Int. j. inf. tecnol. (December 2022) 14(7):3629–3635
https://doi.org/10.1007/s41870-022-01096-4

ORIGINAL RESEARCH

Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets

Stephen Akuma[1] · **Tyosar Lubem**[1] · **Isaac Terngu Adom**[1]

Received: 2 June 2022 / Accepted: 9 September 2022 / Published online: 21 September 2022

Abstract Social media platforms such as Twitter have revolutionized online communication and interactions but often contain components of disdain for its growing user base. This discomforting feed creates instability leading to mental breakdown, and loss of human lives and properties among other results of misuse. Even though the problem posed by the content of social media is obvious, the challenge of detecting hateful content persists. Several algorithms and techniques have been used in the past for detecting hateful content on social media but there is room for improvement. The goal of this paper is to detect hate speech from live tweets on Twitter via a combination of mechanisms. The comparison results of Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) with machine learning models of Logistic Regression, Naïve Bayes, Decision Tree, and K-Nearest Neighbour (KNN), is used to select the best performing model. This model which is integrated into a web system developed with Twitter Application Programming Interface (API) is used in identifying live tweets which are hateful or not. The outcome of the comparative study presented showed that Decision Tree performed better than the other three models with an accuracy of 92.43% using TF-IDF which gives optimal results compared to BoW.

Keywords Social media · Sentiment analysis · Hate speech · Twitter · Machine learning algorithm · Bag of Words · TF-IDF

1 Introduction

There has been an exponential growth of users of online forums like Facebook, Twitter, and Instagram. About 350,000 tweets are generated on Twitter and 50,000 comments are generated on Facebook per second [11]. Participants of these forums come from different races, cultures and educational backgrounds and their opinions, criticism, and personal feelings are all expressed through these platforms. Lack of regulation of freedom of speech on the web often leads some users of social media platforms to hide their identity and use derogatory and offensive words on people. These derogatory words meant to cause psychological damage to users are often referred to as hate speech. There is no consensus definition for hate speech. Attempts have been made to define hate speech based on acts of violence or a prejudiced environment that may promote a violent act on a person or group [15]. In Ref. [18] opined that Hate speech is an offensive language which can be nasty, disrespectful, or harmful to an online and offline individual or communities or a society as a whole. In Ref. [7] presented hate speech as a type of communication that criticizes or dismisses groups based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity, or other factors, and it can take many forms, including subtle forms such as humour and jokes. According to Twitter, hate speech is communication that incites violence against others and directly targets or threatens them because of their ethnicity, race, nationality, age, gender, religious affiliation, sexual

✉ Stephen Akuma
sakuma@bsum.edu.ng

Tyosar Lubem
tyosarlubem9@gmail.com

Isaac Terngu Adom
iadom@bsum.edu.ng

[1] Department of Mathematics, Computer Science and Statistics, Benue State University, Makurdi, Nigeria

orientation, handicap, or major illness. In Ref. [10] defined hate speech as "any statement that promotes violent crime, attacks or seeks to silence a minority, uses a racial or sexist slur; criticizes a minority irrationally; contains stereotyping of a particular minority; and defends sexism, racism, xenophobia, or any other dangerous extremism".

These uncensored behaviour has increased within the last decade and manually detecting and removing such injurious messages and comments from social media is a tedious task to undertake. When automated techniques are used, they can quickly classify hate speech and protect online users from social hate speech harassment [18]. Research has been conducted in the last decade on how to use automated systems to detect hate speech [8, 11, 18]. For instance, Artificial intelligence technology is already being used by companies like Facebook and Google but there are still challenges in detecting some hate words. The difficulty with automatically detecting hate tweets is that the language and expressions have a format that is difficult to annotate [17]. For instance, character redundancies (e.g., kiiiiiind, caaaaar) and unnecessary intonation and exclamations (e.g., Coming…!!!!, yes????) are some of the word abbreviations and expressions overused. As a result, to obtain a format that keeps the original meaning, the source text must be changed through a vital preprocessing process that is compatible with other comparable posts [8]. This makes detecting hate speech from tweets problematic for both robots and humans, as it is exceedingly difficult to distinguish hate speech from other potentially harmful content that does not fall into the hate tweet category [17]. Other research has used traditional machine learning techniques and surface characteristics such as word, deep learning, Term Frequency-Inverse Document Frequency (TF-IDF), Doc2Vec, and character n-grams for hate speech detection [11].

