2	Twitter
SemEval-2019 [35]	10,000	2	Twitter
Waseem (2016) [43]	16,000	2	Twitter
Khan et al. (2022) [44]	DS1 = 80,000 tweets & DS2 = 31,962 tweets	2	Twitter
He et al. (2021) [45]	2400	2	Twitter
Our proposed model	31,962	2	Prepared

For each tweet we performed the following:

Removal of stop words and punctuation: For analysing the sentiment of the tweet, unwanted and irrelevant words are to be deleted because this affects the accuracy of the model. Stop wards removal has helped us to prepare a quality of the input text. Afterwards, punctuation and numbers were also removed from the tweets.

Lowercase: All tweets are converted to lowercase because it is necessary for the input, and normalization has been performed. This also helps us to produce a better prediction.

Tokenization and hashtag extraction: All texts are tokenized in this step. Tokens are a small sentence or word or symbol. Moreover, we perform a hashtag extraction.

Statistics: In Table 3, a comprehensive statistical description of the dataset has been given. In this table id-count, mean, standard deviation, minimum and maximum value have been shown.

Feature Vectors Construction: We have used the count vectorization method to represent the tweets. The vectors are built based on the dimensionality of the size of the tweet vocabulary. The count increases every time one word is encountered, and dimensions are also increased by one. The main objective of a count vectorizer is to fit and learn each word given in the vocabulary. Based on this created vocabulary it creates a document term matrix. In document classification terminologies, count vectorizer is by nature a Boolean model representation, where documents are represented as a collection of terms. In our study, tweets are represented as a set of terms. All terms are either present or absent in the tweet. For a given *i*th tweet, the feature vector $x_{i,}$ is constructed after the binary encoding of each term as

$$x_{i,j} = \begin{cases} 1 & \text{if the } j\text{th term is available} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

Table 3. A statistical description of the train data of Twitter. SD is the standard deviation.

Label	Id-Count	Mean	SD	Min	25%	50%	75%	Max
0	29720	15,974.45	9223.78	1.0	7981.75	15,971.5	23,965.25	31,962.0
1	2242	16,074.90	9267.956	14.0	8075.25	16,095.0	24,022.00	31,961.0

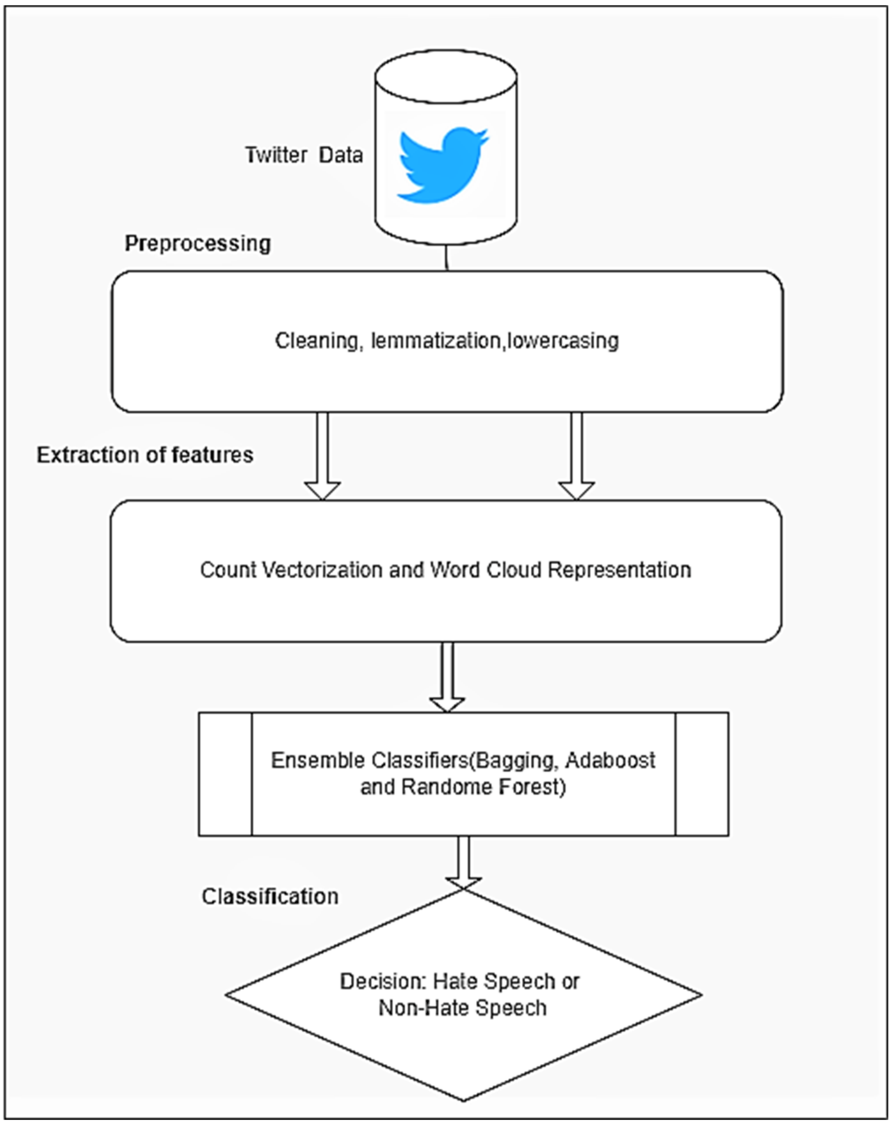
Twitter Data
Preprocessing
Cleaning, lemmatization,lowercasing
Extraction of features
Count Vectorization and Word Cloud Representation
Ensemble Classifiers(Bagging, Adaboost and Randome Forest)
Classification
Decision: Hate Speech or Non-Hate Speech

