IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 1, 2022

MODIFIED TF-IDF WITH MACHINE LEARNING CLASSIFIER FOR HATE SPEECH DETECTION ON TWITTER

[1]L Prasanna Lakshmi , [2] Dr MD Sameeruddin Khan, [3]Ramesh Artham

1,3Assistance Professor, Department of CSE, Sree Dattha Institute of Engineering and Science,

[2]Associate Professor, Department of CSE, Sree Dattha Institute of Engineering and Science,

ABSTRACT

Hate speech refers to any form of communication, whether written, spoken, or symbolic, that discriminates, threatens, or incites violence against individuals or groups based on attributes such as race, religion, ethnicity, gender, sexual orientation, or disability. Social media platforms like Twitter have become hotspots for hate speech due to their wide user base and ease of communication. The sheer volume of tweets generated every day makes it impractical to manually review and classify them for hate speech. Traditional methods for hate speech detection often rely on lexicon-based approaches, where predefined lists of offensive or discriminatory terms are used to flag potentially hateful content. However, these methods often struggle to adapt to the constantly evolving nature of hate speech and lack the context required to accurately distinguish between hate speech and other forms of expression. Given the limitations of traditional approaches, there is a need for advanced techniques that can automatically identify hate speech on Twitter. Machine learning classifiers provide a promising solution by leveraging the power of algorithms to learn patterns and features from large datasets. By using a modified TF-IDF approach, we can capture the unique characteristics of hate speech and develop a robust model capable of accurately detecting such content.

1. INTRODUCTION

1.1 Overview

Detecting hate speech on Twitter is a crucial task in the era of social media, where harmful and offensive content can spread rapidly and have real-world consequences. To address this challenge, machine learning classifiers are employed to automatically identify and flag instances of hate speech. These classifiers leverage a combination of natural language processing (NLP) techniques and supervised learning algorithms to analyze the content of tweets. The process begins with data collection, where a diverse dataset of tweets is gathered, encompassing a wide range of topics, languages, and demographics. This dataset is then preprocessed, which includes text cleaning, tokenization, and stemming or lemmatization to standardize the text and reduce dimensionality. Next, the data is split into training and testing sets, with labeled examples of hate speech and non-hate speech tweets. Feature extraction is a critical step, where the textual data is transformed into numerical features that machine learning algorithms can understand. This involves techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings such as Word2Vec or GloVe to represent words in a meaningful way. Various machine learning algorithms can be employed for hate speech detection, including but not limited to Support Vector Machines (SVM),

Random Forests, and deep learning models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs). These models are trained on the labeled data, optimizing their parameters to learn patterns indicative of hate speech. During training, model evaluation is essential to prevent overfitting and ensure generalization. Cross-validation and hyperparameter tuning are often used to achieve the best performance. Evaluation metrics such as precision, recall, F1-score, and accuracy are calculated on the test set to assess the model's performance.

To improve the classifier's robustness, it's essential to address class imbalance since hate speech is relatively rare compared to non-hate speech. Techniques like oversampling, undersampling, or the use of advanced algorithms like SMOTE (Synthetic Minority Over-sampling Technique) can be applied. Once the classifier is trained and validated, it can be deployed to automatically detect hate speech in real-time Twitter data. Users can be alerted or the flagged content can be subject to moderation and content removal policies, thus helping create a safer online environment. Continuous monitoring and model retraining are necessary to adapt to evolving patterns of hate speech. So, machine learning classifiers for hate speech detection on Twitter play a pivotal role in mitigating online toxicity. These models combine data collection, preprocessing, feature extraction, and the application of various algorithms to automatically identify and combat hate speech, contributing to a more inclusive and respectful online community. However, it's important to emphasize the ongoing challenges of false positives and evolving tactics employed by malicious actors, necessitating continuous research and development in this field.