In this research, we presented a mechanism for identifying hate speech on Twitter that can effectively differentiate between a hate word and a non-hate word. We used a publicly available Twitter dataset to train our classifier model using Bag of Words and TF-IDF and evaluated the system with standard evaluation metrics. A comparative analysis of the results obtained from several classifiers was also carried out and it was found that Decision Tree performed better than other classifiers for TF-IDF. The strategy that comes closest to ours is proposed by [11], which uses TF-IDF and weighted n-gram values. This paper's main contributions are as follows: (1) Analysis of the efficiency of BoW with TF-IDF using various machine learning methods in detecting hate speech; (2) Hate speech identification on Twitter using machine learning techniques; (3) Evaluation of the model using live tweets. The rest of the paper is organized into the following: Sect. 2 presents related work summarizing past research in detecting hate speech. Section 3 provides a detailed description of our approach. The approach is tested and the result is discussed in Sect. 4. Section 5 is the conclusion, and it summarizes our findings and makes recommendations for future research.

2 Related work

A lot of research work on sentiment analysis has been conducted as users of online forums increase [2]. In Ref. [14] researched the polarization of Twitter sentiments and they classified sentiments using emotions from Plutchik's wheel of emotion. Long Short-Term Memory (LSTM) model was used for sentiment analysis in social data by [21]. They obtained an accuracy of 87% in reviewing e-commerce products in Hindi language. The same LSTM approach was fused with an attention encoder to analyze sentiments [19]. They inferred that their aggregated method outperformed the baseline model used. Deep learning approaches are also used for sentiment analysis through a paragragh2vec and Convolution Neural Network (CNN) approach for the classification of hate words [9].

Several works on hate speech detection, some of which are addressed in this study, use more than two classification techniques for computational comparison to determine which method has the best detection performance and accuracy. In Ref. [15] built a model to identify and detect hate speech using a Linear Support Vector Machine with three parameters, including Brown groups, surface n-grams, and word skip-grams. In Ref. [1] worked on the hate speech dataset and they conducted a performance evaluation of feature extraction techniques with several machine learning models. Their study divided tweets under study using the following grouping: hate speech, offensiveness, and neither. Data gathering, data preprocessing, feature engineering, data splitting, classification model design, and classification model evaluation were all part of their methodology used in obtaining reasonable results.

In Ref. [17] described HaterNet, a smart system set up by the Spanish government to combat hate crimes and identify and track the rise of antagonistic relations on Twitter. This measure provides the first intelligent system that uses social network analysis methods to track and show hate speech on social networking sites. It evaluates and contrasts many classification algorithms based on different document representation strategies and text classification methods. They obtained an Area Under the Curve (AUC) of 0.828. In Ref. [5, 13] employed TF-IDF vectorization and word embeddings to extract features in their studies, using various classification algorithms to compare their performance. A hybridization approach of deep learning and TF-IDF was used by [13] to improve document classification. Their result outperformed the traditional classifiers, depicting high accuracy in classifying documents based on the texts. In

Ref. [20] developed a system for detecting hate speech from comments and posts from major social media platforms, as well as remarks and stories from a list of internet sites. The researchers' major goal was to develop software for searching, evaluating, and saving multi-source media and social media posts, with a focus on anti-migrant and anti-refugee hate speech. They created a Natural Language Processing (NLP) script as well as a web service that allows for friendly user interaction.

To identify offensive words in tweets, [8] utilized Linear SVM and Naive Bayes classifiers. The data used in the training procedure was demonstrated to be quite sensitive to the Linear SVM. Data normalization with tags was discovered to make the parameter regulating process more challenging. In Ref. [11] suggested using n-gram traits weighted with TF-IDF values, as well as three prominent machine learning algorithms (SVM, Naive Bayes, and Logistic Regression) to detect hate speech and provocative language on Twitter. They used a grid search for all possible feature parameter combinations and tenfold cross-validation to train each model. The average cross-validation score for each combination of feature parameters was used to evaluate the performance of each algorithm. Their results show that for L2 normalization, SVM performs badly when compared to Naive Bayes and Logistic Regression. However, Logistic Regression performs better with the appropriate n-gram range of 1 to 3 for the L2 normalization of TF-IDF which has 95.6% accuracy.

