Journal of Applied Research and Technology

www.jart.icat.unam.mx

Journal of Applied Research and Technology 22 (2024) 548-559

Original

Hate speech against women and immigrants: A comparative analysis of machine learning and text embedding techniques

A. Hussain[a]* • A. Aslam[b]

[a]*School of Electronics and Control Engineering, Chang'an University, Xi'an, China*
[b]*School of Information Engineering, Chang'an University, Xi'an, China*

Abstract: Hate speech on social media, especially against women and immigrants, is a major issue. Twitter, which promotes public discourse and diverse viewpoints, explicitly rejects violence, discrimination, and assaults based on race, nationality, ethnicity, social status, sexual orientation, age, disability, or severe illness. Hate speech harms individuals and communities, but the volume of internet content makes routine detection impractical. This challenge highlights the need to address and develop effective hate speech detection and categorization systems for women and immigrants. This research describes the deployment of two advanced machine learning paradigms, the Random Forest, and Support Vector Machine (SVM), using text pre-processing, post-processing, and advanced text embedding techniques like TF-IDF, CBOW, and GloVE. Detailed categorization of a Twitter dataset into hate speech and subclassification into aggressive and targeted dimensions is the main goal. Model efficacy is carefully assessed based on the complex interaction of text embeddings and classification typology. The Random Forest classifier excels at hate speech categorization when combined with TF-IDF embeddings. Concurrently, merging GloVE embeddings with the SVM algorithm accurately discriminates between aggressive, non-aggressive, targeted, and non-targeted categories. Also, CBOW embeddings work well for broader hate speech classification. Thus, this work improves social media hate speech identification by providing theoretical and practical insights.

Keywords: Sentiment analysis, machine learning, Twitter sentiment, text embedding, speech analysis

*Corresponding author.
E-mail address: 2022032907@chd.edu.cn (A. Hussain).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

In recent years, automatic hate speech detection has become a research problem (Fortuna & Nunes, 2018; Salawu et al., 2017; Schmidt & Wiegand 2017). The rise of social media platforms worldwide has led to people interacting with each other more through text-based messages. These messages can significantly impact people's thoughts and beliefs, and some social media platforms have enough power to influence how democratic processes work (Fandos & Roose, 2018) . As more people use digital platforms to communicate, there is a growing concern about hate speech and online harassment. It's essential to accurately identify and evaluate these issues to create safe and equal access to these platforms for everyone (Delisle et al., 2019). Scholarly literature and political debates persist in centering around the concept of freedom of speech. Hate speech acceptance is witnessed to varied extents in numerous countries (Howard, 2019). Various communities and individuals have been deeply harmed by hate speech and hate offenses throughout history.

Nevertheless, it is critical to acknowledge that hate speech legislation is founded upon the fundamental tenet of substantive equality. Social media platforms demonstrate a significant level of concern regarding the existence of user-generated content that is considered inappropriate (Nasser Alsager, 2021). The lack of accountability and oversight mechanisms on social media platforms, including Twitter, has contributed to the spreading of hate speech (Erdem, 2021). Although social media companies employ personnel to assist in content moderation, the overwhelming volume of social media posts renders human agent's incapable of effectively monitoring all relevant users.

With increasing digital communications, many public debates are moving to the internet, spanning broadcasts, text, video, and emoticons. These debates manifest the vast array of human experiences, including illuminating and educational dialogues, humorous and entertaining exchanges, and those about political or religious subjects. Additionally, some individuals may demonstrate spiteful and unsightly conduct. Presently, many sophisticated communication platforms and systems are purposefully endeavoring to mitigate the proliferation of harmful content on the internet. The advent of Twitter and community forums has brought about a significant transformation in communication and content generation. Nevertheless, an emerging phenomenon has emerged wherein social media platforms are being employed to facilitate the distribution of hate speech and orchestrate hate-motivated endeavors. At present, there is a notable surge in the prevalence of xenophobia, which is leading to increased sentiments of social discontent and animosity towards communities. The current upsurge in xenophobic sentiment may be linked to the persistent refugee crisis and recent political transformations that have taken place in recent years. Many governmental entities and policymakers are actively involved in addressing the matter, specifically developing tools designed to detect and monitor hate speech.

Machine learning has been implemented in several fields recently, like intrusion detection (Hussain et al., 2024), fraud detection (Aslam & Hussain, 2024), and disease prediction (Hussain & Aslam, 2024). The main objective of this research is to perform binary classification of hateful tweets targeted towards women and immigrants, using machine learning models with several text embedding techniques. The study specifically addresses aggressive comments directed towards women and targeted comments at immigrants. The research proposes an automated hate speech classification system utilizing natural language processing and machine learning techniques. Machine learning is a well-established field that enables software or machines to improve task performance through exposure to data and experiences. This research employs several pre-processing and text embedding techniques, such as TF-IDF, CBOW, and GloVE embeddings. Additionally, machine learning algorithms, including Random Forest and SVM, are trained and evaluated for three binary classifications: Hate speech or non-hate speech (HS vs non-HS), targeted or non-targeted (TG vs. non-TG) if it is hating speech, and aggressive or non-aggressive (AG vs non-AG). The proposed system offers promising results in identifying hateful tweets with high accuracy and can be utilized in social media monitoring and management. The main contributions of this research work are as follows:

- Implementing machine learning models, such as Random Forest and Support Vector Machine, to classify hate speech against women and immigrants. Classify hateful information as aggressive, non-aggressive, targeted, or non-targeted.
- The machine learning models use text embedding techniques, including TF-IDF, Bag of Words, and GloVE embedding. This allows us to experiment and determine each use case's most effective embedding technique.
- The comparison of machine learning models' performance in natural language processing tasks using three text embedding approaches from a technical perspective.

