SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I12P116
Volume 11 Issue 12, 164-170, December 2024


*Original Article*

# A Novel Feature Extraction Classifier for Hateful, Offense, and Neutral Content on X

Anjani Kumar

*Cluster Innovation Centre, University of Delhi, Delhi, India.*

*Corresponding Author : anjaniverma29@gmail.com*

Received: 21 October 2024 Revised: 25 November 2024 Accepted: 12 December 2024 Published: 30 December 2024

***Abstract*** *- With the rapid growth of online social networks, content censorship remains controversial, dividing people into two groups, one supporting hateful content and the other supporting neutral content. This paper addresses the problem of classifying a tweet as hateful, offensive, or neutral content, which uses Term Frequency Inverse Document Frequencies (TFIDFs) for feature extraction. It uses the X dataset to train the proposed classifier model, and the results show that Gaussian Naive Bayes is the best-performing model after hyperparameter tuning of TFIDF features.*

***Keywords*** *- IDF, X, Hate speech, Offensive, Data cleaning.*

## 1. Introduction

Micro-blogging websites and online social networks are much more popular among Internet users than other websites. People of various ages, ethnicities, and hobbies increasingly use the services that Twitter, Facebook, and Instagram offer. Their rapidly growing content is an intriguing example of big data. Researchers interested in understanding people's opinions, users' sentiments, and interests have been attracted to this type of big data. Although these websites provide a public forum for people to express their ideas and beliefs, it is nearly impossible to police the posted content. Taking advantage of this, people with different backgrounds, cultures, and beliefs tend to use aggressive and hateful language [1]. Nowadays, with the growth of online social networks and increasing conflicts around the world, content censorship remains a controversial topic, dividing people into two groups, one supporting it and the other opposing it. It is even easier to spread such trends among young and older generations to other "cleaner" speeches. For these reasons, [1] argues that collecting and analyzing temporal data allows decision-makers to study the escalation of hate crimes following "trigger" events. However, official information regarding such events is scarce, given that hate crimes are often not reported to the police. Social networks [2] in this context present a better and more rich, yet untrustworthy and full of noise, source of information. To overcome noise and the unreliability of data, an efficient way to detect hateful and offensive posts in the data collected from social networks is required. To tackle this issue, "toxic language" must be defined. Toxic language [3] is segregated into two categories: hate speech and offensive language. "Hate speech" is defined as "any speech that attacks a person or a group based on attributes such as religion, race, ethnicity, gender, gender identity, sexual orientation, or disability." Offensive language [3] can be described as a text that includes abusive slurs or derogatory expressions. Filtering hateful tweets [4] manually is not scalable, urging researchers to identify automated alternatives. Most of the earlier work revolves either around manual feature extraction or the use of representational learning methods, which are then followed by a linear classifier. This work classifies a tweet as hateful, offensive, or neutral [20]. The task is challenging due to the intrinsic complexity of the natural language constructs; there are various forms of hatred, hate tweets are targeted at different targets, and the same meaning can be represented in various ways. This paper addresses the task of text classification in terms of hateful content, which uses

- A language-agnostic solution that does not use pre-trained word embedding [5]
- An experiment with the model on an X dataset was conducted to determine its performance on the classification task.

The proposed solution uses the X dataset to train the classifier model using Term Frequency Inverse Document Frequency (TFIDF) for feature extraction. The results show that after hyperparameter tuning of TFIDF features, the Gaussian Naive Bayes is the best-performing model.

The remainder of this paper is organized as follows: Section 2 presents recent work in the background. Section 3 explains the proposed method, and Section 4 describes the experiments and results in detail. Finally, the paper is concluded with future scope in Section 5.

