Hate Speech and Offensive Content: Harnessing Machine Learning for Reliable Analysis and Detection

Deepika Varshney*, Megha Rathi
Department of Computer Science & Information Technology, Jaypee Institute of Information Technology, Noida, India
E-mail: deepikavarshney06@gmail.com, megha.rathi@mail.jiit.ac.in
*Corresponding author

Keywords: hate speech, RF, LR, TF-IDF, NLP, offensive content, machine learning

Received: September 24, 2024

The escalating prevalence of hate speech on social media necessitates effective detection mechanisms to foster a safe and inclusive online community. This research paper aims to enhance hate speech detection accuracy by evaluating the performance of diverse machine learning algorithms: Random Forest (RF), Logistic Regression (LR), and K-Nearest Neighbors (KNN). A diverse dataset comprising text samples from various online platforms, encompassing a wide spectrum of hate speech instances, was meticulously collected. The data underwent careful preprocessing involving tokenization, stemming, and stop-word removal to enhance data quality. Additionally, feature extraction techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings were employed to effectively represent the textual content. The dataset was divided into training and testing sets, and the selected machine learning algorithms were trained on the former. Fine-tuning of hyperparameters was performed using cross-validation techniques to optimize their performance. Evaluation metrics, including accuracy, precision, recall, and F1-score, were employed to assess the models' effectiveness. The experimental findings revealed promising outcomes for hate speech detection across all three algorithms. Notably, Count Vectorizer features demonstrated excellent performance, with Random Forest achieving an accuracy of 0.942 for binary hate speech analysis and Logistic Regression achieving an accuracy of 0.897 for multi-class hate speech analysis, followed by LR and KNN.

Povzetek: Prispevek analizira več klasičnih modelov strojnega učenja in značilk (npr. Count Vectorizer, TF-IDF) za zaznavanje sovražnega govora, pri čemer se najboljši rezultati doseženi z algoritmom naključnih gozdov.

1 Introduction

Hate speech is an alarming issue that we are facing in today's digital world. It refers to harmful and offensive language used to target individuals or groups based on their race, religion, gender, or other characteristics. Unfortunately, social media platforms and online communities have become breeding grounds for spreading such toxic content. As a result, ensuring a safe and inclusive online environment has become more challenging than ever. Detecting hate speech has become a crucial research area to combat this problem. By developing effective mechanisms to automatically identify and address hate speech, we can strive to create a respectful and welcoming online space for everyone.

Certainly! Here are a few real-time examples: Social Media Platforms: Hate speech detection is crucial for maintaining a safe and inclusive environment on social media platforms. For example, platforms like Twitter and Facebook employ hate speech detection algorithms to identify and remove offensive content, ensuring user safety and promoting positive online interactions. Online Communities and Forums: Hate speech detection is essential for monitoring and moderating online communities and forums. Platforms like Reddit and Stack Exchange rely on hate speech detection techniques to identify and remove discriminatory or abusive language, fostering a respectful and inclusive environment for users. Hate speech detection plays a vital role in managing comment sections on news websites. By automatically detecting and filtering out hate speech, news organizations can maintain a constructive and respectful space for readers to engage in discussions.

Chatbots and Virtual Assistants: Hate speech detection is important in the development of chatbots and virtual assistants. By integrating hate speech detection algorithms, these AI-powered systems can respond appropriately to user queries while avoiding the propagation of offensive or harmful content. Online Gaming Communities: Hate speech detection is crucial in online gaming communities to prevent toxic behavior and harassment. By implementing hate speech detection mechanisms, gaming platforms can create a more inclusive and enjoyable gaming experience for players. These examples highlight the practical applications of hate speech detection in various online platforms and communities, emphasizing the significance of your research in addressing the challenges associated with identifying and mitigating hate speech.

Here is a more detailed explanation of hate speech and what is not considered hate speech

Hate speech:

Hate speech refers to any form of expression, whether verbal, written, or symbolic, that targets and discriminates against individuals or groups based on attributes such as race, ethnicity, religion, gender, sexual orientation, national origin, or other protected characteristics. - It typically involves the use of offensive, derogatory, or threatening language that aims to demean, dehumanize, or incite harm, violence, or discrimination against the targeted individuals or communities. - Hate speech can take various forms, including direct calls for violence, spreading stereotypes or derogatory language targeting specific groups, and advocating for the exclusion or denial of rights based on characteristics. - The impact of hate speech can be significant, as it can contribute to the marginalization, stigmatization, and psychological harm of targeted individuals or communities. Some of the examples of hate speech is shown in Figure 1.

Figure 1: Disturbing newspaper report reveals alarming cases of bullying in our communities

Not considered hate speech:

- Expressing a different political opinion or ideology without promoting violence or discrimination against a specific group. It is important to distinguish between expressing disagreement or criticism and engaging in hate speech. - Engaging in a respectful debate or discussion about religious beliefs or practices without resorting to derogatory language or incitement of harm. It is possible to discuss differing religious perspectives without engaging in hate speech. - Expressing personal preferences or beliefs without advocating for the denial of rights or promoting discrimination against individuals based on their sexual orientation or gender identity. It is important to respect the rights and dignity of individuals while expressing personal beliefs.