This paper is organized as follows: Section II covers related works. Section III includes text pre-processing, embedding, and binary classification models. A detailed dataset description and research evaluation measures are also supplied. Section V describes implementation, Section VI discusses outcomes and debate, and Section VII concludes.

2. Related work

This section highlights the previous work related to hate speech analysis and classification. Davidson et al. (2017) used

logistic regression, naive Bayes, decision trees, Random Forests, and linear SVMs for multi-class classification. The researchers compiled tweets containing hate speech keywords by utilizing a crowdsourced lexicon. The tweets were subsequently classified into three groups: those that were found to contain hate speech, those that contained objectionable language but did not express hatred, and those that were devoid of both hate speech and offensive language. In addition, the investigators employed the Porter stemmer method to generate bigram, unigram, and trigram characteristics. The TF-IDF value was utilized to assign weight to each feature. In pursuit of this objective, the NLTK library was implemented. The improved Flesch-Kincaid Grade Level and Flesch Reading Ease scores were used as metrics to assess the content of the tweets to create part-of-speech (POS) tags. Park and Fung (2017) proposed a categorization algorithm based on the Wassem datasets. This methodology necessitated the development of a Convolutional Neural Network (CNN) that operated on Word2Vec word embeddings. The classification results for the subcategories of racism and misogyny hate speech demonstrated in the study validated the effectiveness of this method.

Kamble and Joshi (2018) investigated CNN-1D, LSTM, and Bidirectional LSTM models. Each term was assigned a 300-dimensional Vector using domain-specific embeddings. CNN-1D demonstrated superior performance to the alternative models, as evidenced by its F1-score of 0.8085. Malmasi and Zampieri (2017) employed supervised classification and lexical baselining techniques like character n-grams, word n-grams, and word skip grams. The task of differentiating hate speech from undesirable language was accomplished with a commendable accuracy rate of 78%, employing three unique labels. Wei et al. (2021) proposed a systematic approach to detect inappropriate language within a dataset of tweets accessible to the public. Their approach involves employing a Bi-LSTM model that integrates pre-trained GloVe embeddings and null embeddings. Furthermore, a comparative study uses pre-trained language models such as BERT, DistilBERT, and GPT-2. During the pre-processing stage, hashtags and emoticons in the raw data are managed efficiently. Upon evaluation using the test data, the fine-tuned Bi-LSTM model exhibits an accuracy of 92%, thereby outperforming the transfer learning models. This precision is attained through the application of optimal hyperparameters. Das et al. (2021) addressed the binary categorization problem. The evaluated model defines word vectors using TF-IDF and BERT embeddings, given their superior performance compared to word2vec, GloVe, and other similar alternatives. SVM was employed to classify English and Spanish datasets utilizing TF-IDF embeddings. In addition, the English dataset was analyzed using CNN and a pre-trained model known as BERTweet. The SVM model exhibits a respective accuracy rate of 67% for the English and 81% for the Spanish datasets. Furthermore, CNN achieved a rate of accuracy amounting to 66%.

Gupta et al. (2021) performed a multi-class classification task. They implemented character-level embeddings to mitigate the difficulties posed by grammatical liberties and transliteration variations in mixed-language code. After conducting experiments with twelve distinct models, the authors identified three that exhibited both efficiency and robustness. The study compares three NLP models: GRU, GRU w/attention, and bidirectional LSTM followed by GRU w/attention. Pariyani et al. (2021) implemented machine learning algorithms to categorize material into hate speech or non-hate speech categories. The algorithms were executed using a dataset acquired from the social media platform Twitter. The hate speech dataset has been analyzed using supervised classification methods, specifically logistic regression, Support Vector Machines, and Random Forest. The researchers used the TF-IDF and bag of words methodologies to extract text features. When combined with a bag of words approach, the Random Forest algorithm produces the most optimal outcomes without requiring data preparation. For optimal results, SVM is utilized in conjunction with TF-IDF after preprocessing. On the contrary, TF-IDF is preferred over the bag-of-words method since the latter solely considers the frequency of words, which is subsequently employed in generating vectors. Das et al. (2021) performed hate speech classification using a dataset collected from Twitter. The tweets have been classified into three discrete categories: objectionable language, hate speech, and a category that does not fit into the classifications. A method proposed in the current study uses deep neural networks and word embedding representation. The present study investigated two embedding representations, specifically fastText and BERT. Many methodologies have been analyzed and discussed in the field of classification. A comparative analysis was conducted on the DNN-based classifier, during which the efficacy of the CNN, Bi-LSTM, and CRNN architectures was assessed. Based on the results, it was concluded that the performance of BERT fine-tuning was superior.

The above literature review shows that pervasive works on the same topic have been done worldwide on different datasets, using different techniques. Only a few research studies focus on one aspect of hate speech detection. Our work proposes and narrows down hate speech addressed toward women and immigrants as aggressive or targeted speech. Furthermore, these previous works just perform a broad, open-ended categorization of hate and offensive speech. Our work also addresses whether hate speech targets a particular individual and whether it is aggressive or violent. Thus, the novelty of our work improves upon the existing work in multi-folds.

3. Methodology

The methodology involves three processes: Text preprocessing, Text embedding using TF-IDF, bag of words, and GloVE, and binary classification using Random Forest and Support Vector Machine.

3.1. Text pre-processing

Text preprocessing is a critical component that significantly influences the outcome of any natural language processing endeavor. The model of machine learning is incapable of comprehending what humans inherently comprehend. The data must, therefore, be simplified and reduced in complexity. The data must be cleansed before being inputted into the model. Since we are utilizing the Twitter dataset, the scraped raw data must be cleansed, as it contains many superfluous or extraneous words that could potentially hinder the model's performance. The preprocessing of text for machine learning models consists of several crucial operations. One of these procedures entails the conversion of every word in the text to lowercase to promote uniformity and facilitate the extraction of precise features. Furthermore, stop words and punctuation marks are eliminated to standardize the text and highlight pertinent terms. Particularly crucial in social media texts is removing usernames to eradicate superfluous information. In addition, additional preprocessing procedures, including the exclusion of digits and emoticons, lemmatization, and the removal of URLs and HTML elements, help to acquire more precise and meaningful data for subsequent analysis. Figure 1 shows the text pre-processing.