Figure 1. Proposed framework of Twitter hate-speech detection.

3.2. Classification Methods

The aim of this work is to detect tweets related to hate speech from the Twitter social media platform when a user posts a tweet. This problem can be represented as a binary classification, classifying tweets as hate speech or non-hate speech. Each tweet can be represented as $\{x_1, x_2, x_3, \ldots .\ x_\text{n}\}$ where x_i is the vector encoding the *i*th tweet, and n is the number of tweets. In our work, examples are categorized into two classes. Therefore, $y_i \in \{\text{hate speech, non-hate speech}\}$ is the corresponding label of x_i. To achieve the goal of this work in detecting the hate speech from Twitter, we adopt several ensemble methods explained in the following.

3.2.1. Bagging for Hate Speech Detection

In Bagging (see Algorithm 1), we provide the input training set S consisting of n examples of tweets and corresponding binary labels (see Line 1) [23]. At the first iteration in Lines 2–5 when $k = 1$, a bootstrap sample D_1 is drawn with replacement n time from the training set S. Then, D_1 is provided to a machine-learning algorithm to induce model h. This process (i.e., Lines 2–5) is repeated for additional $k - 1$ times, resulting in $k - 1$ models. For a new tweet (see Line 6), we classify it by taking the majority vote of the k models, where I (.) is an indicator function that generates 1 if the arguments are true. Otherwise, it generates 0.

Algorithm 1 Bagging algorithm

```
1: Bagging($S = \{(x_1,y_1), \ldots, (x_n,y_n)\}$)
2:    for $i = 1$ to $k$ do
3:       Drawing of a bootstrap sample $D_i$ of size $n$ form $S$
4:       Weak learning algorithm takes as input $D_i$ to induce $h_i$        ▷Training Step
5:    end for
6:    $h^*(x_{j*}) = \text{argmax}_y \sum_{i=1}^{k} I(h_i(x_{j*}) = y)$        ▷Testing Step
      {$I(.) = 1$ if the expression is true and 0 otherwise, $y \in$ {hate,none-hate}}
```

3.2.2. Random Forest for Hate Speech Detection

Algorithm 2 shows the steps of Random Forest (RF) to predict tweets as hate speech or non-hate speech. First, a training set S consisting of n tweet examples are provided as input [38]. At the first iteration in Lines 2–5, a bootstrap sample D_1 is created, followed building the tree after consequential selection of best splitting nodes (i.e., attributes) according to the highest gains using gini impurity measure. Then, $k - 1$ trees are generated in the remaining $k - 1$ iterations. For a new tweet (See Line 6), we provide prediction by taking the majority vote of k models.

Algorithm 2 Random forest algorithm

```
1: RandomForest($S = \{(x_1,y_1), \ldots, (x_n,y_n)\}$)
2: for $i = 1$ to $k$ do
3:    Drawing of a bootstrap sample $D_i$ of size $n$ form $S$
4:    Decision tree takes as input $D_i$ to induce $h_i$        ▷Training Step
      after incorporating the following:
         -Select $\sqrt{p}$ features randomly
         -Pick the best split among $\sqrt{p}$ features by highest gain $\Delta$
         using gini impurity measures as
         $\Delta = \text{argmax}_v GINI(parent) - \sum_j \frac{C(v_j)}{C(parent)} GINI(v_j)$
         $\{GINI(t) = 1 - \sum_y [p(y|t)]^2, C(.)$ counts examples of give note$\}$
         -Make a split of two child nodes
5: end for
6: $h^*(x_{j*}) = \text{argmax}_y \sum_{i=1}^{k} I(h_i(x_{j*}) = y)$        ▷Testing Step
   {$I(.) = 1$ if the expression is true and 0 otherwise, $y \in$ {hate,none-hate}}
```