It is crucial to note that the distinction between hate speech and other forms of expression can sometimes be subjective and context-dependent. Legal definitions and interpretations of hate speech may vary across different jurisdictions, so it is important to consult local laws and guidelines when assessing specific cases. Additionally, promoting tolerance, understanding, and respectful dialogue can help foster a more inclusive and harmonious society.

Previous works mostly focused on binary class classification on textual data, distinguishing hate speech from socially acceptable texts. It also emphasized the difficulty of correctly identifying hate speech when mixed with profanity [14].

We delve into the effects of pre-training models for hate speech classification and present some interesting findings. Firstly, we discovered that pre-training solely on hateful tweets doesn't necessarily lead to the best results. Surprisingly, the model trained on a random subset of tweets performed better on two out of three downstream datasets in Marathi and both downstream datasets in Hindi, in terms of macro-F1. Additionally, the model trained on non-hateful data outperformed the one trained on hateful content for most tasks. However, it's worth noting that both hateful models still performed better than the baseline MuRIL model, indicating some benefits of hateful pre-training, but it's not the most optimal approach. We also explored the impact of monolingual retraining versus multilingual models.

The results showed that the MuRIL multilingual model, trained on a vast dataset of 17 Indian languages and billions of tokens, consistently underperformed on all datasets. In contrast, our models retrained on Hindi and Marathi tweets demonstrated significantly better performance than MuRIL. This led us to speculate that focusing on retraining with substantial corpora of a specific language enhances multi-lingual pre-training, potentially outperforming the cumulative semantic knowledge gained from training on multiple large-sized corpora of different languages. Moreover, our experiments revealed that the models pre-trained on a comprehensive dataset of 40 million tweets performed the best on all downstream tasks. Models like MahaTweetBERT and HindTweetBERT outperformed all other variants across different datasets. This result underscores the importance of large-scale pretraining, offering a robust benchmark for future research in hate speech detection.

To sum up, our study provides empirical insights into the impact of hateful pre-training on hate speech classification. We conducted experiments with various pre-training strategies and downstream tasks, showing that the effectiveness of pre-training is not solely determined by focusing on hateful or non-hateful content. Instead, retraining with monolingual datasets and leveraging a substantial corpus for pre-training proves to be more advantageous. The models pre-trained on 40 million tweets emerged as the top performers, showcasing the significance of large-scale pretraining in this domain. These findings hold true for both Marathi and Hindi languages, adding to the robustness of our observations [15]. The rest of the section is organized as follows. The

Section 2 is focused on previous reviews and literature work done in this area, Whereas Section 3 is focused on Methodology part including features and techniques. The Section 4, thoroughly discussed the results and the comparison analysis is shown in Section 5. Finally, the paper is concluded in Section 6.