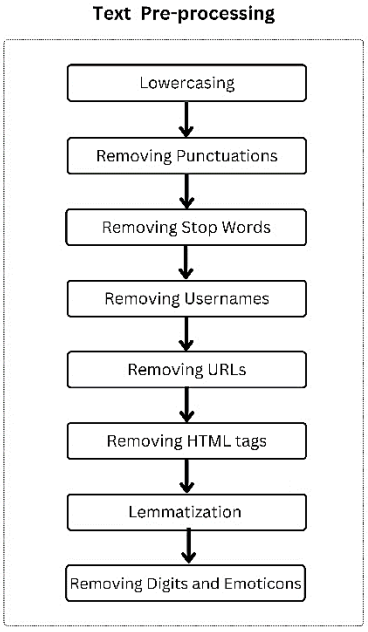

Figure 1. Text pre-processing steps.

After the text pre-processing, Tokenization is performed as post-pre-processing. Tokenization breaks down language into individual words or phrases. Recent tech advances improve accuracy for complex languages

3.2. Text embeddings

Word embedding is the numerical representation of words. We want to quantify semantics. We want to express words in such a way that they catch their meaning in the same way that humans do—not the literal meaning but a contextual one. This is why word embeddings are required, and the word embeddings/vectorization approaches used in this study are discussed below.

- **TF-IDF:** TF-IDF is useful in various applications, including search engine optimization, document clustering, and text classification. This method successfully manages the effects of often recurring terms that may have less importance in analysis. As a result, the model can prioritize different terms and have a higher descriptive value for the specific document. TF-IDF is employed in the models in this study as a vectorization or text embedding procedure. Words are converted into numerical values using the TF-IDF score to enable machine learning algorithms to understand text. This score reflects the importance of each word within a document. TF-IDF is helpful to compare the relevance of words across different documents. In our study, we use this technique to identify hate speech by analyzing content containing similar phrases. This allows us to use machine learning to accurately categorize and analyze hate speech data.
- **CBOW:** The CBOW model utilizes the contextual terms surrounding the center word to predict the present target word. CBOW and other Word2Vec models are classified as unsupervised learning algorithms. This indicates they can generate compact word embeddings from a given corpus without additional labels or data. However, once the corpus has been gathered, a supervised classification technique is necessary to use these embeddings successfully. It is worth emphasizing that this can be done without relying on additional information within the corpus. Compared to the skip-gram model, which seeks to predict several context words for a given source-target combination, building this architecture is easy. Compared to the skip-gram model, the CBOW model has a faster training speed and better accuracy for frequently occurring words. It also outperforms other algorithms when applied to smaller datasets. As a result, instead of using the skip-gram model for word embedding, the CBOW technique is used to estimate the recurrence rate of specific hate speech phrases in each corpus.
- **GloVe embedding:** GloVe is an abbreviation for "Global Vectors." The GloVe method is a form of unsupervised learning that generates word vector representations. The resulting representations highlight significant linear

substructures inside the word vector space, and the training procedure entails utilizing aggregated global word-word co-occurrence information acquired from the corpus. GloVe differentiates itself from Word2vec by including global statistics and local statistics in the production of word vectors. Unlike Word2vec, which relies solely on word context information, GloVe considers word co-occurrence as a crucial factor in its vectorization process. The GloVe technique runs on the idea that the co-occurrence matrix can be used to determine semantic associations between words, which is critically evaluated. The GloVe word vector technique incorporates local and global statistics from a corpus into a well-founded loss function. We employ an existing embedding model for hate speech identification in our model.

3.3. Machine learning models

The classification techniques used in this research work are given below.

- **Random Forest (RF):** Random Forest is a collection of decision trees known as random because they are unrelated trees that work together to form a single model. A Random Forest's core principle is that each tree presents its prediction, and the Random Forest predicts its outcome based on the majority decision. This work utilizes tree-based classifiers to determine whether a remark can be categorized as hate speech. A Random Forest model is employed to predict whether a message is hating speech by considering the majority vote. The same idea applies to the remaining two sub-classifications. The RF algorithm would process vectorized text embeddings from TF-IDF, CBOW, and GloVE to perform the classifications.
- **Support Vector Machine (SVM):** SVM is a supervised machine learning method for regression and classification. Each data point is represented as a point in an n-dimensional space, where n corresponds to the number of characteristics. The technique identifies the optimal hyperplane that effectively partitions the data points into distinct classes. It is particularly efficient for data with many dimensions and provides accurate results even when the number of features exceeds the number of samples. The categorization is then completed by determining the hyper-plane that separates the two classes. SVM uses nonlinear or linear mapping to convert lower-dimensional input into higher-dimensional data. It seeks the linear optimal dividing hyperplane within this new dimension to split the tuples between the sets. When scaling nonlinearly to an appropriate high dimension, the discrepancies between two array scans are inevitably separated by a hyperplane. The SVM determines the hyperplane by utilizing support vectors. Support vectors are specific instances of vectors that closely approach the boundaries. An infinite number of dividing lines could be traced at this location. The objective is to classify the "highest" ones with the least amount of error using previously unknown tuples.