3.2.3. Adaboost for Hate Speech Detection

As shown from Algorithm 3, AdaBoost first receives a training set consisting of n tweet examples [46]. Then, in the first iteration, all examples are assigned equal weights stored in W_1 (Lines 2–4). In the first iteration in lines 5–13, we perform the following. We sampled n times from S based on weights in W, to create D_1, which is then provided to machine learning, to induce model h_1 (lines 6–7). Then, error and importance for h_1 are calculated (Lines 8–9). In lines 10–12, we increase the weight of incorrectly classified examples to pay more attention in the next iteration while decreasing the weight of correctly classified examples. Such a process (i.e., lines 5–13) is repeated for additional $k - 1$ times, which result in k weighted models. For a new tweet, we perform weighted majority vote to generate prediction (Line 14).

Algorithm 3 AdaBoost algorithm

```
1:  AdaBoost ($S$ = {($x_1$,$y_1$), . . . , ($x_n$,$y_n$)})
2:      for $i$ = 1 to $n$ do
3:          $W_1(i) \leftarrow \frac{1}{n}$
4:      end for
5:      for $j$ = 1 to $k$ do
6:          Create a training set $D_j$ by sampling (with replacement) for $S$ base on $W_j$
7:          Weak learning algorithm takes as input $D_j$ to induce $h_j$          ▷Training Step
8:          $\epsilon_j \leftarrow \frac{1}{n}\sum_{i=1}^{n} W_j(i) I[y_i \neq h_j(x_i)]$
9:          $\alpha_j \leftarrow \frac{1}{2}\ln\frac{1-\epsilon_j}{\epsilon_j}$
10:     for $i$ = 1 to $n$ do
11:         $W_{j+1}(i) \leftarrow \frac{W_i(i)exp(-\alpha_j h_j(x_i) y_i)}{Z}$
12:     end for
13:     end for
14:     $h^*(x_{j*}) = \text{argmax}_y \sum_{i=1}^{k} I(h_i(x_{j*}) = y)$          ▷Testing Step
        {$I$(.) = 1 if the expression is true and 0 otherwise, $y \in$ {hate,none-hate}}
```

4. Experiments and Results

In the following sections, we have demonstrated how we carried out the experiments and how they lead us to archive the final results. All experiments have been carried out in Python programming framework with the help of scikit-learn library. The proposed models that have been mentioned earlier are deployed on a personal computer. This personal computer has a processor of Intel® Core™ i5-7200U, CPU 2.70 GHz, 2 Core (s) and 16 GB installed memory (RAM). The personal computer has Windows 10 pro and 64 bits operating system. We conducted a wide set of experiments to evaluate the performance of three classifiers (e.g., Bagging, Random Forest and Adaboost).

4.1. Evaluation Metrics

We considered the evaluation metrics such as accuracy, precision, recall and F-measure. The calculation has been carried out in terms of False Negative (FN), True Negative (TN), False Positive (FP) and True Positive (TP). TP is calculated in terms of hate-speech tweets that were correctly predicted as hate speech tweets. TN is calculated in terms of non-hate-speech tweets that were correctly predicted as non-hate speech. FP is calculated in terms of non-hate-speech tweets that were incorrectly predicted as hate speech. FN is calculated in terms of hate-speech tweets that were incorrectly predicted as none-hate speech. Below we show performance measures considered in this study.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \tag{2}$$

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

4.2. Results and Discussion

In the below section, experimental results obtained by using the proposed models have been shown. We choose the word cloud method to show the most significant words.

Afterwards, we have shown distribution, frequent words and training examples of tweets in terms of visual graphs.

4.2.1. Word Cloud Method

We perform the word cloud method to display the hateful words. It takes tweets followed by performing a resampling controlled with the bilinear interpolation argument. Then, we plot the word cloud. This method displays the most significant hateful words. We show the word cloud for the important words in the tweets in Figure 2. In Figure 3, we show the word cloud of hateful speech in test tweets. By showing the word cloud, we represent the significant data points in these textual data. In addition, the significant textual data points are also highlighted by the word cloud. To generate such images, we have used matplotlib [35], pandas [36] and word-cloud packages [36]. The larger font size found on the word cloud describes the importance of the tweet. In Figure 2, we can see words such as ‘day’ and ‘happy’ with large font sizes, which are the words with positive sentiments. However, in Figure 3, we can see the words related to the hateful words on the word could. The words with large font sizes such as ‘racism,’ ‘Trump,’ ‘Obama,’ ‘white’ and users are contributing to the hateful sentiments.