1.2 Research Motivation

The motivation for conducting research on the development of machine learning classifiers for hate speech detection on Twitter is rooted in the pressing need to address the growing issue of online hate speech and its profound societal impacts. Social media platforms like Twitter have become powerful communication channels, enabling individuals to express their thoughts and opinions on a global scale. However, this widespread accessibility has also given rise to the proliferation of hate speech, which encompasses discriminatory, offensive, or harmful content targeting various groups based on race, religion, gender, and more. One of the primary motivations for this research lies in the harmful consequences of unchecked hate speech on social media. Hate speech can lead to real-world harm, including cyberbullying, harassment, psychological trauma, and even incitement to violence. It can also contribute to the exacerbation of societal divisions, undermining efforts to foster inclusivity, diversity, and respectful discourse in the digital realm. Consequently, addressing hate speech on platforms like Twitter is not only an ethical imperative but also critical for maintaining a healthy online environment.

Moreover, the sheer volume of content on Twitter makes manual moderation an impractical solution. Automated hate speech detection using machine learning models is a scalable approach that can assist human moderators in identifying and addressing problematic content efficiently. It can help social media platforms enforce community guidelines and create a safer and more inclusive online space for users. Another key motivation is the evolving nature of hate speech tactics and the need for adaptive solutions. Hate speech is not static; it

evolves with changing cultural, political, and social contexts. Machine learning classifiers provide a flexible framework that can be updated and retrained as new forms of hate speech emerge, making them a valuable tool for combating the ever-changing landscape of online toxicity.

1.3 Problem Statement

The problem statement for the research on machine learning classifiers for hate speech detection on Twitter revolves around the urgent need to combat the pervasive issue of hate speech within the realm of social media, specifically on the Twitter platform. Hate speech on Twitter poses a multifaceted challenge that requires sophisticated technological solutions. The central problem can be articulated as follows:

"In the age of social media, hate speech has emerged as a profound and pervasive issue on platforms like Twitter, manifesting as discriminatory, offensive, and harmful content targeting various demographic groups. This problem presents a two-fold challenge: firstly, the exponential growth of user-generated content on Twitter makes manual moderation impractical, necessitating automated solutions for the timely detection and mitigation of hate speech. Secondly, the dynamic and evolving nature of hate speech tactics, which adapt to changing societal, cultural, and political contexts, demands adaptive machine learning models capable of recognizing emerging forms of hate speech."

This problem statement underscores several critical aspects of the research challenge:

- **Magnitude of the Problem**: Hate speech on Twitter is not a minor issue; it's a pervasive and escalating concern that impacts millions of users and can have far-reaching consequences, both online and offline.
- **Scalability**: The sheer volume of content on Twitter surpasses human moderation capabilities. Therefore, there is a pressing need for automated systems that can efficiently process and analyze vast amounts of data in real-time.
- **Adaptability**: Hate speech is a constantly evolving problem, with perpetrators frequently modifying their language and tactics to avoid detection. Consequently, the research problem necessitates the development of machine learning classifiers that can adapt to these changing patterns effectively.
- **Safety and Inclusivity**: The overarching goal is to create a safer and more inclusive online environment for all users, fostering respectful discourse while protecting vulnerable communities from hate speech-related harm.
- **Ethical and Societal Implications**: The problem extends beyond technical challenges; it also raises ethical and societal questions about the role of social media platforms in curbing hate speech and the implications of automated content moderation.

2. LITERATURE SURVEY

[1] Akuma, S., Lubem, T. et al. detected hate speech from live tweets on Twitter via a combination of mechanisms. The comparison results of Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) with machine learning models of Logistic Regression, Naïve Bayes, Decision Tree, and K-Nearest Neighbour (KNN), is used to select the best performing model. This model which is integrated into a web system developed with Twitter Application Programming Interface (API) is used in identifying live tweets which are hateful or not. The outcome of the comparative study presented showed that Decision Tree performed better than the other three models with an accuracy of 92.43% using TF-IDF which gives optimal results compared to BoW.