## 2. Background

The problem is primarily portrayed as a supervised document classification task by the existing methods, which fall into two categories: the classic methods, which use manual feature engineering that is then used by algorithms like SVM, Naive Bayes, and Logistic Regression, and the deep learning methods, which use neural networks to learn features from raw data automatically. [6], divided the classification job into two classes using n-gram, linguistic, syntactic, and pre-trained "word2vec and comment2vec" features and achieved an accuracy of 90 [9]. Some projects focus on identifying hate speech on Twitter. [7] used unigram features to target the identification of hostile tweets toward Black individuals, with a binary classification accuracy of 76%. A certain gender, ethnic group, race, or other attribute that made the gathered unigrams connected to that particular group was determined to be the focus of the hate speech.

This makes it more difficult to identify hate speech against other groups using the built-in Unigram lexicon. [2], used word relationships and "bag of words" (BoW) properties to identify hate speech. Since 2018, many studies have addressed cross-dataset generalization due to the rise of datasets on hate speech and foul language. [8], developed a variety of models and applied them to four different datasets.

Among the models were CNN-GRU [9], which performed better than earlier models on six datasets [11], and LSTM, one of the most widely used neural networks in text classification. Using character-level features makes the models systematically more attack-resistant than using word-level features [10], and experiments demonstrate that adversarial training does not fully mitigate the attacks. The challenges in existing research are as follows:

1. It is difficult because language analysis of common datasets reveals that hate speech lacks distinctive and discriminatory characteristics.
2. Out-of-Vocabulary (OOV) terms are a potential problem with pre-trained embedding, particularly on Twitter data, because of the nature of tweets. Preprocessing will, therefore, be carried out in a way that assists in lowering the language's noise and, consequently, the OOV scale.
3. This paper will refrain from dividing lengthy tweets into two during the preprocessing stage because it has been demonstrated that doing so results in the loss of linguistic information, even though tweets hardly ever contain two complete sentences.
4. Training using domain-specific data is anticipated to improve performance on tasks such as hate speech detection [12]. Nevertheless, the findings of earlier studies indicated little advancement and did not demonstrate significant gains in feature capture. As a result, domain-specific corpora are not used to train the model.

## 3. Materials and Methods

Initially, the dataset was preprocessed, making the experiment cleaner. Based on past work results, this paper extracts the features using n-gram from the input text and weights them using TFIDF. Then, these features are fed to different classification heads to compare their performances. Let us start with TF in TFIDF, which means term frequency. In document "d," the frequency represents the number of instances of a given word "t." Therefore, it becomes more relevant when a rational word appears in the text. Since the ordering of terms is insignificant, a vector can describe the text in the bag of term models. Each specific term in the paper has an entry with the term frequency value. The weight of a term that occurs in a document is simply proportional to the term's frequency [19].

$$tf(t,d) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d} \tag{1}$$

Now, let us move on to the IDF in the TFIDF, which stands for Inverse Document Frequency. To understand inverse document frequency, let us first understand document frequency. Document Frequency (DF) tests the meaning of the text, which is very similar to TF, in the whole corpus collection. The only difference is that in document d, TF is the frequency counter for a term "t," while DF is the number of occurrences of the term "t" in the document set N. In other words, the number of papers in which the word is present is DF.

$$df(t) = occurrence\ of\ t\ in\ documents \tag{2}$$

Thus, inverse document frequency mainly tests how relevant the word is. The key aim of the search is to locate the appropriate records that fit the demand. Because TF considers all terms equally significant, the term frequencies can be used to calculate the weight of the term in the paper. First, find the document frequency of a term by counting the number of documents containing the term.

$$df(t) = N(t) \tag{3}$$

Where df(t) is the document frequency of the term "t", and N(t) is the number of documents containing the term "t." Term frequency is only the number of term instances in a single document. However, the frequency of the document is the number of separate documents in which the term appears, and it depends on the entire corpus. Now, let us look at the definition of the frequency of the inverse paper. The IDF of the word is the number of documents in the corpus separated by the frequency of the text [18].