2 Related work

Hate speech detection has emerged as a critical task in natural language processing and social media analysis. Numerous studies have been conducted to address this challenge, employing various machine learning algorithms to effectively identify and combat hate speech and offensive content on online platforms. Among the widely used algorithms, Logistic Regression (LR), Random Forest (RF), and KNearest Neighbors (KNN) have shown promising results in different contexts. In the efforts to identify and address hate speech on online platforms, researchers have relied on different datasets to develop and assess hate speech detection models. One prominent dataset in this domain is the "Hate Speech Identification" dataset from data. world [3]. This dataset has proven to be a valuable asset for both researchers and professionals working on hate speech detection. The "Hate Speech Identification" dataset contains a diverse set of text samples collected from social media platforms and other online sources. Each text is labeled to indicate whether it contains hate speech, offensive language, or is non-hateful. Having labeled data like this is crucial for training supervised machine learning models to detect hate speech effectively. Researchers have made use of this dataset to train and evaluate various algorithms, such as logistic regression, support vector machines, decision trees, and deep learning-based models, among others. By utilizing this dataset, the goal is to create accurate and robust hate speech detection systems capable of identifying offensive content and fostering a safer online environment. [3], Hate Speech Identification Dataset. data. world. The research conducted a comprehensive analysis of hate speech detection using various text classification techniques. They performed 24 experiments and presented the results which showcased precision, recall, F-measure, and accuracy for different feature representations and classifiers. They found that SVM with TFIDF and bigram features performed the best, achieving high recall, precision, accuracy, and F-measure. On the other hand, KNN and MLP classifiers with TFIDF and bigram features showed the worst performance. In their study, they compared three feature engineering techniques: bigram features with TFIDF, Word2vec, and Doc2vec. Bigram features with TFIDF emerged as the most effective, with only a slight difference from Doc2vec. Additionally, SVM outperformed all eight classifiers, including AdaBoost and RF, while LR, DT, NB, KNN, and MLP demonstrated relatively lower performance. This research is crucial for providing a baseline for future studies on automatic hate speech detection using different text classification methods. It holds scientific value as it utilized multiple scientific measures for evaluation. However, they acknowledged two limitations. Firstly, the proposed ML model's real-time prediction accuracy was found to be inefficient for the data. Secondly, their model classified hate speech into three categories without identifying severity levels. The researchers plan to address these limitations in future work by exploring lexicon-based techniques, collecting more data instances for better learning, and developing a model capable of predicting severity levels of hate speech messages [1]. The best performing model for hate speech detection has impressive overall precision, recall, and F1 score at 0.91, 0.90, and 0.90, respectively. However, when we take a closer look at its performance on the hate speech class, we find that about 40% of hate speech tweets are misclassified. The precision and recall for the hate class are only 0.44 and 0.61, indicating a significant issue with correctly identifying hateful content. The model's bias is evident in the misclassification patterns, with most errors occurring in the upper triangle of the confusion matrix. This means that the model tends to label tweets as less hateful or offensive than they actually are, which can be problematic. Interestingly, there are fewer cases where offensive or innocuous tweets are wrongly classified as hate speech, which is represented in the lower triangle of the confusion matrix. To tackle this problem and avoid confusing hate speech with offensive language, researchers suggest using lexical methods to identify potentially offensive terms. However, they also acknowledge that these methods have limitations in accurately detecting hate speech. To improve hate speech classification, the study proposes finding alternative sources of training data that can identify hate speech without solely relying on specific keywords or offensive language. [2] Automated Hate Speech Detection and the Problem of Offensive Language. These models are used for classification tasks and were named CNN-GRU, BiRNN, BiRNN-Attn, BiRNN-HateXplain, BERT, and BERT-HateXplain. To assess their performance, various metrics were used, such as Accuracy, Macro F1 score, and AUROC, which measures the model's ability to distinguish between classes. The researchers also evaluated the models for bias and explainability. They used specific metrics like Generalized Mean Bias (GMB) to determine if the models were biased towards certain subgroups. Additionally, they looked at explainability metrics like Compatibility and Sufficiency to understand how well the models' provided explanations for their decisions. One interesting finding was that the BERT-HateXplain model, which used Attention-based token selection, performed the best overall. It achieved high accuracy, F1 scores, and AUROC, indicating its effectiveness in classifying the data. Moreover, it demonstrated lower bias in its predictions and showed good explainability, providing understandable reasons for its decisions. These results shed light on the strengths and weaknesses of each model [10]. Some various studies and methods employed for hate speech detection. Here's a summary of the findings: The study used a dataset of Islamophobic hate speech tweets and applied a one-versus-one SVM classifier, achieving an accuracy of 0.77.

However, the dataset was specific to the UK context and didn't consider word context. This research used tweets and trained one-versus-rest SVM classifiers for each class, achieving an F1-measure of 0.91. The dataset focused on sexual-orientation hate speech. The SVM model achieved a high F1-score of 0.97, but for the sexual-orientation hate class, the F1-score was lower at 0.51. This study used tweets and employed the J48 graft model to classify tweets into clean, offensive, and hateful (mixed with offensive hate) classes. The F1-measure was 0.78. The paper dealt with Automatic Misogyny Identification using datasets like AMI IberEval, AMI EvalIta, and SRW. The LR classifier achieved accuracy values for the respective datasets. It noted that general sexist tweets may contain hidden sentiments of hate or misogyny. The research used tweets and performed LR classification for three classes (racism, sexism, none). The F1-score was 0.73, with false positives for multi-class labels having an F1-score of 0.53. EVALITA shared task 2018 dataset was used, and LR achieved an accuracy. However, the misogyny classification had a low F1-score of 0.3. This research worked with English tweets and Spanish tweets, using SVM for hate speech detection. The F1-measures for evaluation datasets Task A and Task B were 0.38 and 0.37, respectively. The method had limitations in achieving high performance. The Semeval-2019 task 5 focused on detecting hate speech against immigrants and women on Twitter. The LIBSVM with RBF model achieved accuracies for the three tasks. The study worked with sarcastic tweets and used the RF classifier with accuracy. However, some sarcastic tweets weren't categorized as hate speech, and there were disagreements about labelling sarcasm. The research used Crowd Flower and HASOC datasets, employing SVM with GLOVE embedding. The accuracies achieved were for the Crowd Flower dataset and for the HASOC dataset. The classification focused on binary classes of hate, offensive, and neither. The MMHS150K dataset was used, and the LDA model achieved an F1-score. The study found that using images in the dataset didn't significantly improve results compared to textual models. This research worked with tweets and used MCD + LSTM for classification. However, the classifiers were trained on three classes (hateful, abusive, or neither) despite the dataset having more categories. The study used tweets and employed GRU + CNN for identification of racist and sexist tweets. However, it struggled to correctly identify the 'both' category due to limited examples. The authors of [11], conducted experiments using two different classifiers: Support Vector Machines (SVM) with a radial basis function kernel and the Random Forest Classifier. Since the feature vectors formed were quite large, we employed the chi-square feature selection algorithm to reduce the vector size to 12,004. The Scikit-learn library was utilized for training our classifiers, and throughout the experiments, we conducted 10-fold cross-validation. The results were presented showing the accuracy of each feature and the overall accuracy when using all features for SVM and the Random Forest Classifier, respectively. SVM outperformed the Random Forest Classifier, achieving the highest accuracy of 71.7% when all features were used. Notably, Character N-Grams proved to be the most efficient feature for SVM, while Word N-Grams yielded the highest accuracy for the Random Forest Classifier. We introduced a valuable annotated corpus of Hindi-English code-mixed text in this research, which included tweet IDs along with corresponding annotations for hate speech and normal speech. Additionally, the words in the tweets were annotated with their source language. Our classification system utilized various features, including character n-grams, word n-grams, punctuations, negation words, and a hate lexicon. The best accuracy of 71.7% was achieved when incorporating all features in the feature vector, using SVM as the classification system. The research [12], evaluated different word embedding, including Word2Vec (e.w2v), Twitter (e.twt), and GloVe (e.glv), using various hate speech detection methods. The tables show the results for three implemented state-of-the-art methods (Table 7) and two proposed methods, CNN+sCNNN and CNN+GRU (Table 8). Interestingly, there was no consistent pattern where one type of word embedding outperformed others on all tasks and datasets. Even though Twitter-based embeddings (e.twt) had better coverage of hashtags, they did not consistently achieve the best results. This could be due to the influence of context window size during training, where a large window captured domain relevance but lacked functions of words. Results show that the topical relevance of words might be more critical for hate speech classification. Despite being less complete, Word2Vec embeddings (e.w2v) performed better for racism tweets, while the more comprehensive Twitter embeddings (e.twt) excelled for sexism tweets. This observation aligns with previous findings showing that performance on intrinsic tasks (e.g., word similarity) might not directly correlate with performance on extrinsic or downstream tasks like hate speech detection. In conclusion, the research suggests that the quality of topical relevance captured by word embeddings might be more relevant for hate speech detection than the level of coverage. Empirically, Word2Vec and GloVe embeddings trained on larger, context-rich corpora performed competitively with Twitter-based embeddings for hate speech detection on Twitter datasets [13]. The authors investigated how well different features could be used to detect hate speech. They trained classifiers using various types of features, such as character n-grams and word n-grams, and also combined all the features into a single model. To evaluate the results, they compared them against a baseline model that predicted the majority class and an "oracle" model, which represented the best possible performance. The majority class baseline had a high accuracy because of the imbalanced distribution of classes in the data. On the other hand, the oracle achieved an impressive 91.6% accuracy, indicating that the features alone couldn't accurately classify a significant portion of the samples. Interestingly, the character n-grams, especially the 4-grams, performed quite well in identifying hate speech. Word unigrams also showed good results, but the accuracy decreased when using word