[2] Muzakir, Ari, et al. improved performance in the detection of hate speech on social media in Indonesia, particularly Twitter. Until now, the machine learning approach is still very suitable for overcoming problems in text classification and improving accuracy for hate speech detection. However, the quality of the varying datasets caused the identification and classification process to remain a problem. Classification is one solution for hate speech detection, divided into three labels: Hate Speech (HS), Non-HS, and Abusive. The dataset was obtained by crawling Twitter to collect data from communities and public figures in Indonesia.

[3] Ali, Raza, et al. developed an Urdu language hate lexicon, on the basis of this lexicon we formulate annotated dataset of 10,526 Urdu tweets. Furthermore, as baseline experiments, we use various machine learning techniques for hate speech detection. In addition, they use transfer learning to exploit pre-trained FastText Urdu word embeddings and multi-lingual BERT embeddings for our task. Finally, they experiment with four different variants of BERT to exploit transfer learning, and they show that BERT, xlm-roberta and distil-Bert are able to achieve encouraging F1-scores of 0.68, 0.68 and 0.69 respectively, on our multi class classification task. All these models exhibited success to varying degree but outperform a number of deep learning and machine learning baseline models.

[4] Turki, T.; Roy, S.S. et al. presented a computational framework to examine out the computational challenges behind hate speech detection and generate high performance results. First, they extract features from Twitter data by utilizing a count vectorizer technique. Then, they provide the labeled dataset of constructed features to adopted ensemble methods, including Bagging, AdaBoost, and Random Forest. After training, we classify new tweet examples into one of the two categories, hate speech or non-hate speech.

[5] Dascălu, Ș.; Hristea, F et al. explored different transformer and LSTM-based models in order to evaluate the performance of multi-task and transfer learning models used for Hate Speech detection. Some of the results obtained in this paper surpassed the existing ones. The paper concluded that transformer-based models have the best performance on all studied Datasets.

[6] Ababu, Teshome Mulugeta, et al. developed numerous models that were used to detect and classify Afaan Oromo hate speech on social media by using different machine learning algorithms (classical, ensemble, and deep learning) with the combination of different feature extraction techniques such as BOW, TF-IDF, word2vec, and Keras Embedding layers. To

perform the task, we required Afaan Oromo datasets, but the datasets were unavailable. By concentrating on four thematic areas of hate speech, such as gender, religion, race, and offensive speech, we were able to collect a total of 12,812 posts and comments from Facebook.

[7] Mohiyaddeen, Siddiqi, S., et al. hybrid approach combines nine different machine learning algorithms to make one hybrid machine learning model. Additionally, we used the bag-of-words and TF-IDF techniques with the two-gram approach to extract the features. Significant experiments are carried out on the hate speech dataset. The accuracy gained by the hybrid machine learning model is much higher than that of available conventional machine learning models.

[8] Ojo, Olumide Ebenezer, et al. used a binary classification approach to automatically process user contents to detect hate speech. The Naive Bayes Algorithm (NBA), Logistic Regression Model (LRM), Support Vector Machines (SVM), Random Forest Classifier (RFC) and the one-dimensional Convolutional Neural Networks (1D-CNN) are the models proposed. With a weighted macro-F1 score of 0.66 and a 0.90 accuracy, the performance of the 1D-CNN and GloVe embeddings was best among all the models.

[9] Doan, Long-An, et al. developed system to detect hate speech in Vietnamese YouTube comments using machine learning and big data technology. The streaming data from Youtube is processed in real-time using Kafka, Spark, and machine learning technology. Finally, a dashboard powered by Streamlit will be used to display the results.

[10] Toraman, Cagri, et al. designed to have equal number of tweets distributed over five domains. The experimental results supported by statistical tests show that Transformer-based language models outperform conventional bag-of-words and neural models by at least 5% in English and 10% in Turkish for large-scale hate speech detection. The performance is also scalable to different training sizes, such that 98% of performance in English, and 97% in Turkish, are recovered when 20% of training instances are used. They further examine the generalization ability of cross-domain transfer among hate domains. They show that 96% of the performance of a target domain in average is recovered by other domains for English, and 92% for Turkish. Gender and religion are more successful to generalize to other domains, while sports fail most.