To minimize the data's dimensionality, [6] created a method that used Logistic Regression with L1 regularization. They used Logistic Regression, Naive Bayes, Decision Tree, Random Forests, and Linear SVMs to test a variety of models. They found that Logistic Regression and Linear SVM performed much better than other models. In Ref. [12] developed a model for detecting hate speech in Amharic language using a combination of RNN, LSTM and Gated Recurrent Unit (GRU) techniques. Data labelling, cleaning, normalization and tokenization of the Facebook posts which were used as the primary source was carried out. Word2vec word embeddings and features extraction method were used and they obtained an improved accuracy of the result. Other researchers have employed word representations or embeddings for detecting hate speech [16]. Table 1 is the tabulated review of some of the literature used in this work.

3 Methodology

We used a machine learning model to distinguish between hateful or toxic words and non-hateful language. Preprocessing and feature extraction are carried out on a dataset that has been annotated and made public through Kaggle [3], and a comparative analysis is carried out using a BoW and TF-IDF. The accuracy, F-measure, recall, confusion matrix, and precision of the results generated from the four selected models—Logistic Regression, Naive Bayes, Decision Tree, and K-Nearest Neighbor were analyzed. The Twitter API is used to query or retrieve user tweets from Twitter to detect hate speech or non-hate speech to assess the accuracy of the chosen model. Figure 1 shows the system architecture.

Table 1 Tabulated literature review

S/N	Author(s) and Year	Methodology	Result	Limitations
1	[14], 2022	Plutchik's wheel of emotions and Rule-Based Classification Algorithm	Classification of sentiments from tweets	Plutchik's wheel of emotion has limited dictionary words and this might have affected the prediction accuracy
2	[21], 2022	LSTM and BoW used for sentiment analysis	87% accuracy of Hindi-based sentiment analysis	The domain was limited to E-commerce product reviews
3	[1], 2020	Combined feature engineering and ML algorithms	79% accuracy of hate speech detection	Lack of real-time prediction and less training data
4	[15], 2017	Linear SVM n-grams, word skip-grams and brown clusters	78% accuracy of the result	Limited classifiers, Low-scoped features
5	[17], 2019	LSTM, MLP and frequency features	The area under the curve is 0.828	Limited domain of application
6	[5], 2020	Encoder-decoder, LSTM, GRU, 1D convolutional layers, TF-IDF and word embeddings	77% accuracy	Limited Bangla dataset and testing mechanism for the model
7	[8], 2020	Naive Bayes and SVM	90% accuracy of Naive Bayes and 92% of SVM	Few classifiers used
8	[11], 2018	Logistic Regression, Naive Bayes, SVM, n-gram and TF-IDF	95.6% accuracy	The distancing of words required for n-gram takes a longer time to search through
9	[12], 2020	LSTM, GRU, n-gram and word2vec	97.9% accuracy	Model limited to Amharic text data

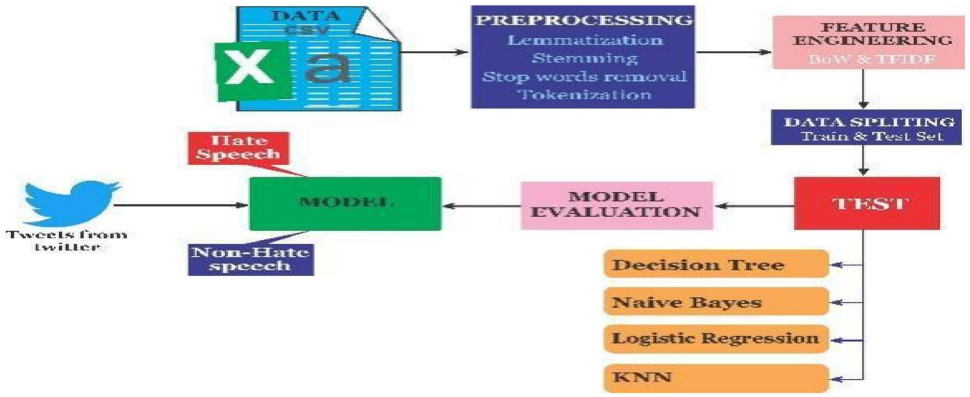


Fig. 1 The System architecture

The algorithm capturing the methodology is presented below:

Step 1 Start
Step 2 Preprocess dataset
Step 3 Build the model using BoW and TF-IDF as features
Step 4 Evaluate with ML algorithms and select the best fit model
Step 5 Use Twitter API to read live tweet *t*
Step 6 Test *t* with the model to classify as "hate" or "non-hate"
Step 7 Stop

3.1 Dataset

This study utilizes the Kaggle Hate Speech and Offensive Language dataset [3], which was created by Andriy Samoshyn. It contains 24,784 tweets, each of which has been classified by crowdflower contributors. The dataset was gathered and annotated to detect hate speech. It distinguishes between tweets containing hate speech, tweets containing offensive language, and tweets that do not contain any offensive or hateful language.