$$idf(t) = \frac{N}{df(t)} = \frac{N}{N(t)} \tag{4}$$

The more common word is supposed to be considered less significant, but the element (the most definite integer) seems too harsh. After that, take the logarithm (with base 2) of the inverse frequency of the paper. So the idf of the term "t" becomes:

$$idf(t) = log\left(\frac{N}{df(t)}\right) \tag{5}$$

Finally, TF-IDF can be computed by multiplying the Term Frequency and Inverse Document Frequency:

$$tfidf(t,d) = tf(t,d) * idf(t) \tag{6}$$

All the clean tweets were combined into one giant corpus and then weighted each element in the corpus by its TF-IDF. A variation for feature extraction is tried wherein, instead of extracting features directly from the corpus, the hate level of each word present in the corpus and weighing TFIDF against those features are identified. (Figures 1 and 2) classified the most frequently occurring words obtained from the dataset as hate or non-hate [16], generated using the WordCloud library in Python:

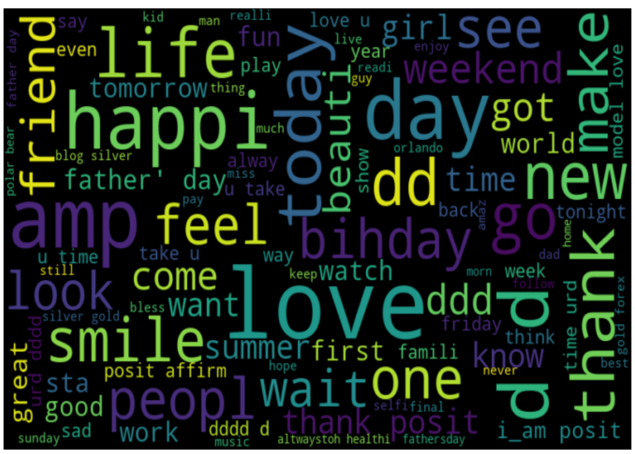

**Fig. 1 Non-hate words**

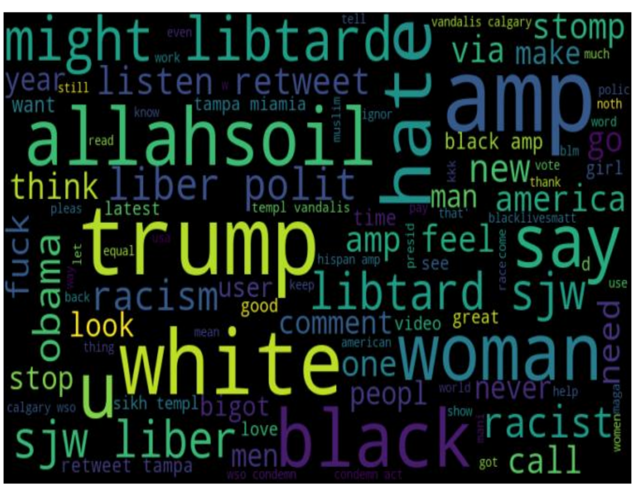

**Fig. 2 Hate words**

This section used the five machine learning algorithms [13] to compare and provide a mathematical foundation for how they work. Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurring. It is a special linear regression case where the target variable is categorical. It uses a log of odds as the dependent variable. Using a logit function, logistic regression predicts the likelihood of occurrence of a binary event.

$$y = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_n X_n \tag{7}$$

Where, y is a dependent variable and $X_1$, $X_2$ ,.., $X_n$ are explanatory variables.