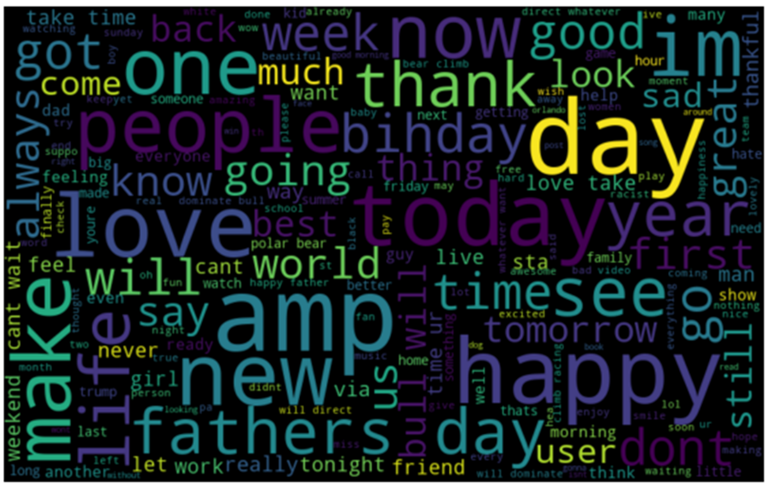

Figure 2. Important words in tweets shown by the word cloud.

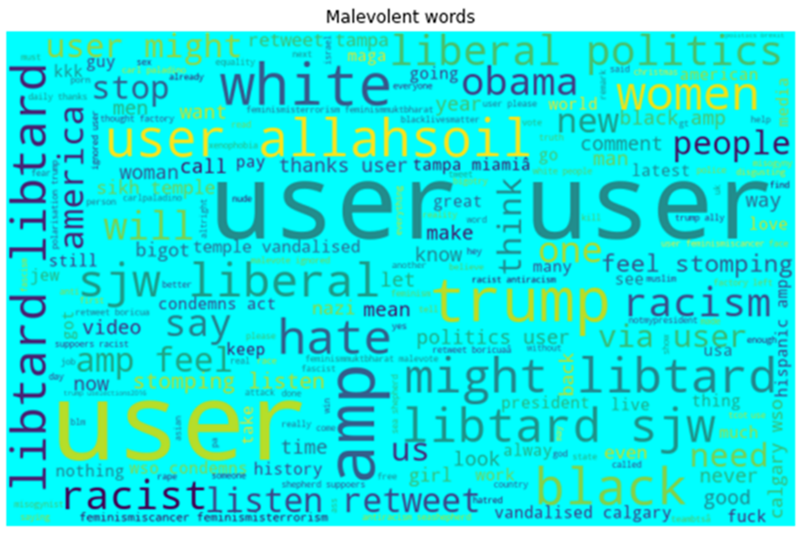

Figure 3. Word cloud of hateful words.

4.2.2. Distribution, Frequent Words and Training Examples of Tweets

In Figure 4, we show the distribution of the tweets in terms of length and frequency. Figure 5 shows the most frequently used words in tweets. It can be seen that the word cloud has found the most common words for hate speech and non-hate speech that are associated with the tweets [30–35].

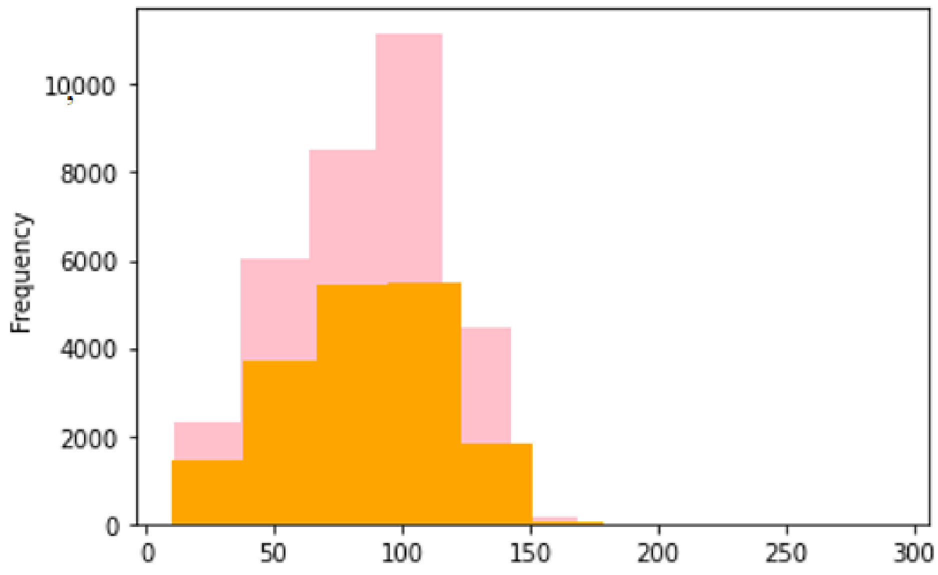

Figure 4. Distribution of tweets.