[11] Ayo, Femi Emmanuel, et al. proposed a hybrid embedding enhanced with a topic inference method and an improved cuckoo search neural network for hate speech detection in Twitter data. The proposed method uses a hybrid embeddings technique that includes Term Frequency-Inverse Document Frequency (TF-IDF) for word-level feature extraction and Long Short-Term Memory (LSTM) which is a variant of recurrent neural networks architecture for sentence-level feature extraction.

[12] Mossie, Zewdie, et al. proposed approach can successfully identify the Tigre ethnic group as the highly vulnerable community in terms of hatred compared with Amhara and Oromo. As a result, hatred vulnerable group identification is vital to protect them by applying automatic hate speech detection model to remove contents that aggravate psychological harm

and physical conflicts. This can also encourage the way towards the development of policies, strategies, and tools to empower and protect vulnerable communities.

3. PROPOSED SYSTEM

3.1 Overview

Develop an effective hate speech detection system using RFC with a modified TF-IDF feature extraction approach, which is tailored to the nuances of Twitter data and hate speech patterns. Remember that fine-tuning and continuous monitoring are essential to maintain the model's accuracy as online hate speech evolves over time. Figure 4.1 shows the proposed system model. The detailed operation illustrated as follows:

Step 1. Dataset Preprocessing:

- **Data Collection**: Begin by collecting a diverse and well-labeled dataset of tweets containing both hate speech and non-hate speech content. Ensure that the dataset is balanced to address class imbalance issues.
- **Text Cleaning**: Perform text cleaning to remove noise, such as special characters, URLs, and HTML tags. This ensures that the text data is in a standardized format.
- **Tokenization**: Tokenize the cleaned text into individual words or tokens. This step breaks down the text into smaller units for further analysis.
- **Stopword Removal**: Remove common stopwords (e.g., "the," "and," "is") as they don't carry much information for hate speech detection.
- **Stemming or Lemmatization**: Apply stemming or lemmatization to reduce words to their base forms. This helps in reducing the dimensionality of the dataset.
- **Feature Engineering**: Extract additional features if necessary, such as the length of the tweet, the presence of specific keywords or hashtags, or sentiment scores.

Step 2. Modified TF-IDF Feature Extraction:

- **TF-IDF Transformation**: Apply the Term Frequency-Inverse Document Frequency (TF-IDF) transformation to the preprocessed text data. However, modified TF-IDF approach, consider customizing it to better suit hate speech detection:
- **Customized Stopword List**: In the TF-IDF vectorization process, use a customized stopword list that includes domain-specific stopwords relevant to Twitter hate speech. These stopwords might include common slurs, offensive terms, or Twitter-specific jargon that is not present in standard stopwords lists.
- **N-grams**: Include bi-grams or tri-grams in addition to single words. This captures the context in which words or phrases appear, which can be crucial for identifying hate speech.
- **Term Weighting**: Experiment with different term weighting schemes. For example, you can assign higher weights to terms that are known hate speech indicators. This

can be done by manually assigning custom weights to specific terms or using domain-specific dictionaries.