The Sigmoid Function:

$$p = \frac{1}{1+e^{-y}} \tag{8}$$

Applying Sigmoid Function on Linear Regression:

$$p = \frac{1}{1+e^{b_0+b_1X_1+b_2X_2+\ldots+b_nX_n}} \tag{9}$$

A variation of Naive Bayes that supports continuous data and adheres to the Gaussian normal distribution is called Gaussian Naive Bayes. The Bayes theorem serves as the foundation for naive Bayes classifiers. Naive Bayes classifiers are based on the Bayes theorem. The strong independence assumptions between the features are one of the assumptions made. These classifiers assume that a feature's value is unrelated to any other feature's value. Naive Bayes classifiers are highly effective at training in supervised learning scenarios. To estimate the parameters required for classification, naive Bayes classifiers require a small amount of training data. A common presumption when working with continuous data is that each class's continuous values are distributed using a normal (or Gaussian) distribution. It is assumed that the features' likelihood is:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - m_y)^2}{2\sigma_y^2}\right) \tag{10}$$

The variance is sometimes assumed to be independent of Y (i.e., i), or independent of $X_i$ (i.e., k) or both. Gaussian Naive Bayes is a supervised learning technique that supports continuous valued features and models. Decision trees are tree-structured classifiers used for classification and regression problems, with decision nodes making decisions and leaf nodes representing outcomes. Random Forest is a machine-learning algorithm based on ensemble learning, combining multiple classifiers to improve model performance. It works in two phases: creating the random forest and making predictions for each tree. Gradient boosting classifiers combine weak learning models to create a strong predictive model [15], often using decision trees. Gradient boosting models are popular for classifying complex datasets and depend on a differentiable loss function. Custom loss functions can be used, and gradient boosting systems do not need to derive new loss functions every time the boosting algorithm is added.

## 4. Results and Discussion

### *4.1. Dataset*

The dataset contains 31962 tweets. After running a quick Pandas profile, the following results are as follows: the "label" column represents the categorization of tweets. 0 represents a clean tweet, whereas 1 represents a hateful tweet. The ratio of hateful tweets to not-hateful tweets is 2242:29720, or approximately 1:13, which means every 1 in 14 tweets is offensive. The dataset structure is shown in Table 1.

**Table 1. Head of the Raw Dataset**

| | Id | Label | Tweet |
|---|---|---|---|
| 0 | 1 | 0 | @user when a father is dysfunctional and is s… |
| 1 | 2 | 0 | @user @user, thanks for #lyft credit I can’t us… |
| 2 | 3 | 0 | bihday your majesty |
| 3 | 4 | 0 | #model i love u take with u all the time in…[14] |
| 4 | 5 | 0 | factsguide society now #motivation [14] |

The Pandas Profiling Report and overview report can be seen in Figures 3 and 4.