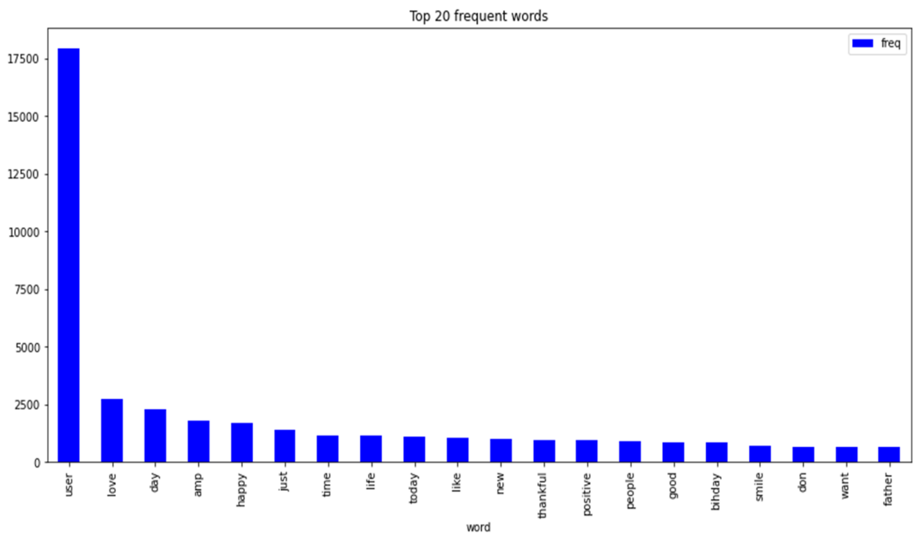

Figure 5. Twenty most frequently words in tweets.

In Table 4, we show a sample of pre-processed and cleaned tweets. It can be observed that even among the nine examples, the number of hate-related tweets is not that much. Moreover, the length of the tweets is not so lengthy, consisting of up to 150 characters. In Table 5, we report performance results of three models (i.e., Bagging, Random Forest, and AdaBoost). Random Forest achieves the highest results of 95% when accuracy and F1 score are considered. Moreover, Random Forest achieves the highest result of 78% when using AUC [37]. In Tables 6–8, we have reported additional performance results (of each model) using additional metrics such as precision, recall, F1 score, support. Moreover, we found macro average and weighted averages of precision, recall, F1 score, and support for all the presented methods.

Table 4. Few training examples for tweets, showing cleaned tweets and their lengths.

	Id	Label	Tweet	Clean Tweet	Length
0	31434.0	0	ode to depression I infusing reality with hope	ode to depression infusing reality with hope	49
1	14785.0	0	what an awful day on #Twitter. crazies on both...	what an awful day on crazies on both ends of...	135
2	7561.0	0	@user @user #wishing you a lovely mid week we...	you a lovely mid week wednesday day	60
3	29558.0			vine by	76
4	903.0	0	"a picture is woh a thousand words." #sundayre...	a picture is woh a thousand words	88
5	22216.0	0	my thoughts and prayers and deepest condolence...	my thoughts and prayers and deepest condolence...	113
6	10853.0	1	@user #allahsoil not all muslims hate america....	not all muslims hate america	102
7	2761.0	0	how to save thanksgiving #thanksgiving	how to save thanksgiving	41
8	12039.0	0	beautiful world	beautiful world ~ happy friday	96
9	9170.0	0	if you don't think every day is a good day, ju...	if you don't think every day is a good day just...	103

Table 5. Comparative study of three models, including Bagging, AdaBoost and Random Forest, in terms of evaluating the generalization performance using accuracy, F1 score and Area Under the ROC Curve (AUC). High performance results are shown in bold.

	Accuracy	F1 Score	AUC
Bagging	0.94	0.60	0.76
Random Forest	**0.95**	**0.95**	**0.78**
AdaBoost	0.94	0.53	0.69

Table 6. Classification report of Bagging.

