Article

Multi-Model Learning to Detect Twitter Hate Speech

Dharmaraj R. Patil[1] and Tareek M. Pattewar[2]

[1] Department of Computer Engineering, R.C. Patel Institute of Technology, Shirpur, Maharashtra, India
[2] Department of Computer Engineering, Vishwakarma University, Pune, Maharashtra, India
* Correspondence: dharmaraj.patil@rcpit.ac.in

Abstract: Users on the social networking platform have the freedom to express themselves freely. Towards the same time, this has created a forum for disagreement and hate directed at someone, society, racism, sexual orientation, and so on. Identifying hate online is a challenging task. Researchers from all around the world have contributed major methods for detecting hate speech, but owing to the issue's complexity, there are still many unresolved issues. In this research, we offer a multi-model learning strategy for detecting hate speech on Twitter. We utilised the Kaggle TwitterHate dataset, which had 31962 tweets categorised as binary hate or non-hate, to evaluate our technique. The suggested method is tested using commonly used machine learning classifiers with multi-model technique. Using TF-IDF features, we acquired detection results of 96.29 %, precision of 96%, recall of 96%, and f1-score of 96%.

Keywords: Hate speech detection; Social media; Machine learning; Multi-model learning

1. Introduction

Nowadays online social networks (OSN) are the most important and fastest means of communication. In fact, it is the popular way to communicate with each other [1]. Offers users freedom of expression. Although it is the main medium and the most popular form of communication, it becomes the platform for spreading hate speech related to individuals, racism, sexual purposes, cyberbullying, etc. Aggressive and intentional acts are reported using OSNs like Twitter, WhatsApp, Facebook, Reddit etc. [3]. Identifying hate speech online is a difficult problem due to the complex and multilingual nature of the support that NSOs provide to users [4,5]. Effective hate speech detection is highly dependent on the availability of benchmark training datasets [6,7,8]. Natural language processing is widely used to detect hate and offensive language [9,10]. Several studies suggested the using machine learning as well as deep learning techniques to effectively detect hate and offensive language in OSNs, and also obtained acceptable results. [11,12]. Researchers around the world propose different language-specific datasets to train the classifiers and finally achieved significant recognition performance [13,14,15,16].

In this paper, we have proposed multi model learning approach to identify hate speech or non-hate speech on Twitter's OSN platform. We used Kaggle's TwitterHate dataset, which contains 31,962 tweets tagged as binary hate or non-hate. The data set was significantly skewed: 93% of tweets or 29,695 tweets contained data from Twitter without hate tags and 7% or 2,240 tweets contained data from Twitter with hate tags. We pre-processed the dataset using NLP (Natural Language Processing) techniques such as stop word removal, tokenization, stemming, lemmatization, bag of words (BOW), hashtag removal, and URL removal. We have used features such as TF-IDF, sentiment polarity score and doc2vector. We used state-of-the-art machine learning classifiers such as Decision Tree (DT), Logistic Regression (LR), XGBoost, Random Forest (RF), Extra Tree (ET), AdaBoost and lastly Support Vector Machines (SVM). Using classifiers mentioned above, we created a multi-model learning classifier using the majority voting technique. It is found that with the proposed approach we achieved more acceptable and significant detection results, such as accuracy 96.29%, precision of 96%, recall of 96% and f1-score of 96% with the TF-IDF features. The major findings of this paper are as follows:

- We proposed the multi-model learning approach for identifying hate-speech and non-hate-speech on Twitter platform.
- The proposed approach was found to achieve results such as 96.29% accuracy, 96% precision, 96% recall and 96% f1-score. The experimental results suggest that our results are acceptable and more reasonable.
- The experimental analysis shows that our results are more stable and acceptable compared to independent machine learning classifiers.

The remainder of this paper is organized as follows. Section 2 discusses brief related work on hate speech detection. Section 3 describes the proposed Materials and Methods in detail. Section 4 discusses the experimental results and the performance of the proposed approach. Finally, in Section 5, the conclusion is presented.

2. Related Work

In recent various approaches have been proposed by researchers towards hate speech detection for major OSNs like Twitter, Facebook, Reddit, Wikipedia, YouTube etc. We described some of the approaches as follows.

Ross, B., et al. have measured the reliability of the hate speech and to what extent in accordance with subjective ratings [18]. Davidson, T. et al. have used a crowd-sourced hate speech lexicon to collect the hate speech from the tweets. They have the multiclass technique to distinguish between different categories of the hate speeches [19]. Badjatiya, P. have used deep learning framework for hate speech detection from tweets. They defined this problem by classifying tweets into categories like racist, sexist or neither. They have used the benchmark dataset of annotated tweets of 16K and stated that the deep learning techniques outperformed the char/n-gram techniques [20].

Gao, L. et al. have used logistic regression and neural network models for the hate speech detection. They have provided the corpus of the hate speech dataset. They have stated that both models have performed well on the benchmark dataset and achieved better results in comparison with the baseline classifiers [21]. Founta, A. M. et al. have proposed incremental and iterative methods for the detection of abusive language on the social media platforms like Facebook and Twitter. According to them, the proposed methodology working better for the reduction but robust labels to characterize the abusive tweets [22]. Zhang, Z. et al. have targeted to identify the characteristics of the tweets like race, and religion. They have stated that the hate speech detection is a challenging task due to unique, discriminative features. They have proposed the Deep Neural Network for the features extraction and to capture the semantics of the hate speech [23].