bigrams, trigrams, and skip-grams. However, the skip-grams might be capturing longer dependencies that complement the other feature types. When all the features were combined, the model's performance did not surpass that of the character 4-grams model. Additionally, the combined model significantly increased the number of features, making it less clear if it truly captured the diverse information provided by all the different feature types. The learning curve for the character 4-grams model showed that accuracy continuously improved as the number of training examples increased. However, the rate of improvement slowed down after reaching 15,000 training instances. Overall, the researchers applied various text classification techniques to distinguish between hate speech, profanity, and other types of texts. The best-performing model achieved 78% accuracy using character 4-grams. The study was unique as it addressed the challenging task of differentiating hate speech and profanity, a scenario that had not been extensively explored before in the context of social media. Previous works mostly focused on binary classification, distinguishing hate speech from socially acceptable texts. It also emphasized the difficulty of correctly identifying hate speech when mixed with profanity [14].

To sum up, our study provides empirical insights into the impact of hateful pre-training on hate speech classification. We conducted experiments with various pre-training strategies and downstream tasks, showing that the effectiveness of pre-training is not solely determined by focusing on hateful or non-hateful content. Instead, retraining with monolingual datasets and leveraging a substantial corpus for pre-training proves to be more advantageous. The models pre-trained on 40 million tweets emerged as the top performers, showcasing the significance of large-scale pretraining in this domain. These findings hold true for both Marathi and Hindi languages, adding to the robustness of our observations [15].

3 Proposed methodology

Below is the workflow diagram of the Proposed Methodology used in the research as shown in Figure 2:

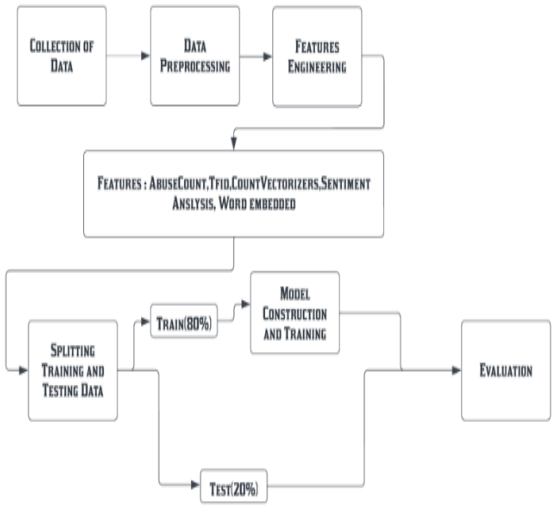

Figure 2: Proposed methodology

3.1 Data collection

In this research study, we collected publicly available hate speech tweets dataset named as Twitter hate speech detection dataset. The dataset is compiled and labeled by CrowdFlower. In this dataset, the tweets are labeled into three distinct classes, namely, hate speech, offensive, and neutral This dataset has 25296 number of tweets. Of these, 16.8 percent of tweets belong to class hate speech. In addition, 77.43 percent of tweets belong to offensive class and the remaining 5.77 percent tweets are neutral. The details of this distribution are also shown in Table 1. To remove the imbalance and possible biases the under sampling and oversampling technique has been used. Below is the table with the collected data:

Table 1: Details of dataset

	Class	Total Instances	Percentage
0	Neutral	1430	5.77
1	Offensive	19190	77.43
2	Hate Speech	4163	16.80
	Total	24783	

3.2 Data Pre-processing

In our pursuit of advancing hate speech detection, a crucial aspect of our research methodology involves the application of text pre-processing techniques. The primary objective is to prepare the dataset of hate speech for classification, by filtering out irrelevant noise and enhancing the overall quality of the input data. By carefully curating and optimizing the text data, our goal is to improve the accuracy and performance of our hate speech detection models. To initiate the text pre-processing process, we start by converting all the hate speech tweets into lower case. This step ensures uniformity in the text and avoids potential discrepancies that may arise due to varying capitalization. By standardizing the text, we create a level playing field for subsequent analysis. Next, we employ pattern matching techniques to remove various elements such as URLs,

usernames, white spaces, hashtags, and punctuation marks from the tweets. These elements are often extraneous and do not contribute significantly to the core context of hate speech. Removing them helps reduce noise and streamlines the data, making it more suitable for further analysis. Furthermore, we implement the removal of stop-words from the hate speech data. Stop-words are commonly occurring words, such as "the," "is," and "and," which carry limited value in conveying the underlying sentiment of hate speech. By eliminating these words, we reduce dimensionality and enhance the efficiency of subsequent classification tasks. In addition to the initial pre-processing steps, we perform tokenization and stemming on the pre-processed tweets. Tokenization involves breaking down each tweet into individual tokens or words, enabling a more granular analysis of the text. This step is crucial for feature extraction and capturing the nuanced language used in hate speech. The stemming process, facilitated by the Porter stemmer algorithm, converts words to their root forms, consolidating similar variations of words. By transforming words like "offended" to "offend," the stemming process simplifies the feature space and contributes to a more effective classification process. Overall, our text pre-processing methodology plays a critical role in preparing the hate speech dataset for classification. By optimizing the data and removing noise, we aim to enhance the efficacy of our hate speech detection models, ultimately contributing to the advancement of hate speech detection and promoting a safer digital environment.

3.3 Features

1. Sentiment analysis: Sentiment analysis is a natural language processing technique used to determine the emotional tone or sentiment of a piece of text. In the context of hate speech detection, sentiment analysis can be utilized to identify the overall emotional context of a text, whether it contains negative, offensive, or hateful sentiments. By incorporating sentiment analysis as part of hate speech detection, we can gain valuable insights into the emotional intent behind the text, aiding in the accurate classification of hate speech instances.

2. Word embeddings:(WE) Word embeddings are dense vector representations of words in a high-dimensional space. In hate speech detection, word embeddings can be employed to capture the semantic meaning of words and their relationships with other words in the context of hate speech. By representing words as numerical vectors, word embeddings enhance the model's ability to understand the underlying linguistic patterns in hate speech, allowing for more nuanced and effective classification.

3. Count vectorizer (CV): Count vectorizer features are a fundamental method for transforming raw text data into a format suitable for machine learning algorithms. By tokenizing and counting the occurrences of words or n-grams, it creates a sparse numerical matrix representing the frequency of each term in each document. This matrix serves as input for algorithms like SVM, Naive Bayes, or logistic regression in various NLP tasks, including sentiment analysis, topic modelling, and spam detection. Count vectorizer preserves the basic linguistic structure of the text while ignoring grammar and context. Despite its simplicity, it provides a strong foundation for text-based ML models, allowing them to learn patterns, relationships, and distinctions within text data efficiently.

4. TF-IDF (Term frequency-inverse document frequency) (TI): TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents. In the context of hate speech detection, TF-IDF can be employed to weigh the relevance of words in distinguishing hate speech from non-hate speech content. By assigning higher weights to words that are prevalent in hate speech but less common in other texts, the model can effectively identify key indicators of hate speech.

5. Abuse count (AC): Count abuse features refer to a set of numerical attributes extracted from textual data, where each attribute represents the frequency of predefined offensive words or phrases. These features are utilized to train machine learning models for the purpose of detecting and classifying offensive language, hate speech, or abusive content. By quantifying the occurrence of offensive terms within the text, count abuse features provide valuable input to the ML algorithms, enabling the development of effective and efficient models for content moderation and sentiment analysis in online platforms and social media.