Class	Precision	Recall	F1-Score	Support
1	0.65	0.55	0.60	520
0	0.96	0.97	0.97	6013
accuracy			0.94	8166
macro avg	0.81	0.77	0.79	8166
weighted avg	0.94	0.94	0.94	8166

Table 7. Classification report of Random Forest.

Class	Precision	Recall	F1-Score	Support
1	0.74	0.58	0.65	520
0	0.96	0.98	0.97	6013
macro avg	0.85	0.78	0.81	6533
weighted avg	0.95	0.95	0.95	6533

Our hate speech classification framework first starts with the collection of a Twitter dataset, and this has been accomplished by preparing the Twitter data with the data augmentation technique nlpaug [35,36]. The count vectorizer technique has not been exploited much by the researchers; therefore, we have adopted a count vectorizer method to check its potential capability as a representation for tweets, where we converted the whole tweets to vectors of token counts [33]. Tokenization method has been incorporated as a parser to remove specific unwanted words from the collected tweets and finally extracting features. We used count vectorizer in Scikit-learn to accomplish this task [39–45]. The outcome of count vectorization provides a typical sparse matrix in terms of counts. The reason that we chose count vectorization is that data sparsity is very common in tweets as we intend to detect hate speech; moreover, the irregular form of hate speech in a short text of tweets makes it sparser by nature.

Table 8. Classification report of AdaBoost.

Class	Precision	Recall	F1-Score	Support
1	0.79	0.40	0.54	672
0	0.95	0.99	0.97	7494
macro avg	0.87	0.70	0.75	8166
weighted avg	0.94	0.94	0.93	8166

To the best of our knowledge, none of the authors have reported the visualization of tweets, especially hateful words, which helps us to understand the specific words from the tweets that play a key role in generating hateful sentiments among youth. In the word cloud, a larger font size of hateful words presents its importance in a more prominent way than the other words of the tweets that are in the overall cluster. As we have an ample number of Twitter data, we obtained a better word cloud visualization. We were able to represent semantically more meaningful visual outcomes.

5. Conclusions and Future Work

In this paper, we propose a machine-learning framework to detect hate speech from Twitter data. First, we use a count vectorizer technique to construct feature vectors, which are then coupled with its corresponding labels to be provided as input to Bagging, AdaBoost, and Random Forest. Second, we perform a word-cloud visualization to inspect the hateful tweets and consequently show the significant textual data. Statistical information has been reported to find the tweets with highest frequency to better understand the data. Our experimental results show that Random Forest outperformed AdaBoost and Bagging by generating the highest performance results when considering accuracy, F1 score, and AUC performance measures. These results demonstrate that Random Forest, when coupled with our framework, is a candidate classifier for the task of detecting hate speech within Twitter.

Future work includes (1) utilizing the computational framework under different machine-learning settings, including active learning and transfer learning; and (2) tackling other natural language processing tasks in medical domains using our framework.