Gröndahl, T. et al. have suggested that the data and labeling is more important than the accurate hate speech detection models [24]. Mishra, P. et al. have addressed the problem of obfuscation of words by users to evade detection model. They have designed the model for embeddings for unseen words. They have stated that their approach achieved significant improvement in the detection of hate speech on Twitter and Wikipedia datasets [25].

Kshirsagar, R. et al. have presented neural network based approach for classification of hate and non-hate Twitter speech basically in racist and sexist. They have used three datasets namely, Sexist/Racist (SR), HATE and HAR. They have used word embedding and pooling features to train the deep neural network [26].

Qian, J., et al. presented two large fully-labeled datasets collected from Gab and Reddit. They evaluated the datasets in order to better understand common intervention tactics and to investigate the effectiveness of common automatic response generation methods [27]. Mansourifar, H. et al. have collected significant dataset from the clubhouse. They have analyzed the dataset stastically using the Google Perspective Scores. According to them, the Perspective Scores outperforms the Bag of Words andWord2Vec textual features [28].

Gautam, A. et al. have presented the dataset related to MeToo movement. They have manually annotated the dataset for five different linguistic aspects like, relevance, stance, hate speech, sarcasm, and dialogue acts [29]. Silva, L. have analyzed the targets of hate speech in online social media. The have collected the traces of the two social media like Twitter and Whisper. According to them, their approach identifies the hate speeches and provides the directions for prevention and detection approaches [30].

Salminen, J. et al. have manually labeled the posts from YouTube and Facebook videos. They have created taxonomy of different types of targets and trained machine learning classifiers to automatically detect the online hate speeches. They have conducted experiments using machine learning classifiers, like Logistic Regression, Decision Tree, Random Forest, Adaboost, and Linear SVM, to generate a multi-class, multi-label classification model that automatically detects and classifies hateful comments. They have achieved best performance using Linear SVM with an average F1 score of 0.79 using TF-IDF features [31]. ElSherief, M. et al. have presented the comparative study of hate speech instigators and target users on Twitter. According to them, hate instigators target more popular and high profile Twitter users [32].

3. Materials and Methods

3.1. Proposed Twitter Hate Speech Detection System

Figure 1 shows our proposed Twitter hate detection system. It has four stages like pre-processing, feature extraction and learning.

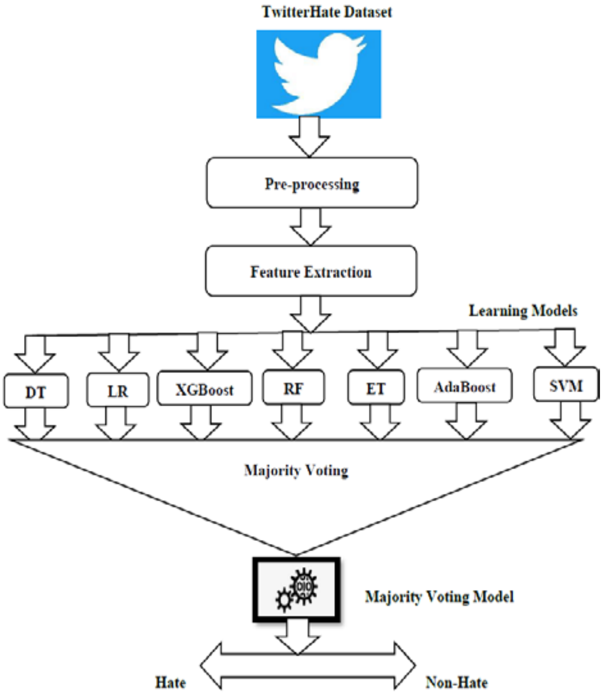

Figure 1. Framework of Proposed Twitter Hate Speech Detection System.

3.2. Pre-processing

We passed the raw TwitterHate dataset from Kaggle into the Python pre-processing function. We employ a text preparation technique that includes:

- Punctuation Removal,

- Tokenization,
- Stop Word Removal,
- Word Stemming,
- URL Removal, and
- Names to eliminate undesired items from the dataset.

3.3. Feature Extraction

The processed text is then passed on to feature extraction, which extracts features such as n-gram TF-IDF weights, sentiment polarity scores, and the doc2vec vector.

3.3.1. TF_IDF Features

The TF_IDF statistic is intended to assess the relevance of a word in a set of texts (or corpus). It is represented by an equation (1). The frequency of the term, TF (t, d), is the frequency of occurrence of the term t internal document d.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \qquad (1)$$

Where $f_{t,d}$ is the raw count of a term in a document, i.e., the number of times that term t occurs in document d.

A word's inverse document frequency represents the fraction of documents in the corpus that include the term. As seen in the equation (2), words that are unique to a small percentage of documents have greater relevance values than terms that are common to all documents,

$$IDF(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \qquad (2)$$

Where N: total number of documents in the corpus.

$|\{d \in D : t \in d\}|$: number of documents where the term t appears.