Tweet categorization is performed according to hate speech (HS), targeted (TR), and aggressive (AG):

1. **HS –** HS value is either 1 or 0, depending on whether an event of HS against one of the targets in the list has occurred.
2. **TR** - If HS occurs (meaning that a certain feature has a value of 1), then the binary value is used to classify whether the target is a generic group of persons (0) or a single individual (1).
3. **AG** - If HS happens (which means that the HS characteristic has a value of 1), then the binary value is used to classify the tweet as aggressive (1) or not (0).

Figure 2 shows the methodology of this research work.

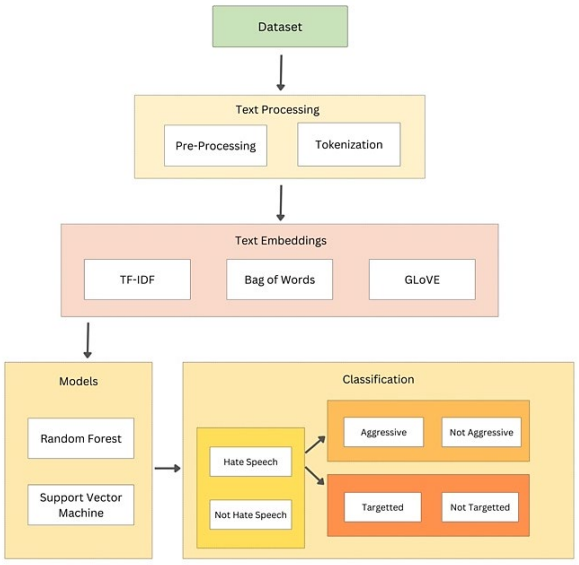

Figure 2. Methodology.

3.4. Dataset

The dataset on automatic hate speech identification in social media was obtained from the data repository maintained by the University of Turin (UniTO) and is presented in CSV File format. It has 12,200 tweets and has been partitioned into separate train and test sets. The train set consists of 9,200 tweets, while the test set consists of 3,000 tweets.

3.5. Evaluation metrics

The classification model's performance is being evaluated using the following assessment metrics.

- **Accuracy:** When compared to the total number of cases evaluated, accuracy is the percentage of correctly identified cases. It is a popular metric for evaluating performance in machine learning and related domains that deal with classification tasks; it indicates how accurate a model is at making predictions.

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN) \quad (1)$$

- **Precision:** The "precision" metric measures how many of the projected positive cases were identified as such. Put another way, it is a way to determine how well a model or algorithm makes accurate predictions.

$$\text{Precision} = (TP)/(TP+FP) \quad (2)$$

- **Recall:** In binary classification, recall measures a model's capacity to accurately detect all positive cases out of the total actual positives. It is the ratio of the number of correct results to the total number of correct results plus the number of false negatives.

$$\text{Recall} = (TP)/(TP+FN) \quad (3)$$

- **F1-Score:** One takes the harmonic mean of the model's recall and precision to find the F1 score, which measures a classification model's performance.

$$F1 = 2 * ((\text{Precision} * \text{Recall})/(\text{Precison}+\text{Recall})) \quad (4)$$

- **Specificity:** The specificity score evaluates a model's ability to forecast the true negatives for each available category reliably. This statistic shines when assessing the performance of machine learning models for specific technical tasks.

$$\text{Specificity} = (TN)/(TN+FP) \quad (5)$$

4. Implementation

The dataset is separated into training and testing, with an 80/20 ratio. Subsequently, test preparation is conducted on the data utilizing all the techniques elucidated in the preceding section. Word tokenization is conducted during post-preprocessing using a pre-existing NLTK tokenizer. In addition, before utilizing the tokenized words to generate word embeddings. Figure 3 shows the word clouds.

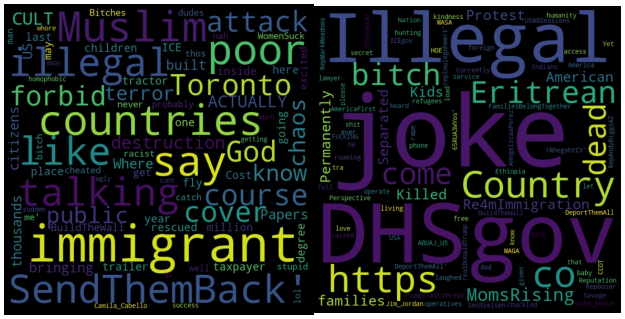

Figure 3. Word clouds.

After the text processing, text embeddings, including TF-IDF, CBOW, and GloVE, are perfumed. The initial dataset used in this study is the pre-processed Twitter dataset. The TF-IDF technique is implemented using the Scikit Learn library. The initial stage entails employing the TF-IDF vectorizer module to execute the vectorization process on the dataset. The input data is pre-processed and subjected to a fitting and transformation procedure. This method results in generating a matrix containing the TF-IDF values for each word present in the document. The matrix then serves as the input for classifiers, SVM, and Random Forest. The parameter updates of the word2vec model are derived from the research conducted by Rong (2014). The final weight vectors are obtained by utilizing backpropagation, employing a learning rate 0.0001, and implementing a window size of 4. This window size encompasses two words preceding and one word succeeding the center word. The word embeddings, characterized by a dimensionality of 4, are subsequently utilized to train the Random Forest and Support Vector Machines (SVM) machine learning models.

Each word is associated with a vector in a 100-dimensional space. While GloVe vectors may not be directly compatible with classification models, they can be transformed into the word2vec format using the genism package in Python. The procedure assigns a distinct vector to each word included in the corpus. Sentence embeddings must be generated to accommodate the sentences in the training data. The process involves calculating the average word embeddings for each word in a sentence, resulting in a vector representing the phrase in the dataset. Sentence embeddings are generated for each sentence, along with their respective labels. The dataset is partitioned into training and testing sets using an 80-20 distribution. The SVM framework commonly employs the Radial Basis Function (RBF) kernel. The hyperparameters C and gamma are optimized using GridSearchCV. The range of gamma values includes 0.0001, 0.001, 0.01, 0.1, 1, and 10, while C values include 0.01, 0.1, 1, 10, and 100. Different SVM models are evaluated on the training data using various combinations of C and gamma. The highest accuracy combination is then employed to classify the test data.