Author Contributions: Conceptualization, T.T. and S.S.R.; methodology, T.T.; software, S.S.R.; validation, T.T. and S.S.R.; formal analysis, T.T.; investigation, S.S.R.; resources, T.T.; data curation, T.T.; writing—original draft preparation, S.S.R.; writing—review and editing, S.S.R.; visualization, T.T.; supervision, S.S.R.; project administration, T.T.; funding acquisition, T.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by Institutional Fund Project under grant no (IFPIP: 924-611-1442). Therefore, authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data used in this study are free open source.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Araque, O.; Iglesias, C.A. An ensemble method for radicalization and hate speech detection online empowered by sentic computing. *Cogn. Comput.* **2021**, *14*, 48–61. [CrossRef]
2. MacAvaney, S.; Yao, H.R.; Yang, E.; Russell, K.; Goharian, N.; Frieder, O. Hate speech detection: Challenges and solutions. *PLoS ONE* **2019**, *14*, e0221152. [CrossRef] [PubMed]
3. Hajibabaee, P.; Malekzadeh, M.; Ahmadi, M.; Heidari, M.; Esmaeilzadeh, A.; Abdolazimi, R.; James, H., Jr. Offensive language detection on social media based on text classification. In Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 26–29 January 2022; pp. 92–98.
4. Chia, Z.L.; Ptaszynski, M.; Masui, F.; Leliwa, G.; Wroczynski, M. Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection. *Inf. Process. Manag.* **2021**, *58*, 102600. [CrossRef]
5. Van Hee, C.; Lefever, E.; Verhoeven, B.; Mennes, J.; Desmet, B.; De Pauw, G.; Daelemans, W.; Hoste, V. Detection and fine-grained classification of cyberbullying events. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP, Online, 1–3 September 2015; pp. 672–680.
6. Shekhar, S.; Garg, H.; Agrawal, R.; Shivani, S.; Sharma, B. Hatred and trolling detection transliteration framework using hierarchical LSTM in code-mixed social media text. *Complex Intell. Syst.* **2021**, 1–14. [CrossRef]
7. Mihaylova, T.; Gencheva, P.; Boyanov, M.; Yovcheva, I.; Mihaylov, T.; Hardalov, M.; Kiprov, Y.; Balchev, D.; Koychev, I.; Nikolova, I.; et al. SUper Team at SemEval-2016 Task 3: Building a feature-rich system for community question answering. *arXiv* **2021**, arXiv:2109.15120.
8. Alnazzawi, N. Using Twitter to Detect Hate Crimes and Their Motivations: The HateMotiv Corpus. *Data* **2022**, *7*, 69. [CrossRef]
9. Gambäck, B.; Sikdar, U.K. Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 4 August 2017; pp. 85–90.
10. Schmidt, A.; Wiegand, M. A survey on hate speech detection using natural language processing. In Proceedings of the International Workshop on Natural Language Processing for Social Media, SocialNLP, ACL, Valencia, Spain, 3 April 2017; pp. 1–10.
11. Greevy, E.; Smeaton, A.F. Classifying racist texts using a support vector machine. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Sheffield, UK, 25–29 July 2004; pp. 468–469.
12. Alkomah, F.; Ma, X. A Literature Review of Textual Hate Speech Detection Methods and Datasets. *Information* **2022**, *13*, 273. [CrossRef]
13. Abro, S.; Shaikh, Z.S.; Khan, S.; Mujtaba, G.; Khand, Z.H. Automatic Hate speech Detection using Machine Learning: A Comparative Study. *Mach. Learn.* **2020**, *11*, 484–491. [CrossRef]
14. Diao, S.; Xu, R.; Su, H.; Jiang, Y.; Song, Y.; Zhang, T. Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, 1–6 August 2021; Long Papers; Volume 1, pp. 3336–3349.
15. Agarwal, S.; Chowdary, C.R. Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. *Expert Syst. Appl.* **2021**, *185*, 115632. [CrossRef]
16. Sadiq, S.; Mehmood, A.; Ullah, S.; Ahmad, M.; Choi, G.S.; On, B.W. Aggression detection through deep neural model on twitter. *Future Gener. Comput. Syst.* **2021**, *114*, 120–129. [CrossRef]
17. Beddiar, D.R.; Jahan, M.S.; Oussalah, M. Data expansion using back translation and paraphrasing for hate speech detection. *Online Soc. Netw. Media* **2021**, *24*, 100153. [CrossRef]
18. Alammary, A.S. Arabic Questions Classification Using Modified TF-IDF. *IEEE Access* **2021**, *9*, 95109–95122. [CrossRef]
19. Sharma, A.; Kabra, A.; Jain, M. Ceasing hate with MoH: Hate Speech Detection in Hindi–English code-switched language. *Inf. Processing Manag.* **2022**, *59*, 102760. [CrossRef]
20. Roy, P.K.; Tripathy, A.K.; Das, T.K.; Gao, X.Z. A Framework for Hate speech Detection Using Deep Convolutional Neural Network. *IEEE Access* **2020**, *8*, 204951–204962. [CrossRef]
21. Al-garadi, M.A.; Varathan, K.D.; Ravana, S.D. Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. *Comput. Hum. Behav.* **2016**, *63*, 433–443. [CrossRef]
22. Mohapatra, S.K.; Prasad, S.; Bebarta, D.K.; Das, T.K.; Srinivasan, K.; Hu, Y.C. Automatic Hate speech Detection in English-Odia Code Mixed Social Media Data Using Machine Learning Techniques. *Appl. Sci.* **2021**, *11*, 8575. [CrossRef]
23. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
24. Zisad, S.N.; Hossain, M.S.; Andersson, K. Speech emotion recognition in neurological disorders using convolutional neural network. In Proceedings of the International Conference on Brain Informatics, Padua, Italy, 19 September 2020; Springer: Cham, Switzerland, 2020; pp. 287–296.
25. Goel, K.; Rajani, N.; Vig, J.; Tan, S.; Wu, J.; Zheng, S.; Xiong, C.; Bansal, M.; Ré, C. Robustness gym: Unifying the nlp evaluation land-scape. *arXiv* **2021**, arXiv:2101.04840.
26. Thakur, N.; Reimers, N.; Daxenberger, J.; Gurevych, I. Augmented sbert: Data augmentation method for improv-ing bi-encoders for pairwise sentence scoring tasks. *arXiv* **2020**, arXiv:2010.08240.
27. Ciolino, M.; Noever, D.; Kalin, J. Multilingual Augmenter: The Model Chooses. *arXiv* **2021**, arXiv:2102.09708.