Although hate speech detection has been achieved with deep learning modelssuch as BERT and LSTMs, this study focuses on traditional machine learning models due to their interpretability, lower computational cost, and effectiveness on smaller datasets. In this work we have mainly focused on Machine learning based models. From the previous studies we have analysed that Random Forest classifier outperform all other classifier. The authors of [16], reported 86% accuracy on Random Forest. From the earlier report Random Forest, logistic regression, KNN are prominently used classifier on text for the detection of Hate speech. Taking this into consideration, we have explored and presented the in-depth performance analysis of the machine learning models on diverse set of features considering both the aspect of Binary and multiclass classification. The evaluation has been done on the feature including Abuse count, sentiment analyser, word2vec, TF-IDF and count vectorizer. It has been analysed that the performance is enhanced when using binary class classification instead of Multi class classification and analysis shows Random Forest again performing best among others. In our research work on hate speech detection, the integration of sentiment analysis, word embeddings, n-gram, and TF-IDF plays a critical role in developing a robust and accurate hate speech detection model. By leveraging these techniques, we aim to improve the model's understanding of the emotional context,

semantic meaning, and linguistic patterns in hate speech. The combination of these methodologies enhances our ability to identify hate speech instances more effectively and contributes to fostering a safer and more inclusive digital environment.

4 Results

In this section, we present the results of our hate speech detection model and the comparative analysis of different machine learning algorithms: Logistic Regression (LR), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Random Forest (RF), AdaBoost and SupportVectorMachine (SVM). The Table 2, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11 shows the performance analysis of our proposed work incorporating diverse set of features using both binary and multi class classification. We have considered the following parameters Accuracy (Acc), Weighted Precision (WP), Weighted Recall (WR) Weighted F1 score (WF1).

Table 2: Result analysis by incorporating Abuse count feature for binary class classification.

Model	Acc	WP	WR	WF1
Logistic Regression	0.94	0.94	0.94	0.97
K-Nearest Neighbors	0.94	0.91	0.94	0.97
Classification (CART)	0.94	0.91	0.94	0.94
Random Forest	0.94	0.94	0.94	0.94
AdaBoost	0.94	0.94	0.94	0.97
(SVM)	0.94	0.94	0.94	0.97

Table 3: Result analysis by incorporating Abuse count feature for Multi class classification

Model	Acc	Precision	Recall	F1score
LR	0.818	0.770	0.820	0.770
K-NN	0.800	0.790	0.810	0.770
(CART)	0.821	0.790	0.810	0.790
RF	0.821	0.790	0.810	0.790
AdaBoost	0.821	0.790	0.810	0.790
(SVM)	0.821	0.790	0.810	0.790

Table 4: Result analysis by incorporating Sentiment analyzer feature for Binary class classification

Model	Acc	WP	WR	WF1
LR	0.93	0.88	0.93	0.90
K-NN	0.93	0.90	0.93	0.91
(CART)	0.89	0.89	0.89	0.89
RF	0.93	0.88	0.92	0.90
AdaBoost	0.93	0.88	0.93	0.90
(SVM)	0.93	0.88	0.93	0.90

Table 5: Result analysis by incorporating sentiment analyzer feature for Multi class classification

Model	Acc	WP	WR	WF1
LR	0.77	0.59	0.77	0.68
K-NN	0.76	0.69	0.76	0.70
(CART)	0.71	0.68	0.71	0.68
RF	0.75	0.68	0.75	0.71
AdaBoost	0.77	0.63	0.77	0.67
(SVM)	0.77	0.59	0.77	0.68

Table 6: Result analysis by incorporating word2vec feature for binary class classification

Model	Acc	WP	WR	WF1
LR	0.93	0.89	0.92	0.90
K-NN	0.93	0.89	0.93	0.91
(CART)	0.88	0.89	0.88	0.88
RF	0.93	0.89	0.92	0.90
AdaBoost	0.93	0.88	0.93	0.90
(SVM)	0.93	0.87	0.93	0.90

Table 7: Result analysis by incorporating Word2vec feature for Multi class classification

Model	Acc	WP	WR	WF1
LR	0.82	0.78	0.82	0.79
K-NN	0.80	0.75	0.80	0.76
(CART)	0.73	0.74	0.73	0.73
RF	0.82	0.77	0.82	0.78
AdaBoost	0.80	0.74	0.80	0.76
(SVM)	0.82	0.75	0.82	0.78

Table 8: Result analysis by incorporating TF-IDF feature for Binary class classification

Model	Acc	WP	WR	WF1
LR	0.883	0.940	0.800	0.850
KNN	0.865	0.940	0.940	0.940
CART	0.873	0.990	0.830	0.900
RF	0.887	0.990	0.830	0.900
AdaBo.	0.876	0.930	0.730	0.810
(SVM)	0.774	0.890	0.940	0.910

Table 9: Result analysis by incorporating TF-IDF feature for Multi class classification

Model	Acc	WP	WR	WF1
LR	0.7796	0.66	0.78	0.72
K-NN	0.7816	0.65	0.78	0.69
(CART)	0.7916	0.66	0.79	0.70
RF	0.8050	0.67	0.80	0.73
AdaBoost	0.7916	0.67	0.80	0.73
(SVM)	0.7771	0.65	0.78	0.69

Table 10: Result analysis by incorporating Count Vectorizer feature for Binary class classification