The Term frequency–inverse document frequency (TF-IDF) is calculated using equation (3),

$$TFIDF(t,d,D) = TF(t,d) \cdot IDF(t,D) \qquad (3)$$

3.3.2. Sentiment Polarity Score Features

The polarity numeric number determines whether a statement is negative or positive. Subjectivity, on the other hand, relates to whether a text is objective or subjective.

3.3.3. Doc2vec Features

Doc2vec is the NLP tool for representing documents as vectors which is a generalization of the word2vec method. Figure 2 illustrates the architecture of the Doc2Vec model. Figure1 is based on BOW (bag of words) model, but lieu of just analyzing neighbouring word to predict word, we have included another feature vectors that is unique to the document. Therefore, when training the word vectors W, the document vector D also trained and, at the end of the training, carries a numeric representation of the document.

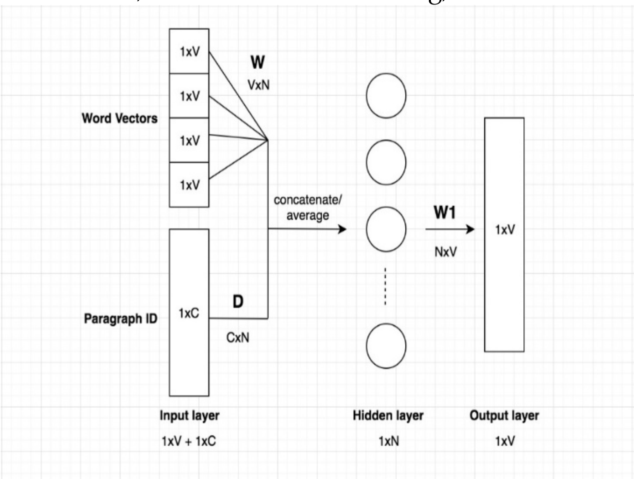

Figure 2. The architecture of Doc2Vec model

3.4. *Machine Learning Classifiers for Twitter Hate Speech Detection*

We have evaluated the performance of 7 machine learning classifiers for the detection of Twitter hate speech, including Decision Tree, XGBoost, Logistic Regression, Random Forest, Extra Tree, AdaBoost and Support Vector Machines. From these 7 machine learning classifiers we have built a multi-model classifier using majority voting for the final decision. We have used the scikit-learn python library implementation of each classifier [33]. The brief discussion of each classifier is given as below.

3.4.1. Decision Tree

One of the most popular classification approaches is decision tree learning. It is highly efficient and has classification accuracy comparable to other learning methods. A decision tree is a tree that reflects the classification model that has been learned. It's an easy-to-understand decision tree classification paradigm. The method evaluates all feasible data split tests and chooses the one with the highest information gain [34,35].

3.4.2. Logistic Regression

Logistic regression is a popular Machine Learning algorithm that is used in the Supervised Learning approach. It is used to predict the categorical dependent variable from a set of independent variables. The outcome of a categorical dependent variable is predicted using logistic regression. As a result, the outcome must be a categorical or discrete value. It can be Yes or No, 0 or 1, true or False, and so on, but instead of displaying precise values like 0 and 1, it offers probability values that lie between 0 and 1 [36,37].

3.4.3. XGBoost

The XGBoost (eXtreme Gradient Boosting) approach is well-known and successful. Gradient boosting is a supervised learning approach that combines estimates from a series of simpler and weaker models to properly predict a target variable. Because of its strong handling of a wide range of data kinds, relationships, and distributions, as well as the huge range of hyperparameters that can be fine-tuned, the XGBoost technique performs well in machine learning issues. XGBoost is capable of dealing with regression, classification (binary and multiclass), and ranking issues [17].

3.4.4. Random Forest

A random forest, as the name implies, is composed of a huge number of individual decision trees that collaborate as an ensemble. The random forest generates a class prediction for each tree, and the class with the highest votes becomes the forecast of our model. The Random Forest's basic principle is communal knowledge, which is both simple and powerful. The random forest model is particularly successful because it is composed of a large number of largely uncorrelated models (trees) that collaborate to outperform each of the individual constituent models [38,39].

3.4.5. Extra Tree

Extra trees classifier is type of ensemble learning techniques that aggregates the classification results of several non correlated decision trees gathered in "Forest" to obtain its classification results. It is conceptually very similar to Random Forest Classifier and varies mostly in the manner the decision trees in the forest are formed. The Extra Trees Forest's Decision Trees are constructed from training samples. Then, at each test node, each tree is given random samples of k feature from the feature sets. From which each of the decision tree must select the best features to divide data using important mathematical criterion. This random selection of features results in the construction of several de-correlated decision trees [40,41].

3.4.6. AdaBoost

AdaBoost is the most frequently used and researched algorithm, with applications in a wide range of fields. Freund and Schapire created the AdaBoost algorithm in 1995. Abstract Boosting is a machine learning strategy that combines a large number of weak and incorrect classifiers to generate a highly accurate classifier. It's simple to use, quick, and simple to comprehend. It does not require any previous information from the weak learner, hence it may be utilised in combination with any weak hypothesis identification technique [42,43].