3.2 Data processing

Stop words were removed, and stemming, tokenizing, and lemmatization were among the data preprocessing techniques employed in the study. Because Twitter user communication is occasionally informal, the data is inconsistent and noisy, necessitating its cleaning and transformation into a format that the classification model can understand.

3.3 Feature extraction techniques

Machine learning classification techniques necessitate the correct presentation of tweets, with each tweet, turned into a feature vector containing only different words. The feature vector is used as an input to the classifier, implying that a good feature vector leads to improved classification results. The feature extraction methods used are Bag of Words (BoW) and TF-IDF as explained in Sects. 3.3.1 and 3.3.2.

3.3.1 Bag of Words

This method eliminates features from textual expressions so that they can be used in modelling, such as in machine learning models. Since all information about the sequence or structure of words in a document is removed, it is described as a "bag" of words. This model just cares about whether or not known terms appear in a document, not where they appear. The goal is to turn each document into a vector that can be input into or extracted from a machine learning model. The simplest scoring approach is to assign a Boolean value to the presence of words, with 0 indicating absence and 1 indicating presence. For this work, this feature extraction technique is combined with other techniques and algorithms.

3.3.2 Term Frequency–Inverse Document Frequency (TF-IDF)

The Term Frequency–Inverse Document Frequency is a statistical method that measures how important a word is in a set of documents. This is calculated by multiplying two metrics over a series of texts: the total number of times a word appears in a document (TF) and the word's inverse document frequency (IDF). The TF-IDF is useful in machine learning models and Natural Language Processing (NLP) tasks for text analysis where the count of the occurrence of words is of paramount importance. Thus, the formula for computing the TF-IDF of term *t* present in document *d* is given in Eq. 1.

$$tf - idf\,(d,t) = tf(t) * idf(d,t) \tag{1}$$

Table 2 The first experiment conducted with a Bag of Words and the models

S/N	Algorithm	Accuracy	Recall	Precision	F-Measure
1	Naïve Bayes	25.45%	0.003	0.8	0.20
2	KNN	66.21%	0.79	0.76	0.53
3	Logistic Regression	74.79%	1.00	0.75	0.43
4	Decision Tree	67.16%	0.76	0.79	0.58

Table 3 The second experiment conducted with TF-IDF and the models

S/N	Algorithm	Accuracy	Recall	Precision	F-Measure
1	Naïve Bayes	75.27%	0.79	0.86	0.69
2	KNN	85.76%	0.88	0.91	0.81
3	Logistic Regression	90.46%	0.99	0.88	0.85
4	Decision Tree	92.43%	0.95	0.95	0.84

4 Results and discussion

There are numerous machine learning algorithms accessible for use in machine learning projects; however, for each dataset, there is a corresponding algorithm that fits better; so, several algorithms were examined to determine which algorithm has the highest performance accuracy. In light of this, four (4) machine learning algorithms, namely Naïve Bayes, Decision Tree, Logistic Regression, and K-Nearest Neighbor, were put to the test. The following results were acquired and presented in Tables 2 and 3 in terms of precision, recall, F-measure, accuracy, and confusion matrix from the study conducted on the chosen machine learning models using the TF-IDF and BoW as feature extraction methods. The experiment was run twice with multiple classifiers using the TF-IDF and BoW techniques.

4.1 Experimental Setup

The experiment was run on a PC with Windows 10 operating system, 8 GB RAM and a 4 GHz processor, however, a GPU will be quicker. The Jupyter notebook in Anaconda 3 and Google Colab were used for analysis. The data was divided into two categories: training data and test data. The models were created using Tensorflow and sci-kit learn. Packages for natural language processing were also used. For Naïve Bayes and Logistic Regression models, default parameters were used. The random state parameter for K-Nearest Neighbour is set at 3. The Decision Tree was given a random state of 1 instead of the default value of 0.