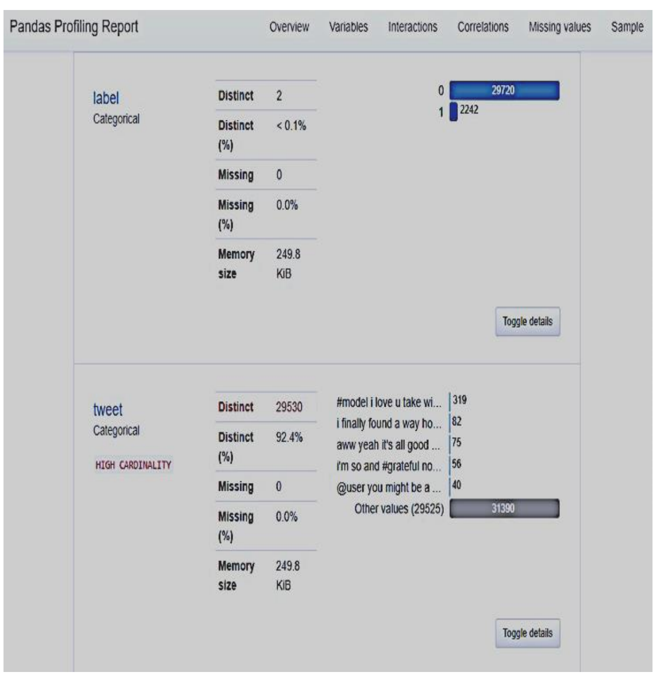


**Fig. 3 Pandas profiling report**

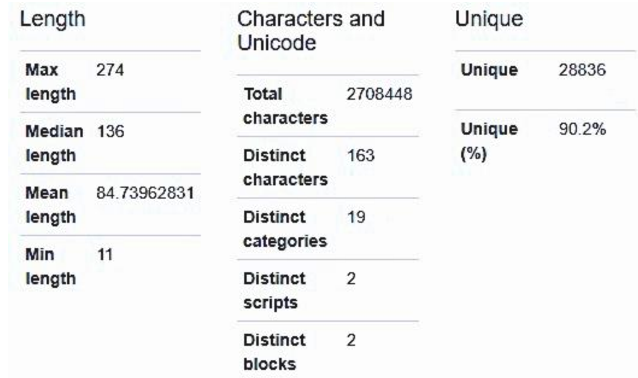


**Fig. 4 Tweet column overview**

The dataset is preprocessed by removing "@" tags and "words" and making the sentences all lowercase to achieve the final dataset (see Table 2).

### *4.2. Experimental Setup*

The scikit-learn [10] library in Python is used for training and experimentation. Jupyter Lab is used to execute the experiments. TFIDF is used for feature extraction from a document or a set of words.

**Table 2. Head of final clean tweet dataset**

| | Label | Tweet |
|---|---|---|
| 1 | 0 | when a father is dysfunctional and is so selfi…[14] |
| 2 | 0 | thanks for the credit I can’t use cause they don’t…[14] |
| 3 | 0 | bihday your majesty |
| 4 | 0 | i love u take with urd+-!!! |
| 5 | 0 | factsguide: society now |

*4.2.1. Experiment A*

A corpus of tweets used for feature extraction using TF-IDF is presented in Figure 5.

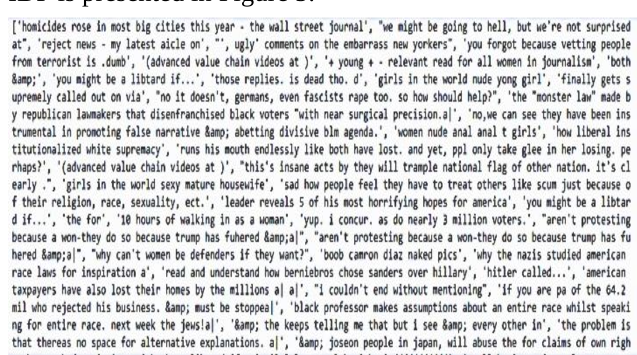
['homicides rose in most big cities this year - the wall street journal', "we might be going to hell, but we're not surprised at", 'reject news - my latest aicle on', "', ugly' comments on the embarrass new yorkers", 'you forgot because vetting people from terrorist is .dumb', '(advanced value chain videos at )', '+ young + - relevant read for all women in journalism', 'both &', 'you might be a libtard if...', 'those replies. is dead tho. d', 'girls in the world nude yong girl', 'finally gets supremely called out on via', "no it doesn't, germans, even fascists rape too. so how should help?", 'the "monster law" made by republican lawmakers that disenfranchised black voters "with near surgical precision.a|', 'no,we can see they have been instrumental in promoting false narrative & abetting divisive blm agenda.', 'women nude anal anal t girls', 'how liberal institutionalized white supremacy', 'runs his mouth endlessly like both have lost. and yet, ppl only take glee in her losing. perhaps?', '(advanced value chain videos at )', "this's insane acts by they will trample national flag of other nation. it's clearly .", 'girls in the world sexy mature housewife', 'sad how people feel they have to treat others like scum just because of their religion, race, sexuality, ect.', 'leader reveals 5 of his most horrifying hopes for america', 'you might be a libtard if...', 'the for', '10 hours of walking in as a woman', 'yup. i concur. as do nearly 3 million voters.', "aren't protesting because a won-they do so because trump has fuhered &a|", "aren't protesting because a won-they do so because trump has fuhered &a|", "why can't women be defenders if they want?", 'boob camron diaz naked pics', 'why the nazis studied american race laws for inspiration a', 'read and understand how berniebros chose sanders over hillary', 'hitler called...', 'american taxpayers have also lost their homes by the millions a| a|', "i couldn't end without mentioning", 'if you are pa of the 64.2 mil who rejected his business. & must be stoppea|', 'black professor makes assumptions about an entire race whilst speaking for entire race. next week the jews!a|', '& the keeps telling me that but i see & every other in', 'the problem is that thereas no space for alternative explanations. a|', '& joseon people in japan, will abuse the for claims of own righ