There are more variable hyperparameters in the Random Forest. Therefore, the RandomisedSearchCV method was employed instead of the GridSearchCV method. The range of estimators was adjusted from 10 to 100, the max split parameter was set to either log2 or sqrt, the maximum levels of the tree were altered from 10 to 110, and the minimum number of splits at each node was set to 2, 5, or 10. The minimal number of leaves at each node was selected from a range of values, specifically [1, 2, 4]. The combination with the highest performance on the training data was selected and applied to the test data.

5. Results and discussion

This section shows the results of the machine learning models using TF-IDF, CBOW, and GloVe text embeddings. Both models are evaluated using text embeddings, TF-IDF, CBOW, and GloVe, to determine their accuracy, precision, recall, f1 score, and specificity.

5.1. Models using TF-IDF embeddings

The Random Forest algorithm employing TF-IDF text embedding for sentiment analysis produces performance measures for HS vs. non-HS, AG vs. on-AG, and TR vs. non-TR. In HS vs. non-HS classification, the approach has 73.29% accuracy and 81.82% precision. The system's F1 score is 78.04% due to its 75.73% recall rate. Additionally, the method has 71.02% specificity. For AG vs. non-AG, accuracy is 67.19%. The precision and recall are 76.64% and 71.45%, respectively, and the F1 score of 73.24%. A specificity of 61.98% suggests the categorization model is less accurate at categorizing non-aggressive content. The model does well at TR vs. non-TR classification, with 83.64% accuracy, 85.15% precision, 88.36% recall, and 86.72% F1-score. Specificity is also higher at 76.44%, suggesting a better distinction of accurate negative outcomes in this category. Figure 4 shows Random Forest performance with TF-IDF.

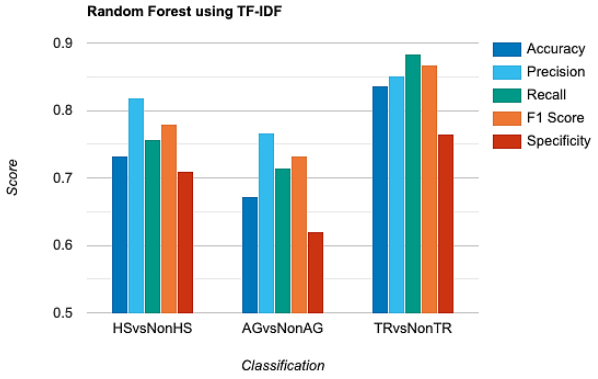

Figure 4. Random Forest performance using TF-IDF.

SVM sentiment analysis performance depends on classification when employing TF-IDF text embedding for feature extraction. The SVM method classifies HS and non-HS with 72.62% accuracy and 79.02% precision. The recall rate is 75.08%, with an F1-score of 77.00% and a specificity of 68.77%. AG vs. non-AG classification accuracy improves to 67.88%. The precision and recall metrics are 76.26% and 69.92%, respectively, giving an F1-score of 72.95%. This category has 64.58% specificity, indicating that the model can moderately recognize non-aggressive content. SVM excels at separating TR from non-TR sentiment. It has 85.33% accuracy. Precision is 87.83%, recall is 88.67%, and F1-score is 88.25%. The SVM has 79.84% specificity. When using CBOW text embedding, the SVM is quite good at distinguishing targeted and non-targeted content. Figure 5 shows the TF-IDF evaluation of SVM.

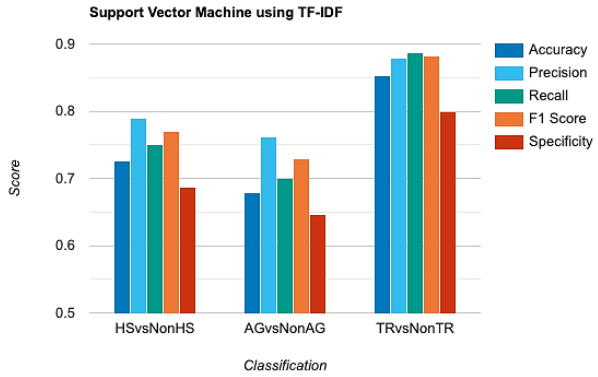

Figure 5. SVM Performance using TF-IDF.

The Random Forest and SVM algorithms, both employing TF-IDF text embeddings, exhibit discernible variations in their performance.

- **HS and non-HS:** The Random Forest algorithm exhibits a minor advantage over the SVM algorithm in accuracy (73.29% vs 72.62%), precision, recall, F1-score, and specificity.
- **AG and non-AG:** The Random Forest algorithm exhibits slightly higher accuracy (67.19% vs. 67.88%) and F1-score. On the other hand, the SVM algorithm demonstrates a slightly better recall.
- **TR vs non-TR:** SVM outperforms Random Forest in all metrics, notably achieving greater accuracy (85.33% vs 83.64%) and F1-score (88.25% vs 86.72%).

The performance comparison of both models using TF-IDF text embedding is shown in Table 1.

5.2. Models using CBOW embeddings

Random Forest using CBOW text embedding distinguishes HS from non-HS with 72.95% accuracy. The F1-score is balanced at 76.2% due to its strong precision and recall rates of 76.4% and 74.4%, respectively. The specificity is moderately high at 68.7%. Classification accuracy drops to 68.38% when separating violent and non-aggressive behavior. However, the F1-score of 86.9% suggests a good balance between precision and recall, with a specificity of 76.1%. Compared to non-TR, the model performs better. Its 84.14% accuracy, 84.3% precision, 86.8% recall, and 86.9% F1 score are impressive. This context maintains 76.1% specificity. This implies that the Random Forest approach using CBOW embeddings effectively detects text attitudes. Figure 6 shows the Random Forest results with CBOW embedding.