4.2 Experiment 1 results

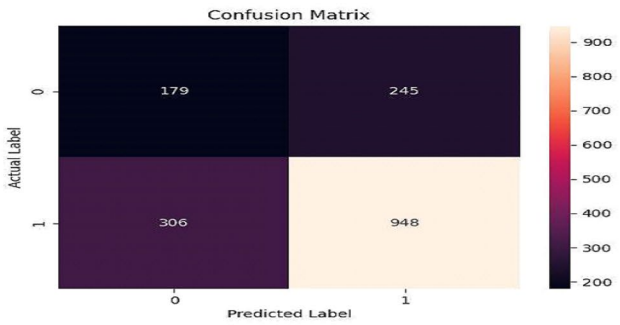
The classifiers' performance, looking at Table 2, shows that Logistic Regression obtained the highest accuracy with 74.79% compared to KNN: 66.21%, Decision Tree: 67.16% and Naïve Bayes: 25.45%. It is further observed that the Decision Tree obtained the highest recall of 0.76, precision of 0.79 and 0.58 in the F-measure. Comparing Logistic Regression and KNN, Logistic Regression performed better than KNN in recall with 1.00 while KNN got 0.79. Logistic Regression obtained 0.75 while KNN obtained 0.76. The F-measure for the two is 0.43 for Logistic Regression and 0.53 for KNN respectively. On the whole, Naïve Bayes performs poorly with an accuracy of 25.45% being the lowest accuracy while Logistic Regression obtained 74.79% to be the highest accuracy using the BoW approach. Figure 2 shows the result for the confusion matrix for the algorithms tested with BoW.

4.3 Experiment 2 results

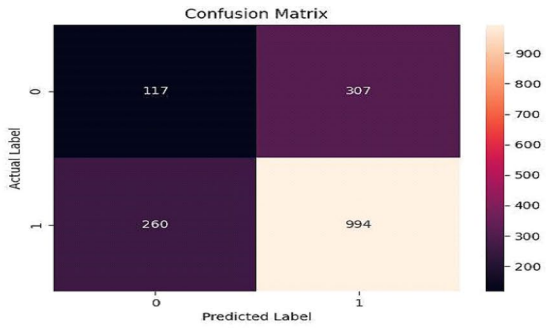
From the performance of the classifiers as shown in Table 3, it was observed that the Decision Tree obtained the highest accuracy with 92.43% compared to Naves Bayes with 75.27%, Logistic Regression obtained the second-highest accuracy after the Decision Tree with 90.46% and KNN: 85.75% making it the third-best performing classifier. It is further observed that the Decision Tree obtained the maximum recall value of 0.95, with a precision of 0.95 and 0.84 in the F-measure. The recall value means that only 5% of hateful tweets were misclassified by the system. Figure 3 shows the result for the confusion matrix for the algorithms tested with TF-IDF.

4.3.1 Discussion

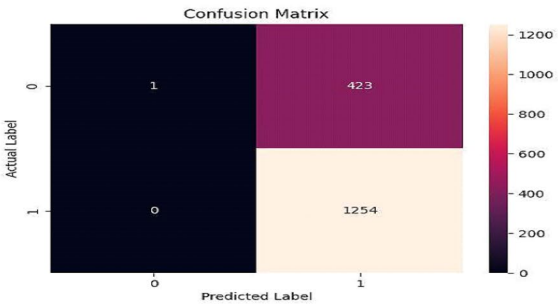
As the experimental results obtained from the BoW and TF-IDF comparison with the models show, it can be concluded that the Decision Tree achieved the highest accuracy than other algorithms. The Decision Tree classifier outperformed KNN, Naïve Bayes and Logistic Regression classifiers, and obtained higher experimental results based on the evaluation metrics used in this work. In addition, for the feature extraction methods used in the study, the Decision Tree classifier obtained better experimental results when combined with TF-IDF compared to BoW. Furthermore, the Logistic Regression classifier was the second-best classifier compared with the rest of the classifiers. When both BoW and TF-IDF were used as feature extraction techniques, the result revealed that TF-IDF outperformed BoW, allowing the Decision Tree classifier to