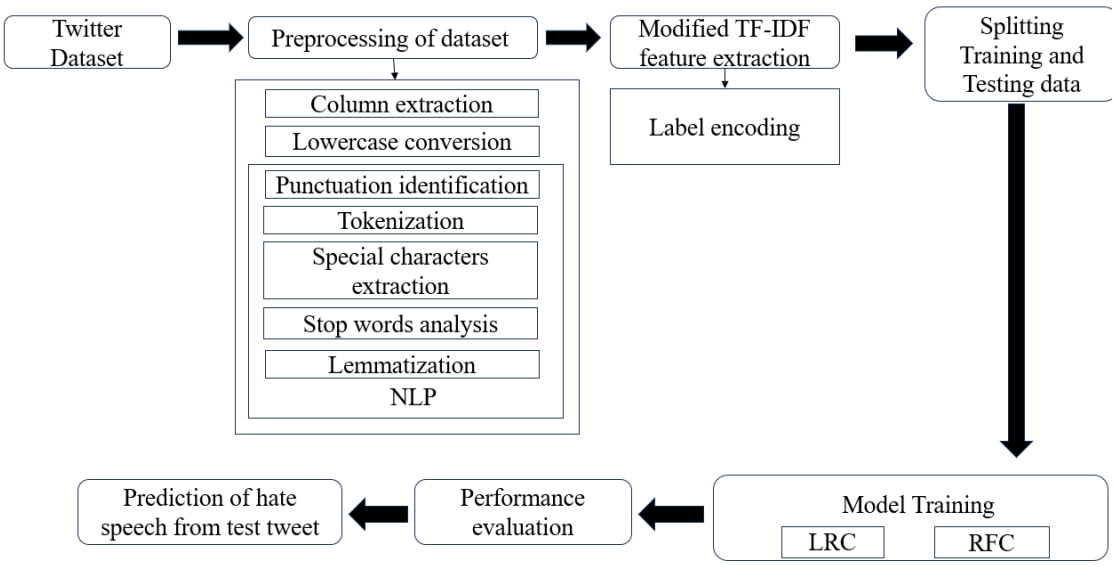

Figure 3.1: Block Diagram of Proposed System.

Step 3. Random Forest Classifier (RFC):

- **Data Splitting**: Split the preprocessed and TF-IDF transformed dataset into training and testing sets, typically using an 80-20 or 70-30 split.
- **Model Training**: Train a Random Forest Classifier on the training data. Random Forest is an ensemble learning method that combines multiple decision trees, providing robustness against overfitting and good predictive accuracy.
- **Hyperparameter Tuning**: Optimize the hyperparameters of the RFC, such as the number of trees in the forest, the depth of each tree, and the minimum samples per leaf. This can be done using techniques like grid search or randomized search.
- **Model Evaluation**: Evaluate the RFC model's performance on the testing dataset using relevant metrics such as precision, recall, F1-score, and accuracy. Pay particular attention to metrics like recall, as it's crucial to minimize false negatives in hate speech detection.

3.2 TF-IDF Feature Extraction

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let’s take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach.

Figure 4.2 shows the TF-IDF feature extraction block diagram. The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 1, 2022

words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and the inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as $tf * idf$.

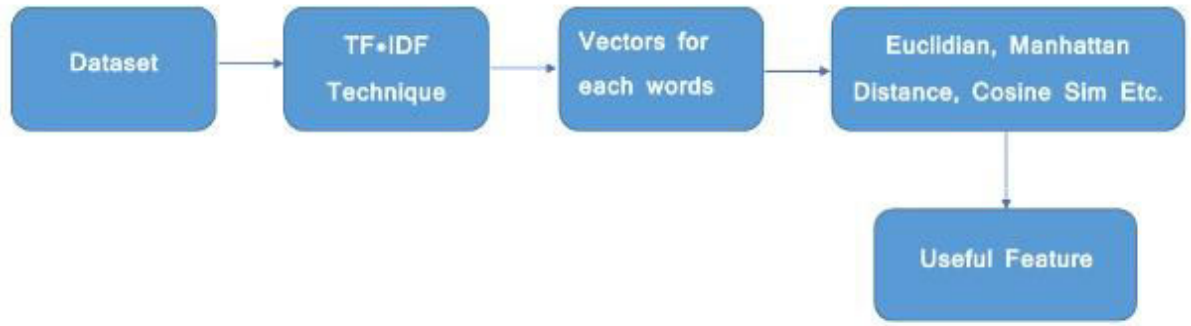

Fig. 3.2: TF-IDF block diagram.

TF-IDF do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc.