Table 1. Models' performance using TF-IDF.

Models	Classification	Accuracy	Precision	Recall	F1 score	Specificity
Random Forest	HS vs non-HS	0.7329	0.8182	0.7573	0.7804	0.7102
	AG vs non-AG	0.6719	0.7664	0.7145	0.7324	0.6198
	TR vs non-TR	0.8364	0.8515	0.8836	0.8672	0.7644
Support Vector Machine	HS vs non-HS	0.7262	0.7902	0.7508	0.7700	0.6877
	AG vs non-AG	0.6788	0.7626	0.6992	0.7295	0.6458
	TR vs non-TR	0.8533	0.8783	0.8867	0.8825	0.7984

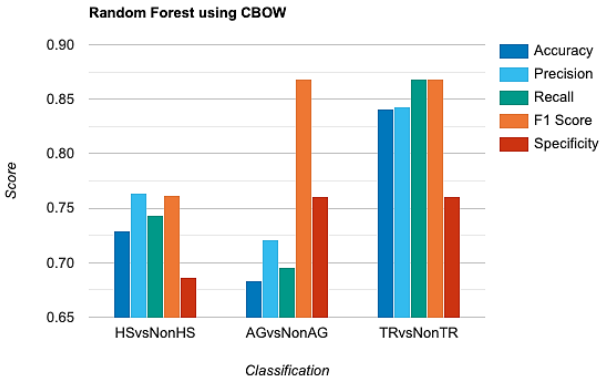

Figure 6. Random Forest performance using CBOW.

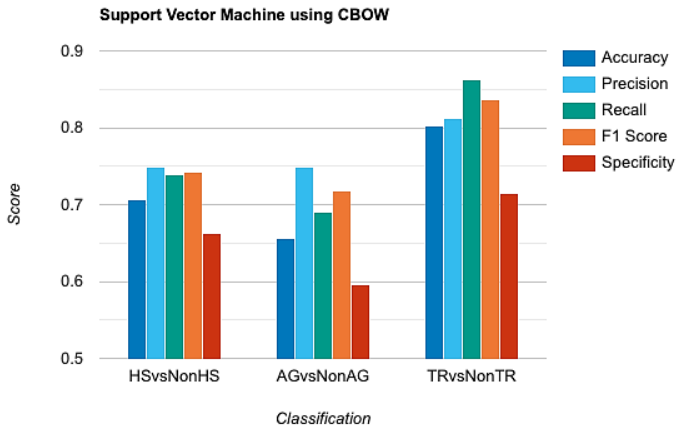

Figure 7. SVM performance using CBOW.

The SVM utilizing CBOW text embeddings achieves 70.70% accuracy for HS vs. non-HS, 74.8% precision, 73.9% recall, 74.3% F1-score, and 66.2% specificity. AG or non-AG categorization accuracy is 65.60%. Precision is 74.9%, recall 69.0%. F1-score is 71.8%, and specificity is 59.6%. TR or non-TR categorization improves to 80.27% accuracy. The precision is 81.3%, reflecting the percentage of TR events accurately categorized. The recall is 86.3%, representing the percentage of TR events successfully classified. The precision-recall F1-score is 83.7%. The specificity, or percentage of accurately diagnosed non-TR cases, is 71.5%. The SVM model uses the CBOW algorithm to distinguish TR from non-TR well. Figure 7 shows CBOW-trained SVM model results.

The comparison of the Random Forest and SVM algorithms using CBOW text embedding for the classification is provided below.

- HS and non-HS: the Random Forest model has a marginally superior performance in terms of accuracy (72.95% vs. 70.70%), precision (76.4% vs. 74.8%), and recall (74.4% vs. 73.9%). The Random Forest model has better values for the F1-score and specificity metrics, suggesting a more equitable capacity to classify instances within this category.
- AG vs. non-AG: the Random Forest algorithm demonstrates superior accuracy (68.38% vs. 65.60%) and precision (72.1% vs. 74.9%). Nevertheless, the F1-score of the Random Forest model exhibits a substantial increase (86.9% compared to 71.8%), indicating a more favorable equilibrium between precision and recall. The Random Forest algorithm performs better than the SVM algorithm in terms of specificity, with 76.1% and 59.6%, respectively. This suggests that Random Forest is more effective in accurately identifying non-aggressive cases.
- TR versus non-TR: Random Forest performs better, exhibiting better levels of accuracy (84.14% vs. 80.27%), precision (84.3% vs. 81.3%), and recall (86.8% vs. 86.3%). The F1 scores exhibit comparability, with the Random Forest model marginally outperforming the SVM model. The level of specificity is equivalent for both models within this classification.

The performance of both models using CBOW text embedding is shown in Table 2.

Table 2. Models' performance using CBOW.

Models	Classification	Accuracy	Precision	Recall	F1 score	Specificity
Random Forest	HS vs non-HS	0.7295	0.764	0.744	0.762	0.687
	AG vs non-AG	0.6838	0.721	0.696	0.869	0.761
	TR vs non-TR	0.8414	0.843	0.868	0.869	0.761
Support Vector Machine	HS vs non-HS	0.7295	0.764	0.744	0.762	0.687
	AG vs non-AG	0.6838	0.721	0.696	0.869	0.761
	TR vs non-TR	0.8414	0.843	0.868	0.869	0.761

5.3. Models using GloVe embeddings

GloVe text embeddings affect the Random Forest algorithm's sentiment analysis performance per category. The HS vs. non-HS categorization task has 66.66% accuracy. Precision, or the percentage correctly categorized as hate speech, is 64.52%. The recall, which measures the percentage of hate speech incidents, is 45.83%. Thus, the F1-score, which combines precision and recall, is 53.59%. However, at 81.75%, the specificity—the percentage of non-hate speech cases properly classified—is high. At 64.12%, AG vs. non-AG accuracy is lower. Precision is 59.20%, recall 45.73%. The F1-score is 51.60% and specificity is 77.34%. The TR classification outperforms the non-TR classification with 82.16% accuracy, 75.93% precision, 73.40% recall, 74.647% F1-score, and 87.03% specificity. This shows that the model can accurately classify genuine negatives, especially TR and non-TR. Figure 8 shows glove-based Random Forest performance.