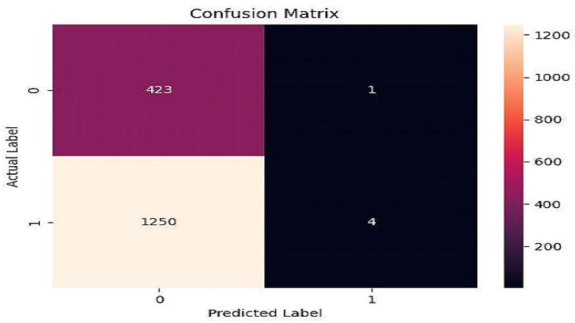
Confusion matrix chart for decision tree using bag of words



Confusion matrix chart for KNN using BoW

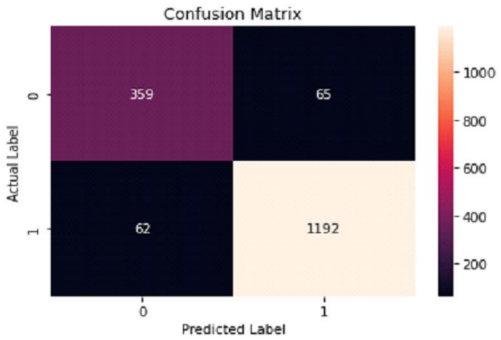


Confusion matrix chart for logistic regression using BoW

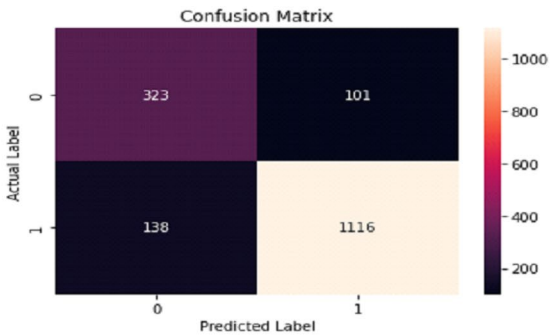


Confusion matrix chart for Naïve Bayes using BoW

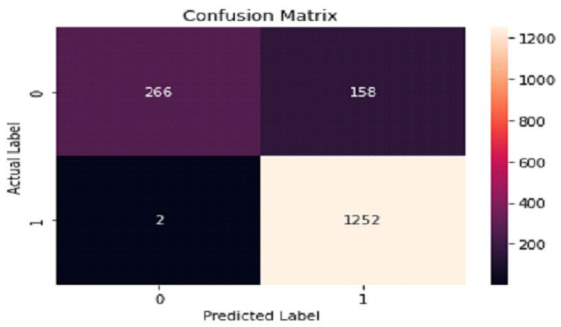
Fig. 2 Confusion matrices for Experiment 1 result



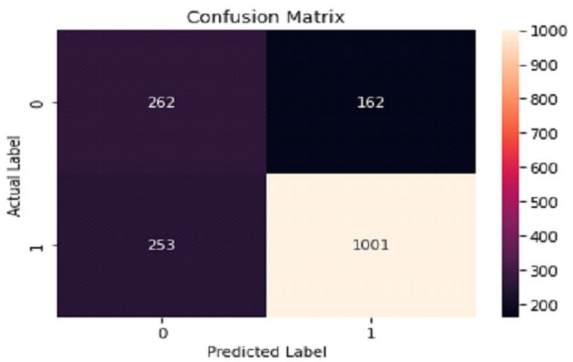
Confusion matrix chart for decision tree using TF-IDF



Confusion matrix chart for KNN using TF-IDF



Confusion matrix chart for logistic regression using TF-IDF



Confusion matrix chart for Naïve Bayes using TF-IDF

Fig. 3 Confusion matrices for Experiment 2 result

reach the maximum accuracy, while BoW performed better for the Logistic Regression classifier.

Comparing our work with similar work presents an improvement in prior studies. Some of these attempts include [11] where they used n-grams and TD-IDF for hate speech detection. The drawback of their approach is the long distance between related words which is not computationally efficient. [4]'s system for detecting cyber hate on Twitter based on limited characteristics lacks the generalization that our model provides. Limiting the categorization to race, social orientation and disability is not comprehensive enough. [8]'s approach for detecting offensive Twitter posts using Machine Learning and feature selection achieved an accuracy of 92% lower than ours. Comparability is also limited as they used only Naive Bayes and SVM for their classification which affected their result.