28. Hu, Z.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; Tu, K. Multi-View Cross-Lingual Structured Prediction with Minimum Supervision. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, 1–6 August 2021; Long Papers; Volume 1, pp. 2661–2674.
29. Gao, Y.; Zhu, H.; Ng, P.; Santos CN, D.; Wang, Z.; Nan, F.; Zhang, D.; Nallapati, R.; Arnold, A.O.; Xiang, B. Answering ambiguous questions through generative evidence fusion and round-trip prediction. *arXiv* **2020**, arXiv:2011.13137.
30. William, P.; Gade, R.; esh Chaudhari, R.; Pawar, A.B.; Jawale, M.A. Machine Learning based Automatic Hate Speech Recognition System. In Proceedings of the 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 7–9 April 2022; pp. 315–318.
31. Garcia, K.; Berton, L. Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Appl. Soft Comput.* **2021**, *101*, 107057. [CrossRef] [PubMed]
32. Carvalho, J.; Plastino, A. On the evaluation and combination of state- of-the-art features in twitter sentiment analysis. *Artif. Intell. Rev.* **2021**, *54*, 1887–1936. [CrossRef]
33. Singh, C.; Imam, T.; Wibowo, S.; Grandhi, S. A Deep Learning Approach for Sentiment Analysis of COVID-19 Reviews. *Appl. Sci.* **2022**, *12*, 3709. [CrossRef]
34. Daghriri, T.; Proctor, M.; Matthews, S. Evolution of Select Epidemiological Modeling and the Rise of Population Sentiment Analysis: A Literature Review and COVID-19 Sentiment Illustration. *Int. J. Environ. Res. Public Health* **2022**, *19*, 3230. [CrossRef]
35. Gorrell, G.; Kochkina, E.; Liakata, M.; Aker, A.; Zubiaga, A.; Bontcheva, K.; Derczynski, L. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 845–854.
36. Ayo, F.E.; Folorunso, O.; Ibharalu, F.T.; Osinuga, I.A.; Abayomi-Alli, A. A probabilistic clustering model for hate speech classification in twitter. *Expert Syst. Appl.* **2021**, *173*, 114762. [CrossRef]
37. Vel, S.S. Pre-Processing techniques of Text Mining using Computational Linguistics and Python Libraries. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; IEEE: Manhattan, NY, USA, 2021; pp. 879–884.
38. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recogni-tion, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
39. Gholizadeh, S. *Top Popular Python Libraries in Research*; Authorea Preprints; ResearchGate: Berlin, Germany, 2022.
40. Pajankar, A.; Joshi, A. Introduction to Pandas. In *Hands-on Machine Learning with Python*; Apress: Berkeley, CA, USA, 2022; pp. 45–61.
41. Jokić, D.; Stanković, R.; Krstev, C.; Šandrih, B. A Twitter Corpus and lexicon for abusive speech detection in Serbian. In Proceedings of the 3rd Conference on Language, Data and Knowledge (LDK 2021), Zaragoza, Spain, 1–4 September 2021; Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Wadern, Germany, 2021.
42. Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; Villata, S. A multilingual evaluation for online hate speech detection. *ACM Trans. Internet Technol. TOIT* **2020**, *20*, 1–22. [CrossRef]
43. Waseem, Z. Are you a racist or am I seeing things? Annotator influence on hate speech detection on twitter. In Proceedings of the First Workshop on NLP and Computational Social Science, Austin, TX, USA, 5 November 2016; pp. 138–142.
44. Khan, S.; Kamal, A.; Fazil, M.; Alshara, M.A.; Sejwal, V.K.; Alotaibi, R.M.; Baig, A.R.; Alqahtani, S. HCovBi-caps: Hate speech detection using convolutional and Bi-directional gated recurrent unit with Capsule network. *IEEE Access* **2022**, *10*, 7881–7894. [CrossRef]
45. He, B.; Ziems, C.; Soni, S.; Ramakrishnan, N.; Yang, D.; Kumar, S. Racism is a virus: Anti-asian hate and counterspeech in social media during the COVID-19 crisis. In Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Virtual Event, The Netherlands, 8–11 November 2021; pp. 90–94.
46. Schapire, R.E. Explaining adaboost. In *Empirical Inference*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 37–52.