**Fig. 5 Corpus head**

The extraction features from the above corpus using TF-IDF are done. After feature extraction, the feature dataframe is obtained. Then split the feature dataframe into 2 categories: hate features with label = 1 and non-hate features with label = 0. After splitting the features, the training set contained 90 percent of all the hate features and 40 percent of all the non-hate features. The proportion is such that the train set overall contained a balanced proportion of hate to non-hate tweets. The ROC scores obtained are shown in Table 3.

**Table 3. Results of Experiment A**

| Classifier | Accuracy Score | ROC AUC Score |
|---|---|---|
| Logistic Regression | 0.804 | 0.785 |
| Gaussian NB | 0.547 | 0.667 |
| Random Forest | 0.759 | 0.754 |
| Decision Tree | 0.668 | 0.691 |
| Gradient Boosting | 0.791 | 0.710 |

The ROC curve for the classifiers using the previously-mentioned ROC scores is depicted. To briefly explain ROC AUC scores, an ROC curve, also known as a receiver operating characteristic curve, is a graph that displays a classification model's performance overall classification thresholds. The False Positive Rate (FPR) and True Positive Rate (TPR) are plotted. The acronym AUC represents "Area Under the ROC Curve." That is, AUC calculates the total two-dimensional area from (0,0) to (1,1) beneath the complete ROC curve (think integral calculus). AUC (Area Under the ROC Curve) is shown in Figure 6.

AUC provides an overall performance metric across all potential classification criteria. Classifiers that produce curves toward the upper-left corner indicate better performance. As the curve approaches the 45-degree diagonal, the test is less accurate.

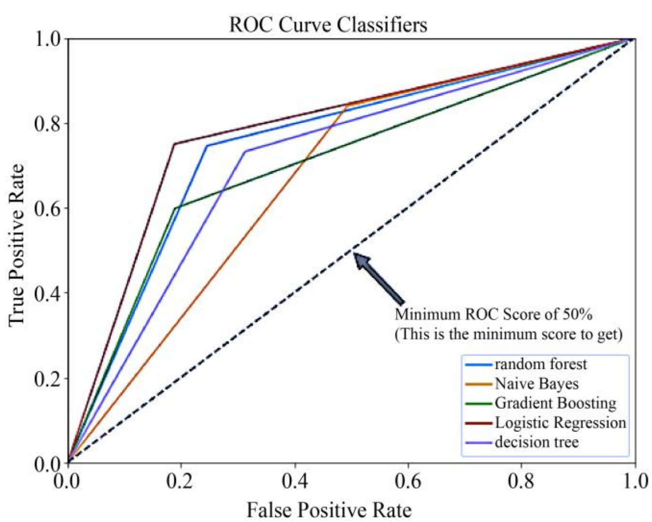


**Fig. 6 ROC Curve of Experiment A**

As the ROC scores and ROC curves show, logistic regression produces the best results for hate speech detection.

*4.2.2. Experiment B*

Instead of directly extracting features from the corpus, the corpus is split further into words. Then, the 1000 most frequently occurring words in the dataset (see Figure 7) are obtained using a counter. Now, assign a hate level to each of these words. Since each tweet was classified using labels, that classification lost its semantic sense when splitting the tweets into words. For example, the word "the" occurs in a hate tweet with a label = 1 and a non-hate tweet with a label = 0. The concept of hate level is brought up to decide whether the word "the" can be classified as hate.