3.4.7. Support Vector Machine

The Support Vector Machine (SVM) is a supervised machine learning technology that may be used to handle classification and regression issues. It is, however, mostly used in classification difficulties. Each data item is represented as a point in n-dimensional space, with the value of each feature being the value of a specific coordinate in the SVM algorithm. Then, we achieve classification by selecting the hyper-plane that best separates the two classes. Individual observation coordinates are utilised to compute support vectors. The SVM classifier is a frontier that best differentiates between the two classes (hyper-plane/line) [44,45].

3.4.8. Majority Voting

Voting is the most basic ensemble strategy, and it is usually quite effective. It may be applied to classification and regression issues. In this scenario, it divides a model into two or more sub-models, in this case five. To integrate predictions from each sub-model, the majority voting technique is employed. Figure 3 depicts the majority voting process. It is a meta-classifier that identifies machine learning classifiers that are conceptually similar or different using a majority vote. We anticipate the final class label using majority voting, which is the class label that classification algorithms most commonly predict. Using equation (4) and the majority vote of each classifier Cj, we predict the class label y [46, 47].

$$y = mode\{C1(x), C2(x), ..., Cm(x)\} \qquad (4)$$

where,y = predicted class label and C1(x), C2(x),..., Cm(x)=classification models.

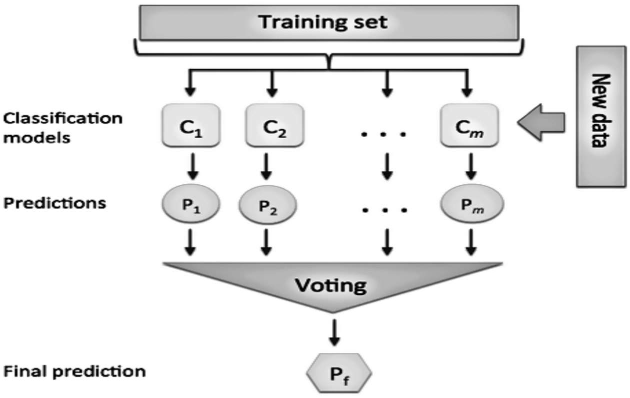

Figure 3. Majority Voting Algorithm

4. Experimental Setup and Evaluation

4.1. Dataset and Data Source

We utilized the TwitterHate Kaggle dataset, which contains 31,962 tweets [48]. The datasets were significantly skewed, having 93% tweets, or 29695 tweets, consisting of non hate labeled Twitter data and 7%, or 2240 tweets, consisting of hate-labeled Twitter data. For training and testing, the classifiers divided the dataset in an 80:20 ratios.

4.2. Evaluation Measures

We utilized the following metrics to assess classifier performance. A binary classifier labels all data elements in a test dataset with a 0 or 1. True positive (TPH), true negative (TNH), false positive (FPH), and false negative (FNH) are the four results of this classification [49]. The following equations are used to compute the $Accuracy_H$, $Precision_H$, $Recall_H$, and $F1-score_H$ measures.

$$Accuracy_H = \frac{TP_H + TN_H}{TP_H + TN_H + FP_H + FN_H} \qquad (5)$$

$$Precision_H = \frac{TP_H}{TP_H + FP_H} \qquad (6)$$

$$Recall_H = \frac{TP_H}{TP_H + FN_H} \qquad (7)$$

$$F1-score_H = 2 \cdot \frac{\mathrm{Precision}_H \cdot \mathrm{Recall}_H}{\mathrm{Precision}_H + \mathrm{Recall}_H} \qquad (8)$$

4.3. Performance Evaluation of classifiers using Doc2Vect Features on TwitterHate dataset

Table 1 and Figure 4 shows the performance of the models on the TwitterHate dataset using Doc2Vect features in terms of $Accuracy_H$, $Precision_H$, $Recall_H$ and $F1-score_H$. The Decision Tree model achieved the $Accuracy_H$ of 86.67%, $Precision_H$ of 88%, $Recall_H$ of 87% and $F1-score_H$ of 87%. The Logistic Regression model achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ 90%. The XGBoost model achieved the $Accuracy_H$ of

doi:10.20944/preprints202203.0333.v1

93.38%, $Precision_H$ of 94%, $Recall_H$ of 93% and $F1-score_H$ of 90%. The Random Forest model achieved the $Accuracy_H$ of 93.40%, $Precision_H$ of 93%, $Recall_H$ of 93% and $F1-score_H$ of 90%. The Extra Trees model achieved the $Accuracy_H$ of 93.45%, $Precision_H$ of 93%, $Recall_H$ of 93% and $F1-score_H$ of 91%. The AdaBoost model achieved the $Accuracy_H$ of 93.21%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ of 90%. The Support Vector Machine model achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ of 90%. Our proposed Multi-model model using the majority voting achieved the $Accuracy_H$ of 93.48%, $Precision_H$ of 92%, $Recall_H$ of 93% and $F1-score_H$ of 91%. It is found that, our proposed approach achieved more stable and acceptable results in comparison with aforementioned models.