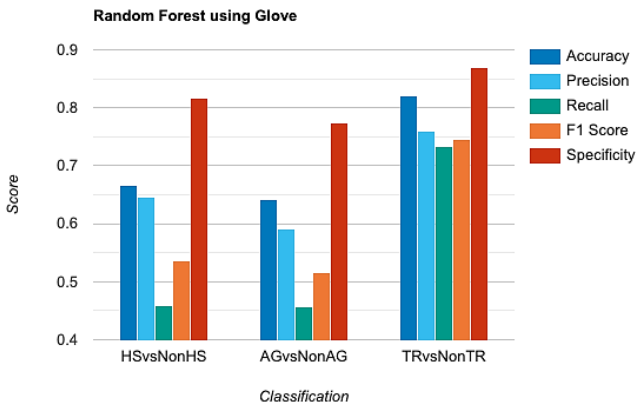

Figure 8. Random Forest performance using GloVe.

SVM using GloVe text embedding for the HS vs. non-HS achieves an accuracy of 70.20%, precision of 66.37%, and recall of 60.03%. Thus, its F1-score is 63.049% and its specificity is 77.67%. The method classifies AG vs. non-AG with 81.87% accuracy. It has 84.61% precision and 82.47% recall. Additionally, its F1-score of 84.81% and 99.89% specificity are exceptional. The SVM model performs well in sentiment analysis, particularly TR vs. non-TR. SVM model accuracy is 84.79%, precision 86.97%, recall 85.19%, F1-score 84.93%, and high specificity 96.32%. This study shows the SVM's precision and specificity in identifying AG, non-AG, TR, and non-TR categories. Figure 9 shows the Average performance in classifying HS versus non-HS categories. Glove SVM performance.

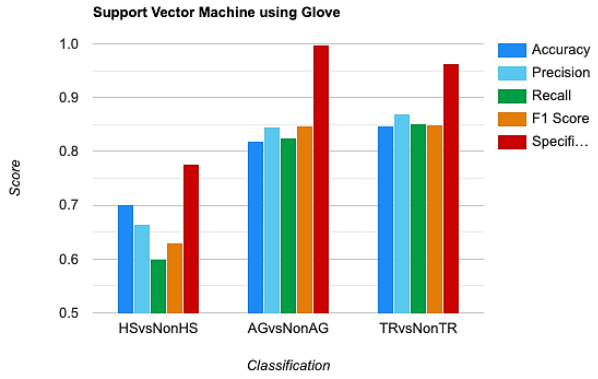

Figure 9. SVM performance using CBOW.

The comparison of the Random Forest and SVM algorithms using glove text embedding for the classification is provided below.

- HS vs non-HS: SVM shows better performance in terms of accuracy, precision, recall, and F1-score when compared to Random Forest. Specifically, SVM achieves an accuracy rate of 70.20%, surpassing the 66.66% accuracy rate achieved by Random Forest. Nevertheless, it is worth noting that the Random Forest algorithm exhibits a higher specificity

rate of 81.75% compared to the alternative method, which achieves a specificity rate of 77.67%.

- AG vs. non-AG: SVM outperforms Random Forest across all parameters. SVM demonstrates a much greater accuracy (81.87% compared to 64.12%), superior precision, recall, F1-score, and perfect specificity (99.89% compared to 77.34%).
- TR vs. non-TR: SVM demonstrates superior performance over Random Forest, exhibiting greater accuracy (84.79% vs. 82.16%), precision, recall, and F1-score. The Random Forest algorithm has an elevated level of specificity, with a recorded value of 87.03%. However, the SVM surpasses this level of specificity, achieving a higher value of 96.32%.

The performance of both models using GloVE text embedding is shown in Table 3.

6. Conclusions

The contemporary digital landscape is witnessing an unprecedented escalation in the prevalence of hate speech across social media platforms. This phenomenon necessitates an urgent and strategic intervention to address the burgeoning bias and the rampant abuse of anonymity, which often serves as a shield for disseminating malevolent and injurious rhetoric against vulnerable individuals. In response to this challenge, the current research endeavor operationalizes two sophisticated machine learning architectures: the Random Forest and Support Vector Machine (SVM) algorithms. These methodologies are meticulously integrated with a suite of text pre-processing techniques and advanced text embeddings, including Term Frequency-Inverse Document Frequency (TF-IDF), Continuous Bag of Words (CBOW), and Global Vectors for Word Representation (GloVE). The primary objective is to execute a nuanced classification of Twitter data, delineating hate speech and further bifurcating it into aggressive and targeted subcategories. The process involves an initial text pre - processing phase, as well as refining and preparing Twitter data for analysis. Subsequently, text-embedding methodologies are employed to transmit word data into a numerical format, enhancing the models' capacity for training, classification, and prediction. This study undertakes three distinct classifications: A primary differentiation between hate speech (HS) and non-hate speech (non-HS), followed by two sub-classifications of hate speech into targeted (TR) vs. non-targeted (non-TR) and aggressive (AG) vs. non-aggressive (non-AG) categories. Upon an exhaustive evaluation of the results, utilizing a variety of metrics, it emerges that the TF-IDF embedding, and vectorization technique is optimally suited for the primary HS vs. non-HS classification. Conversely, the sub-classifications focusing on aggressiveness and target specificity yield the most accurate results when employing GloVE embedding in conjunction with the SVM classifier.