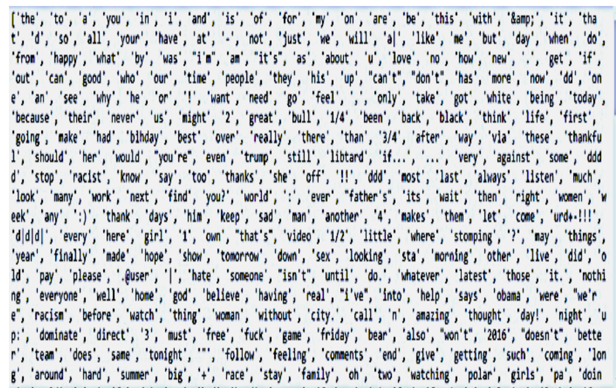


**Fig. 7 Head of the Top 1000**

Each word was assigned a hate level relative to the maximum hate level possible. So, for example, if a word occurs in 100 tweets, with 80 being hate tweets, the word will be classified as having a hate level greater than 0.5.

Assigning 0.5 as the breaking point for the experiment that decides a word having a hate level greater than 0.5 will be classified as a hate word, and any word having a hate level of 0.5 or less will be classified as a non-hate word. The features are extracted using TF-IDF based on the hate levels assigned to the words. The following ROC scores were obtained in Table 4.

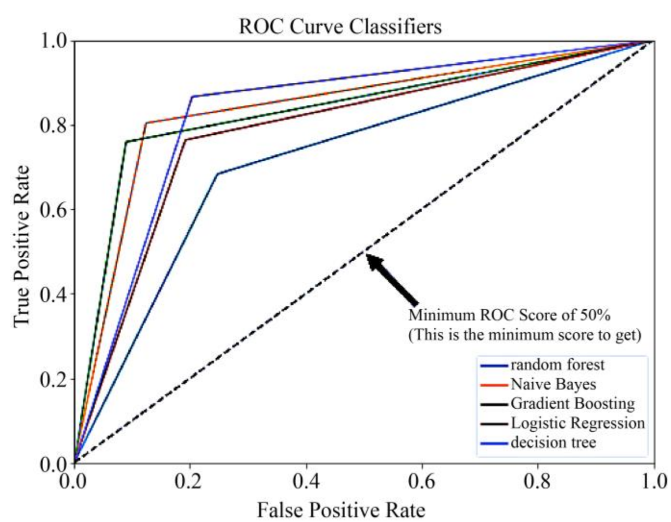


**Fig. 8 ROC Curve of Experiment B**

**Table 4. Results of Experiment B**

| Classifier | Accuracy Score | ROC AUC Score |
|---|---|---|
| Logistic Regression | 0.741 | 0.785 |
| Gaussian NB | 0.870 | 0.839 |
| Random Forest | 0.747 | 0.717 |
| Decision Tree | 0.801 | 0.831 |
| Gradient Boosting | 0.899 | 0.834 |

Using the above ROC scores, the ROC curve for the classifiers is plotted in Figure 8.

## 5. Conclusion

Existing hate speech detection models perform poorly on new, previously unseen datasets. This is due to the limitations of existing NLP methods, the difficulty of constructing datasets, and the nature of online hate speech, which are frequently interrelated. The behavior of social media users, and particularly haters, poses an added challenge to existing NLP approaches. Feature engineering and dataset extraction are major in determining the outcome.

In the future, the aim is to build a richer dictionary of hate speech patterns that can be used, along with unigram, bigram, and trigram features that include users' tendencies to post hateful and offensive online texts. The major plan is to experiment with state-of-the-art deep learning architectures like LSTM and GRU to increase accuracy. The work also intends to use larger datasets from across the web to train and test the classifiers and investigate the rise and fall of cyber-hatred on online social media platforms.