Table 1: Performance Evaluation of classifiers using Doc2Vect Features on TwitterHate dataset

Model	$Accuracy_H$ (%)	$Precision_H$ (%)	$Recall_H$ (%)	$F1-score_H$ (%)
DT	86.67	88	87	87
LR	93.23	87	93	90
XGBoost	93.38	94	93	90
RF	93.40	93	93	90
ET	93.45	93	93	91
AdaBoost	93.21	87	93	90
SVM	93.23	87	93	90
Proposed Approach	93.48	92	93	91

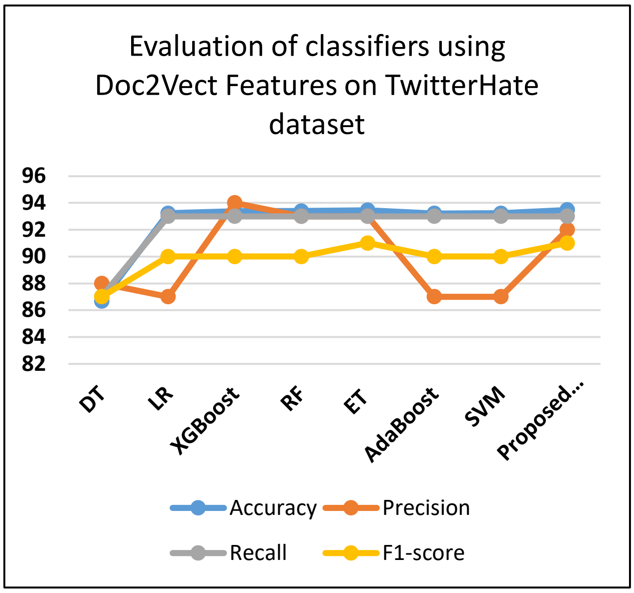

Figure 4. Evaluation of classifiers using Doc2Vect Features on TwitterHate dataset

4.4. Performance Evaluation of classifiers using Sentiment Polarity Score Features on TwitterHate dataset

Table 2 Figure 5 shows the performance of the models on the TwitterHate dataset using Sentiment Polarity Score features in terms of $Accuracy_H$, $Precision_H$, $Recall_H$ and

$F1-score_H$. The Decision Tree model has achieved the $Accuracy_H$ of 91.57%, $Precision_H$ of 90%, $Recall_H$ of 92% and $F1-score_H$ of 90%. The Logistic Regression model achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ of 90%. The XGBoost model achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ of 90%. The Random Forest model achieved the $Accuracy_H$ of 92.85%, $Precision_H$ of 90%, $Recall_H$ of 93% and $F1-score_H$ of 91%. The Extra Trees model achieved the $Accuracy_H$ of 92.68%, $Precision_H$ of 90%, $Recall_H$ of 93% and $F1-score_H$ of 91%. The AdaBoost model achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ of 90%. The Support Vector Machine model achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 87%, $Recall_H$ of 93% and $F1-score_H$ of 90%. Our proposed Multi-model model using the majority voting achieved the $Accuracy_H$ of 93.43%, $Precision_H$ of 92%, $Recall_H$ of 93% and $F1-score_H$ of 91%. It is found that, our proposed approach achieved more stable and acceptable results in comparison with aforementioned classifiers.

Table 2. Performance Evaluation of classifiers using Sentiment Polarity Score Features on TwitterHate dataset.

Model	$Accuracy_H$ **(%)**	$Precision_H$ **(%)**	$Recall_H$ **(%)**	$F1-score_H$ **(%)**
DT	91.57	90	92	90
LR	93.23	87	93	90
XGBoost	93.23	87	93	90
RF	92.85	90	93	91
ET	92.68	90	93	91
AdaBoost	93.23	87	93	90
SVM	93.23	87	93	90
Proposed Approach	93.43	92	93	91

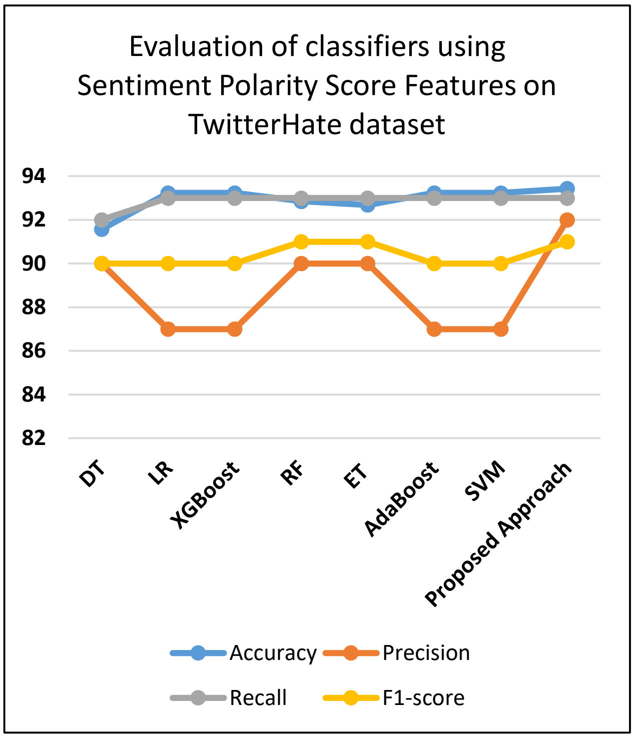

Figure 5. Evaluation of classifiers using Sentiment Polarity Score Features on TwitterHate dataset