Model	Acc	WP	WR	WF1
LR	0.942	0.934	0.939	0.936
K-NN	0.940	0.935	0.939	0.935
(CART)	0.937	0.931	0.941	0.926

RF	0.942	0.934	0.941	0.925
AdaBoost	0.942	0.938	0.941	0.919
(SVM)	0.942	0.934	0.938	0.926

Table 11: Result analysis by incorporating Count Vectorizer feature for Multi Class classification

Model	Acc	WP	WR	WF1
LR	0.897	0.870	0.870	0.870
K-NN	0.777	0.880	0.900	0.890
(CART)	0.867	0.780	0.780	0.750
RF	0.879	0.870	0.880	0.870
AdaBoost	0.889	0.880	0.890	0.860
(SVM)	0.875	0.890	0.890	0.870

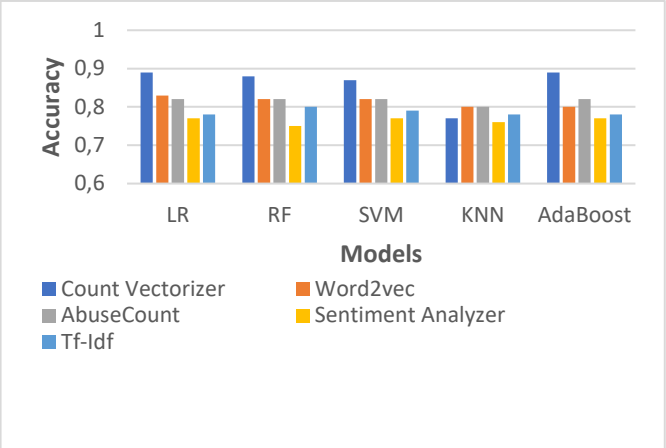

Figure 3: Performance metric analysis (Accuracy) on diverse set of features on our proposed model.

5 Comparative analysis

We have compared accuracy of our work with the earlier state-of-the-art model on the Accuracy performance metric of the features and ML model used in Automatic Hate Speech Detection using Machine Learning: A Comparative Study by Sindhu Abro Sarang Shaikh, Zafar Ali Sajid Khan, Ghulam Mujtaba. The Table 12 and Figure3 shows the performance metric analysis (Accuracy) on diverse set of features on our proposed model. Whereas, the Table 13 shows the state-of-the-art performance metric analysis (Accuracy) on diverse set of features

Table 12: Performance metric analysis (Accuracy) on diverse set of features on our proposed model.

Features	LR	RF	SVM	KNN	AdaBoost
CV	0.89	0.88	0.87	0.77	0.89
W2V	0.83	0.82	0.82	0.80	0.80
AC	0.82	0.82	0.82	0.80	0.82
Sentiment Analyzer	0.76	0.77	0.77	0.75	0.77
Tf-Idf	0.78	0.80	0.79	0.78	0.78

Table 13: State-of-the-art performance metric analysis (Accuracy) on diverse set of features ML model used in Automatic Hate Speech Detection using Machine Learning: A Comparative Study by Sindhu Abro, Sarang Shaikh, Zafar Ali Sajid Khan, Ghulam Mujtaba.

Features	LR	RF	SVM	KNN	AdaBoost
Bigram	0.75	0.75	0.79	0.57	0.78
Word2vec	0.68	0.72	0.68	0.73	0.61
Doc2vec	0.72	0.67	0.72	.0.65	0.67

In our comparative study, we evaluated the performance of various machine learning algorithms on different feature extraction techniques for a specific task (not specified in the question). The research paper's findings were based on three feature extraction methods: Bigram, Word2vec, and Doc2vec. They used five classification algorithms: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and AdaBoost.

On the other hand, our research findings included five different feature extraction techniques: Count Vectorizer, Word2vec, AbuseCount, Sentiment Analyzer, and Tf-Idf. We also assessed the performance of the same five classification algorithms: LR, RF, SVM, KNN, and AdaBoost. Count Vectorizer: In our study, Count Vectorizer outperformed Bigram from the research paper in all classification algorithms except for SVM, where the two were comparable. Our findings show higher accuracy for Count Vectorizer across the board, with AdaBoost achieving the highest accuracy of 0.89. Word2vec: In both studies, Word2vec demonstrated competitive performance, although our research found slightly higher accuracy for this feature extraction method in all classifiers except for KNN, which was the same as in the research paper. AbuseCount and Sentiment Analyzer: These two features were not present in the research paper's study, so there's no direct comparison. However, in our findings, both AbuseCount and Sentiment Analyzer achieved consistent and competitive accuracy scores across all classifiers.

Tf-Idf: Our research showed comparable performance to Word2vec in most classifiers, and it achieved reasonable accuracy overall. In summary, our research findings indicate that Count Vectorizer and AdaBoost were the most effective combination for the task, consistently outperforming the other feature extraction methods and classifiers. AbuseCount and Sentiment Analyzer also showed promising results, while Tf-Idf and Word2vec demonstrated competitive performance but did not consistently outshine the other methods. It's essential to consider the specific task and dataset characteristics when selecting the most appropriate feature extraction and classification techniques.