In a comparative analysis of models utilizing diverse text embeddings, it is discerned that the amalgamation of the Random Forest classifier with TF-IDF embedding emerges as the most productive model for classifying hate speech. The GloVE embeddings, especially when paired with the SVM classifier, demonstrate unparalleled proficiency in the AG vs. non-AG and TR vs. non-TR classification tasks, registering the highest accuracy metrics. Moreover, integrating CBOW embeddings with the Random Forest classifier exhibits commendable performance across various sentiment analysis endeavors. This highlights the distinctive advantages of each embedding strategy under specific contextual applications. The study thus illuminates the unique merits of each embedding technique within situational paradigms. Looking ahead, exploring additional datasets, employing a spectrum of text embeddings and augmented machine learning methodologies, could further enrich the comparative analysis and enhance the efficacy of hate speech detection and classification.

Table 3. Models' performance using GloVe.

Models	Classification	Accuracy	Precision	Recall	F1 score	Specificity
Random Forest	HS vs non-HS	0.6666	0.6452	0.4583	0.5359	0.8175
	AG vs non-AG	0.6412	0.5920	0.4573	0.5160	0.7734
	TR vs non-TR	0.8216	0.7593	0.7340	0.74647	0.8703
Support Vector Machine	HS vs non-HS	0.7020	0.6637	0.6003	0.63049	0.7767
	AG vs non-AG	0.8187	0.8461	0.8247	0.8481	0.9989
	TR vs non-TR	0.8479	0.8697	0.8519	0.8493	0.9632

Conflict of interest

The authors have no conflict of interest to declare.

Funding

The authors received no specific funding for this work.

References

Aslam, A., & Hussain, A. (2024). A Performance Analysis of Machine Learning Techniques for Credit Card Fraud Detection. Journal of Artificial Intelligence (2579-0021), 6.

Das, K. G., Garai, B., Das, S., & Patra, B. G. (2021). Profiling Hate Speech Spreaders on Twitter. In CLEF (Working Notes) (pp. 1892-1898).

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media* (Vol. 11, No. 1, pp. 512-515).
https://doi.org/10.1609/icwsm.v11i1.14955

Delisle, L., Kalaitzis, A., Majewski, K., de Berker, A., Marin, M., & Cornebise, J. (2019). A large-scale crowdsourced analysis of abuse against women journalists and politicians on Twitter. *arXiv preprint arXiv:1902.03093*.
https://doi.org/10.48550/arXiv.1902.03093

Erdem, B. (2021). Fighting Infodemic Becomes Must After Covid-19 Pandemic's Onslaught on Truth, Knowledge. *European Journal of Natural Sciences and Medicine*, *5*(2), 111-124.
https://doi.org/10.26417/778kzy96j

Fandos, N., & Roose, K. (2018). Facebook identifies an active political influence campaign using fake accounts. *The New York Times*, *7*.
https://www.nytimes.com/2018/07/31/us/politics/facebook-political-campaign-midterms.html

Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, *51*(4), 1-30.
https://doi.org/10.1145/3232676

Gupta, V., Sehra, V., & Vardhan, Y. R. (2021). Hindi-english code mixed hate speech detection using character level embeddings. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1112-1118). IEEE.
https://doi.org/10.1109/ICCMC51019.2021.9418261

Howard, J. W. (2019). Free speech and hate speech. *Annual Review of Political Science*, *22*(1), 93-109.
https://doi.org/10.1146/annurev-polisci-051517-012343

Hussain, A., Khatoon, A., Aslam, A., & Khosa, M. A. (2024). A Comparative Performance Analysis of Machine Learning Models for Intrusion Detection Classification. *Journal of Cybersecurity (2579-0072)*, *6*.
https://doi.org/10.32604/jcs.2023.046915

Hussain, A., & Aslam, A. (2024). Cardiovascular Disease Prediction Using Risk Factors: A Comparative Performance Analysis of Machine Learning Models.
https://doi.org/10.32604/jai.2024.050277

Kamble, S., & Joshi, A. (2018). Hate speech detection from code-mixed hindi-english tweets using deep learning models.
https://doi.org/10.48550/arXiv.1811.05145

Malmasi, S., & Zampieri, M. (2017). Detecting hate speech in social media.
https://doi.org/10.48550/arXiv.1712.06427

Nasser Alsager, H. (2021). Towards a Stylometric Authorship Recognition Model for the Social Media Texts in Arabic (2021). *Arab World English Journal* (AWEJ) *11* (4) Available at SSRN: https://ssrn.com/abstract=3764890

Park, J. H., & Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter.
https://doi.org/10.48550/arXiv.1706.01206

Pariyani, B., Shah, K., Shah, M., Vyas, T., & Degadwala, S. (2021). Hate speech detection in twitter using natural language processing. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 1146-1152). IEEE.
https://doi.org/10.1109/ICICV50876.2021.9388496

Rong, X. (2014). word2vec parameter learning explained.
https://doi.org/10.48550/arXiv.1411.2738

Salawu, S., He, Y., & Lumsden, J. (2017). Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on Affective Computing*, *11*(1), 3-24. https://doi.org/10.1109/TAFFC.2017.2761757

Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media* (pp. 1-10). https://doi.org/10.18653/v1/W17-1101

Wei, B., Li, J., Gupta, A., Umair, H., Vovor, A., & Durzynski, N. (2021). Offensive language and hate speech detection with deep learning and transfer learning. https://doi.org/10.48550/arXiv.2108.03305