4.5. Performance Evaluation of classifiers using TF-IDF Features on TwitterHate dataset

Table 3 and Figure 6 shows the performance of the models on the TwitterHate dataset using TF-IDF features in terms of $Accuracy_H$, $Precision_H$, $Recall_H$ and $F1-score_H$. The Decision Tree model has achieved the $Accuracy_H$ of 94.48%, $Precision_H$ of 95%, $Recall_H$ of 95% and $F1-score_H$ of 95%. The Logistic Regression model achieved the $Accuracy_H$ of 95.10%, $Precision_H$ of 95%, $Recall_H$ of 95% and $F1-score_H$ of 94%. The XGBoost model achieved the $Accuracy_H$ of 95.48%, $Precision_H$ of 95%, $Recall_H$ of 95% and $F1-score_H$ of 95%. The Random Forest model achieved the $Accuracy_H$ of 96.48%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 96%. The Extra Trees model achieved the $Accuracy_H$ of 96.51%, $Precision_H$ of 96%, $Recall_H$ of 97% and $F1-score_H$ of 96%. The AdaBoost model achieved the $Accuracy_H$ of 94.76%, $Precision_H$ of 94%, $Recall_H$ of 95% and $F1-score_H$ of 94%. The Support Vector Machine model achieved the $Accuracy_H$ of 96.17%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 96%. Our proposed Multi-model model using the majority voting achieved the $Accuracy_H$ of 96.29%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 96%. It is found that, using TF-IDF features all the models have achieved significant improvement in results in terms of accuracy, precision, recall and f1-score as compare to other features. Also, our proposed approach achieved more stable and acceptable results in comparison with aforementioned models.

Table 3: Performance Evaluation of classifiers using TF-IDF Features on TwitterHate dataset

Model	$Accuracy_H$ (%)	$Precision_H$ (%)	$Recall_H$ (%)	$F1-score_H$ (%)
DT	94.98	95	95	95
LR	95.10	95	95	94
XGBoost	95.48	95	95	95
RF	96.48	96	96	96
ET	96.51	96	97	96
AdaBoost	94.76	94	95	94
SVM	96.17	96	96	96
Proposed Approach	96.29	96	96	96

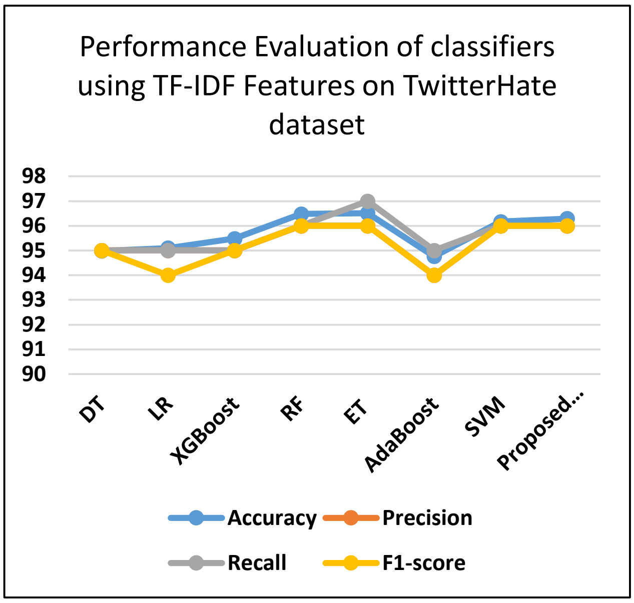

Figure 6. Evaluation of classifiers using TF-IDF Features on TwitterHate dataset

doi:10.20944/preprints202203.0333.v1

4.6. Performance Evaluation of classifiers using All Features (Doc2Vect+ Sentiment Polarity Score+ TF-IDF) on TwitterHate dataset

Table 4 and Figure 7 shows the performance of the models on the TwitterHate dataset using combination of all features (Doc2Vect+ Sentiment Polarity Score+ TF-IDF) in terms of $Accuracy_H$, $Precision_H$, $Recall_H$ and $F1-score_H$. The Decision Tree model has achieved the $Accuracy_H$ of 93.23%, $Precision_H$ of 94%, $Recall_H$ of 94% and $F1-score_H$ of 94%. The Logistic Regression model achieved the $Accuracy_H$ of 95.56%, $Precision_H$ of 95%, $Recall_H$ of 95% and $F1-score_H$ of 94%. The XGBoost model achieved the $Accuracy_H$ of 95.53%, $Precision_H$ of 95%, $Recall_H$ of 96% and $F1-score_H$ of 95%. The Random Forest model achieved the $Accuracy_H$ of 95.81%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 95%. The Extra Trees model achieved the $Accuracy_H$ of 96.50%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 96%. The AdaBoost model achieved the $Accuracy_H$ of 94.45%, $Precision_H$ of 94%, $Recall_H$ of 94% and $F1-score_H$ of 94%. The Support Vector Machine model achieved the $Accuracy_H$ of 96.09%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 96%. Our proposed Multi-model model using the majority voting achieved the $Accuracy_H$ of 96.14%, $Precision_H$ of 96%, $Recall_H$ of 96% and $F1-score_H$ of 96%. It is found that, our proposed approach achieved more stable and acceptable results in comparison with aforementioned models.