6 Conclusion

In conclusion, this machine learning project focused on sentiment analysis using various feature extraction techniques, including Word Embeddings, Count

Vectorizer, TF-IDF, and Abuse Count features. We evaluated multiple ML models, such as Logistic Regression, K-Nearest Neighbors, Classification and Regression Trees, Random Forest, Ada Boost, and SVM, for both binary and multi-class sentiment analysis. The results indicated that Abuse Count features in combination with Logistic Regression achieved the highest accuracy of 0.94 for binary sentiment analysis and Random Forest with an accuracy of 0.821 for multi-class sentiment analysis. These features showed a remarkable ability to capture offensive language and emotions in text, making them effective for sentiment classification tasks. Furthermore, Word2Vec features demonstrated competitive performance, with Logistic Regression achieving an accuracy of 0.93 for binary and 0.82 for multi-class sentiment analysis. The Word2Vec embeddings effectively captured semantic meaning and relationships between words, contributing to their strong performance. On the other hand, TF-IDF features showed good performance in binary sentiment analysis, with Random Forest achieving an accuracy of 0.887, and in multi-class sentiment analysis, with Random Forest achieving an accuracy of 0.805.

Lastly, Count Vectorizer features demonstrated excellent performance, with Random Forest achieving an accuracy of 0.942 for binary sentiment analysis and Logistic Regression achieving an accuracy of 0.897 for multi-class sentiment analysis.

Overall, the paper highlighted the importance of choosing appropriate feature extraction techniques based on the task at hand. The combination of Abuse Count features with Logistic Regression and Count Vectorizer features with Random Forest emerged as the best performing models for binary and multi-class sentiment analysis, respectively. These findings can serve as valuable insights for sentiment analysis tasks and text classification in various real-world applications. The limitation of the work that we have not explored deep learning models currently. In the coming future we are also planning to add deep learning models like LSTM or Bert, such as exploring neural network-based approaches or hybrid models that combine the strengths of multiple algorithms. As well as we need to extend our work by doing analysis on other datasets. The current dataset is imbalanced and have to explore techniques to handle it efficiently in future.

References

[1] S. Abro, S. Shaikh, Z. H. Khand, Z. Ali, S. Khan, and G. Mujtaba, "Automatic Hate Speech Detection using Machine Learning: A Comparative Study," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 8, 2020. DOI: 10.14569/IJACSA.2020.0110861

[2] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," in Proceedings of the 11th International AAAI Conference on Web and social media (ICWSM), May 2017. DOI: https://doi.org/10.1609/icwsm.v11i1.14955

[3] Crowdflower. (n.d.). Hate Speech Identification Dataset. https://doi.org/10.6084/m9.figshare.19333298.v1.

[4] Papadopoulou O, Zampoglou M, Papadopoulos S, Kompatsiaris Y, & Denis Teyssou. (2018). InVID Fake Video Corpus v2.0 (Version 2.0) [Data set]. Zenodo. DOI. https://doi.org/10.5281/zenodo.

[5] Children Victims Cyber Bullying. (n.d.). Doi: 10.3389/fpubh.2021.634909.

[6] Facebook Post: PNG Ples Mi Lesu, January 2019.

[7] Online Abuse Picture. Retrieved from https://imature.in/DesktopModules/UserDefinedTable/MakeThumbnail.ashx?image=Images%2fNews_Articles%2fTOI-online-abuse1.jpg&h=150&PortalId=0

[8] BBC News. (2015, August 10). Twitter 'will hide' abusive tweets from public.

[9] Twitter Post: Anti-Bullying Pro, May 2014. Retrieved from https://twitter.com/AntiBullyingPro/status/471225746460254208

[10] Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 8. DOI: https://doi.org/10.1609/aaai.v35i17.17745

[11] Fatimah Alkomah and Xiaogang Ma. (2022). A Literature Review of Textual Hate Speech Detection Methods and Datasets. International Journal of Advanced Computer Science and Applications (IJACSA),13(6),273. DOI. https://doi.org/10.3390/info13060273

[12] Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed S. Akhtar, Manish Shrivastava (2018) A Dataset of Hindi-English Code-Mixed social media Text for Hate Speech Detection (Association for Computational Linguistics). DOI: https://doi.org/10.18653/v1/W18-1105.

[13] Zhang, Z., & Luo, L. (2018). Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. Retrieved from https://arxiv.org/pdf/1803.03662.pdf

[14] Malmasi, S., & Zampieri, M. (2017, December 26). Detecting Hate Speech in social media. DOI: https://doi.org/10.48550/arXiv.1712.06427

[15] Gokhale, O., Kane, A., Patankar, S., Chavan, T., & Joshi, R. (2022, December 11). Spread Love Not Hate: Undermining the Importance of Hateful Pre-training for Hate Speech Detection. DOI: https://doi.org/10.48550/arXiv.2210.04267

[16] Jain, A., and S. Sharma. "Hate Speech Detection based on Word Embedding and Linguistic Features." Indian Journal of Science and Technology 16.41 (2023): 3704-3713.DOI: https://doi.org/10.17485/IJST/v16i41.2128