## Acknowledgments

Thanks to the students, namely, Ishan, Mehak, and Jayshree, for their valuable support in completing this article.

## References

[1] Pete Burnap, and Matthew L. Williams, "Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making," *Policy and Internet*, vol. 7, no. 2, pp. 223-242, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[2] Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," *IEEE Access*, vol. 6, pp. 13825-13835, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] William Warner, and Julia Hirschberg, "Detecting Hate Speech on the World Wide Web," *Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012)*, Montreal, Canada, pp. 19-26, 2012. [Google Scholar] [Publisher Link]

[4] Nemanja Djuric et al., "Hate Speech Detection with Comment Embeddings," *WWW '15 Companion: Proceedings of the 24th International Conference on World Wide Web*, New York, United States, pp. 29-30, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth, "Effective Hate-Speech Detection in Twitter Data Using Recurrent Neural Networks," *Applied Intelligence*, vol. 48, no. 12, pp. 4730-4742, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Njagi Dennis Gitari et al., "A Lexicon-Based Approach for Hate Speech Detection," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 215-230, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Chikashi Nobata et al., "Abusive Language Detection in Online User Content," *Proceedings of the 25th International Conference on World Wide Web*, Switzerland, pp. 145-153, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[8] Irene Kwok, and Yuzhou Wang, "Locate the Hate: Detecting Tweets Against Blacks," *In Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 27, no. 1, pp. 1621-1622, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[9] Garima Koushik, K. Rajeswari, and Suresh Kannan Muthusamy, "Automated Hate Speech Detection on Twitter," *2019 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, India, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Tommi Gröndahl et al., "All You Need is "love": Evading Hate Speech Detection," *AISec '18: Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, New York, United States, pp. 2-12, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ziqi Zhang, David Robinson, and Jonathan Tepper, "Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network," *The Semantic Web: 15th International Conference, ESWC 2018*, Heraklion, Crete, Greece, vol. 10843, pp 745-760, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Wenjie Yin, and Arkaitz Zubiaga, "Towards Generalisable Hate Speech Detection: A Review on Obstacles and Solutions," *PeerJ Computer Science*, vol. 7, pp. 1-38, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Dan Nelson, Gradient Boosting Classifiers in Python with Scikit-Learn, Stack Abuse, 2023, [Online]. Available: https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/

[14] Hemant Kumar Soni, Sanjiv Sharma, and G. R. Sinha, *Text and Social Media Analytics for Fake News and Hate Speech Detection*, 1st ed., A Chapman and Hall Book, CRC Press, pp. 1-324, 2024. [Google Scholar] [Publisher Link]

[15] Supriya Raheja, "Analysis of Psychological Distress during COVID-19 among Professionals," *International Journal of Software Innovation (IJSI)*, vol. 10, no. 1, pp.1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Anjum, and Rahul Katarya, "Hate Speech, Toxicity Detection in Online Social Media: A Recent Survey of State of The Art and Opportunities," *International Journal of Information Security*, vol. 23, pp. 577-608, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Patil Sushma, "Text Mining - A Comparative Review of Twitter Sentiments Analysis," *Recent Advances in Computer Science and Communications*, vol. 17, no. 1, pp. 21-37, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Chenxuan Yang et al., "Timothy Barnett, Analysis of First Responder-Involved Traffic Incidents By Mining News Reports," *Accident Analysis & Prevention*, vol. 192, 2023.[CrossRef] [Google Scholar] [Publisher Link]

[19] Abdullah Havolli, Arianit Maraj and Lorik Fetahu, "Building A Content-Based Recommendation Engine Model Using Adamic Adar Measure; A Netflix Case Study," *11th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, pp. 1-8, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Pinkesh Badjatiya et al., "Deep Learning for Hate Speech Detection in Tweets," *WWW '17 Companion: Proceedings of the 26th International Conference on World Wide Web Companion*, Switzerland, pp. 759-760, 2017. [CrossRef] [Google Scholar] [Publisher Link]