Table 4. Performance Evaluation of classifiers using All Features (Doc2Vect+ Sentiment Polarity Score+ TF-IDF) on TwitterHate dataset.

Model	$Accuracy_H$ (%)	$Precision_H$ (%)	$Recall_H$ (%)	$F1-score_H$ (%)
DT	94.23	94	94	94
LR	95.26	95	95	94
XGBoost	95.53	95	96	95
RF	95.81	96	96	95
ET	96.50	96	96	96
AdaBoost	94.45	94	94	94
SVM	96.09	96	96	96
Proposed Approach	96.14	96	96	96

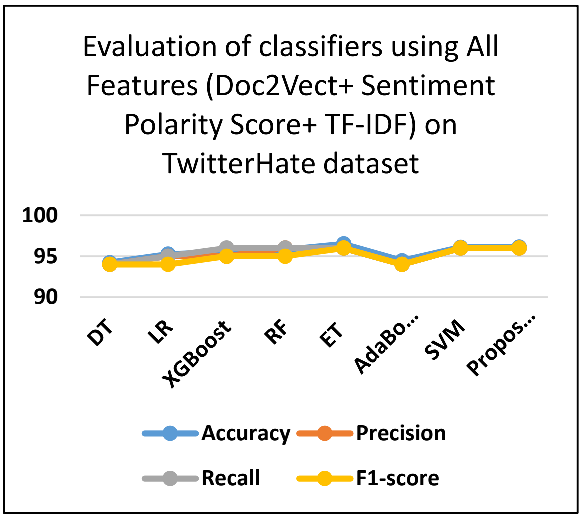

Figure 7. Evaluation of classifiers using All Features (Doc2Vect+ Sentiment Polarity Score+ TF-IDF) on TwitterHate dataset.

5. Conclusions

In this work, we proposed a multi-model learning technique for detecting Twitter hate speech. We have used Twitter Hate speech dataset, which consists of 31962 samples. The dataset consists of 93% tweets, or 29695 tweets, including non hate labeled Twitter data and 7%, or 2240 tweets, containing hate labeled Twitter data. We utilized commonly used machine learning classifiers such as Decision tree, Logistic regression, XGBoost, Random forest, Extra tree, AdaBoost, and Support vector machine to analyze the dataset. We used TF-IDF, sentiment polarity score, and doc2vector features. Our experimental results reveal that, when compared to other features, all of the classifiers achieved considerable hate speech identification using TF-IDF features, even better than all of the combined features. Using multi-model settings and TF-IDF features, we were able to achieve more consistent and acceptable detection results with accuracy of 96.29 %, precision of 96%, recall of 96%, and f1-score of 96%.

References

1. Mullah, N. S., Zainon, W. M. N. W. Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review. IEEE Access, 2021.
2. Salawu, S., He, Y. Lumsden, J. Approaches to automated detection of cyberbullying: A survey. IEEE Transactions on Affective Computing, 2017, 11(1), 3-24.
3. Chiril, P., Pamungkas, E. W., Benamara, F., Moriceau, V., Patti, V. Emotionally informed hate speech detection: a multi-target perspective. Cognitive Computation, 2022, 14(1), 322-352.
4. Laaksonen, S. M., Haapoja, J., Kinnunen, T., Nelimarkka, M., Pöyhtäri, R. The datafication of hate: expectations and challenges in automated hate speech monitoring. Frontiers in big Data, 2020, 3.
5. MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., Frieder, O. Hate speech detection: Challenges and solutions. PloS one, 2019, 14(8), e0221152.
6. Vidgen, B., Derczynski, L. Directions in abusive language training data, a systematic review: Garbage in, garbage out. Plos one, 2020, 15(12), e0243300.
7. Kovács, G., Alonso, P., Saini, R. Challenges of hate speech detection in social media. SN Computer Science, 2021, 2(2), 1-15.
8. Yin, W., Zubiaga, A. Towards generalisable hate speech detection: a review on obstacles and solutions. PeerJ Computer Science, 2021, 7, e598.
9. Schmidt, A., Wiegand, M. A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain. Association for Computational Linguistics,2019, pp. 1-10.
10. Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., Margetts, H. Challenges and frontiers in abusive content detection. In: Proceedings of the third workshop on abusive language online, 2019, pp. 80-93.
11. Abro, S., Sarang Shaikh, Z. A., Khan, S., Mujtaba, G., Khand, Z. H. Automatic hate speech detection using machine learning: A comparative study. Machine Learning, 2020, 10(6).
12. Putri, T. T. A., Sriadhi, S., Sari, R. D., Rahmadani, R., Hutahaean, H. D. A comparison of classification algorithms for hate speech detection. In: Iop conference series: Materials science and engineering (Vol. 830, No. 3, p. 032006). IOP Publishing, 2020.
13. Fortuna, P., Soler-Company, J.,Wanner, L. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets?. Information Processing & Management, 2021, 58(3), 102524.
14. Kocoń, J., Figas, A., Gruza, M., Puchalska, D., Kajdanowicz, T., Kazienko, P. Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. Information Processing & Management, 2021, 58(5), 102643.
15. Sreelakshmi, K., Premjith, B., Soman, K. P. Detection of hate speech text in Hindi-English code-mixed data. Procedia Computer Science, 2020, 171, 737-744.
16. Das, A. K., Al Asif, A., Paul, A., Hossain, M. N. Bangla hate speech detection on social media using attention-based recurrent neural network. Journal of Intelligent Systems, 2021, 30(1), 578-591.
17. Chen, T., Guestrin, C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, ACM, 2016, pp. 785-794.
18. Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., Wojatzki, M. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. arXiv preprint arXiv:1701.08118, 2017.
19. Davidson, T., Warmsley, D., Macy, M.,Weber, I. Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI Conference on Web and Social Media, 2017, Vol. 11, No. 1, pp. 512-515.
20. Badjatiya, P., Gupta, S., Gupta, M., Varma, V. Deep learning for hate speech detection in tweets. In: Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 759-760.
21. Gao, L., & Huang, R. Detecting online hate speech using context aware models. arXiv preprint arXiv:1710.07395, 2017.
22. Founta, A. M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G.,Kourtellis, N. Large scale crowdsourcing and characterization of twitter abusive behavior. In: Twelfth International AAAI Conference on Web and Social Media, 2018.