5 Conclusion

This research evaluated four supervised machine learning algorithms to track hateful posts or tweets on Twitter. An experimental study was carried out and the results showed that the machine learning models yielded considerably better results when tested using the TF-IDF approach than BoW with the Decision Tree yielding 92.43% when tested with TF-IDF, outperforming the other algorithms. Logistic Regression obtained the highest accuracy among the four classifiers tested with BoW with an accuracy of 74.79%. The developed model used the technique with the highest accuracy to determine the presence or absence of hateful connotations in a given tweet. An intriguing look at how hateful words are detected and how they are manifested in intolerance, religion, gender, racism, and misinformation gives the motivation for this research.

This research used tweets in text format on Twitter to detect hate speech. Future research will explore the use of emotions like emojis, optical character recognition and video images in detecting hate speech. Larger datasets with other feature extraction techniques will be used with machine learning methods for optimal results.

References

1. Abro S, Shaikh S, Hussain Z, Ali Z, Khan S, Mujtaba G (2020) Automatic hate speech detection using machine learning: a comparative study. Int J Adv Comput Sci Appl 11(8):484–491
2. Akuma S, Obilikwu P, Ahar E (2021) Sentiment analysis of social media content for music recommendation. Nigerian Ann Pure Appl Sci 4(1):95–107
3. Andrii S (2019) Kaggle. Dataset, 2022
4. Burnap P, Williams ML (2016) Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Sci 5(11):1–15. https://doi.org/10.1140/epjds/s13688-016-0072-6
5. Das AK, Asif AA, Paul A, Hossain N (2020) Bangla hate speech detection on social media using attention-based recurrent neural network. J Intell Syst 30(1):578–591
6. Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language. ICWSM
7. Fortuna P (2017) Automatic detection of hate speech in text: an overview of the topic ad dataset annotation with hierarchical classes. Thesis, Faculdade de engenharia da universidade do porto
8. De Souza GA, Da Costa-Abreu M (2020) Automatic offensive language detection from Twitter data using machine learning and feature selection of metadata. In Anonymous. In: 2020 international joint conference on neural networks (IJCNN). 2020, pp 1–6
9. Gambäck B, Sikdar UK (2017) Using convolutional neural networks to classify hate-speech. In Anonymous. In: Proceedings of the first workshop on abusive language online. (Vancouver, BC, Canada). Association for computational linguistics, pp 85–90
10. Gao L (2018) Detecting online hate speech using both supervised and weakly-supervised approaches. Master's thesis, Texas A & M University
11. Gaydhani A, Doma V, Kendre S, Bhagwat L (2018) Detecting hate speech and offensive language on twitter using machine learning: an N-gram and TFIDF based approach. Arxiv Abs/1911.02989, abs/1809.08651
12. Getachew S, Kakeba K (2020) Department of software engineering, big data and HPCCoE. Addis Ababa Science and Technology University, Addis Ababa
13. Kalra V, Kashyap I, Kaur H (2022) Improving document classification using domain-specific vocabulary: hybridization of deep learning approach with TFIDF. Int J Inf Technol 14:2451–2457
14. Kumar P, Vardhan M (2022) PWEBSA: Twitter sentiment analysis by combining Plutchik wheel of emotion and word embedding. Int J Inf Technol 14:69–77
15. Malmasi S, Zampieri M (2017) Detecting hate speech in social media. In: Advances in natural language processing (RANLP), pp 467
16. Mikolov T, Chen K, Corrado GS, Dean J (2013) Efficient estimation of word representations in vector space. ICLR
17. Pereira-Kohatsu JC, Quijano-Sánchez L, Liberatore F, Camacho-Collados M (2019) Detecting and monitoring hate speech in Twitter. Sens J 19(12):4654
18. Salminen J, Hopf M, Chowdhury SA, Jung S, Almerekhi H, Jansen BJ (2020) Developing an online hate classifier for multiple social media platforms. Hum-centric Comput Inf Sci 10(1):1–34. https://doi.org/10.1186/s13673-019-0205-6
19. Soni J, Mathur K (2022) Sentiment analysis based on aspect and context fusion using attention encoder with LSTM. Int J Inf Technol
20. Vrysis L, Vryzas N, Kotsakis R, Saridou T, Matsiola M, Veglis A, Arcila-Calderón C, Dimoulas C (2021) Web interface for analyzing hate speech. Future Internet 13(3):80. https://doi.org/10.3390/fi13030080
21. Yadav V, Verma P, Katiyar V (2022) Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. Int J Inf Technol. https://doi.org/10.1007/s41870-022-01010-y