23. Zhang, Z., Luo, L. Hate speech detection: A solved problem? the challenging case of long tail on twitter. Semantic Web, 2019, 10(5), 925-945.
24. Gröndahl, T., Pajola, L., Juuti, M., Conti, M., Asokan, N. All you need is" love" evading hate speech detection. In: Proceedings of the 11th ACM workshop on artificial intelligence and security, ACM, 2018, pp. 2-12.
25. Mishra, P., Yannakoudakis, H., Shutova, E. Neural character-based composition models for abuse detection. arXiv preprint arXiv:1809.00378, 2018.
26. Kshirsagar, R., Cukuvac, T., McKeown, K., McGregor, S. Predictive embeddings for hate speech detection on twitter. arXiv preprint arXiv:1809.10644, 2018.
27. Qian, J., Bethke, A., Liu, Y., Belding, E., Wang, W. Y. A benchmark dataset for learning to intervene in online hate speech. arXiv preprint arXiv:1909.04251, 2019.
28. Mansourifar, H., Alsagheer, D., Fathi, R., Shi, W., Ni, L., Huang, Y. Hate Speech Detection in Clubhouse. arXiv preprint arXiv:2106.13238, 2021.
29. Gautam, A., Mathur, P., Gosangi, R., Mahata, D., Sawhney, R., Shah, R. R. # metooma: Multi-aspect annotations of tweets related to the metoo movement. In: Proceedings of the International AAAI Conference on Web and Social Media, 2020, Vol. 14, pp. 209-216.
30. Silva, L., Mondal, M., Correa, D., Benevenuto, F., Weber, I. Analyzing the targets of hate in online social media. In: Tenth international AAAI conference on web and social media, 2016.
31. Salminen, J., Almerekhi, H., Milenković, M., Jung, S. G., An, J., Kwak, H., Jansen, B. J. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In: Twelfth International AAAI Conference on Web and Social Media, 2018.
32. ElSherief, M., Nilizadeh, S., Nguyen, D., Vigna, G., Belding, E. Peer to peer hate: Hate speech instigators and their targets. In: Proceedings of the International AAAI Conference on Web and Social Media, 2018, Vol. 12, No. 1.
33. scikit-learn Machine Learning in Python, available online, https://scikit-learn.org/stable/
34. Quinlan, J. R. Induction of decision trees. Machine learning, 1986, 1(1), 81-106.
35. sklearn.tree.DecisionTreeClassifier, available online, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html, (accessed 15 December 2021).
36. Logistic Regression in Machine Learning, available online, https://www.javatpoint.com/logistic-regression-in-machine-learning, (accessed 15 December 2021).
37. sklearn.linear_model.LogisticRegression, available online, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html, (accessed 15 December 2021).
38. Houtao Deng, An Introduction to Random Forest, https://towardsdatascience.com/random-forest-3a55c3aca46d (accessed 15 December 2021).
39. sklearn.ensemble.RandomForestClassifier, available online, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, (accessed 15 December 2021).
40. Extra Tree Classifier, available online, https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/, (accessed 15 December 2021).
41. sklearn.ensemble.ExtraTreesClassifier, available online, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html, (accessed 15 December 2021).
42. Schapire, R. E. Explaining adaboost. In: Empirical inference. Springer, Berlin, Heidelberg, 2013, pp. 37-52.
43. sklearn.ensemble.AdaBoostClassifier, available online, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html, (accessed 15 December 2021).
44. Understanding Support Vector Machine(SVM) algorithm, available online, https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/, (accessed 15 December 2021).
45. sklearn.svm.SVC, available online, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html, (accessed 15 December 2021).
46. EnsembleVoteClassifier:https://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/, (accessed 15 December 2021).
47. sklearn.ensemble.VotingClassifier, available online, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html, (accessed 15 December 2021).
48. Twitter hate speech, available online, https://www.kaggle.com/vkrahul/twitter-hate-speech, (accessed 15 December 2021).
49. Saito, T. and Rehmsmeier, M. Basic Evaluation Measures from the Confusion Matrix. https://classeval.wordpress.com/%20introduction/basic-evaluation-measures/, (accessed 15 December 2021).