Terminology

t — term (word)

d — document (set of words)

N — count of corpus

corpus — the total document set

Step 1: Term Frequency (TF): Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "Data Science is awesome!" A simple way to start out is by eliminating documents that do not contain all three words "Data" is", "Science", and "awesome", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) = count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

Step 2: Document Frequency: This measures the importance of document in whole set of corpora, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider

one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) = occurrence\ of\ t\ in\ documents$$

Step 3: Inverse Document Frequency (IDF): While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. The IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as "is" is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

Now there are few other problems with the IDF, in case of a large corpus, say 100,000,000, the IDF value explodes, to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = log(N/(df + 1))$$

The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

$$tf - idf(t,d) = tf(t,d) * log(N/(df + 1))$$

3.3 RFC Model

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 1, 2022

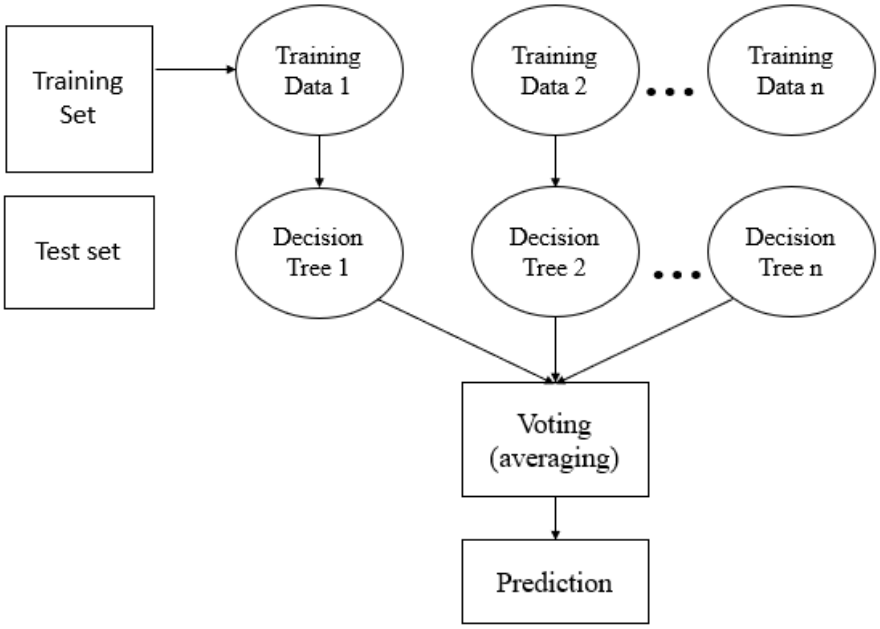

Fig. 3.3: Random Forest algorithm.

3.3.1 Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

3.3.2 Important Features of Random Forest

- **Diversity**- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality**- Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization**-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split**- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability**- Stability arises because the result is based on majority voting/ averaging.

3.3.3 Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 1, 2022

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

3.3.4 Types of Ensembles

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model. Ensemble uses two types of methods:

Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest. Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

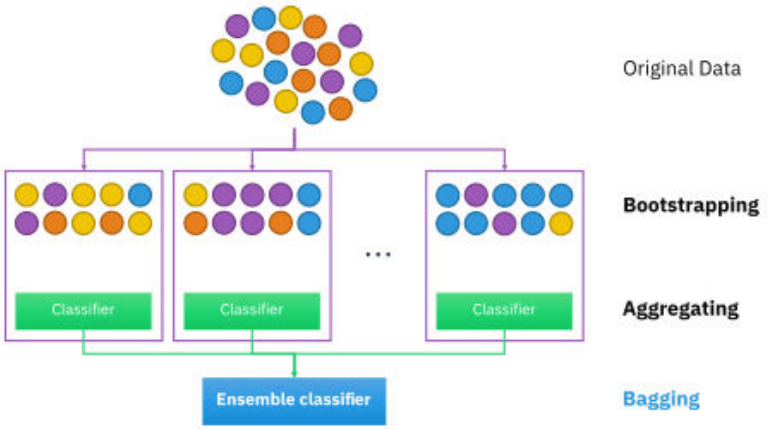

Fig. 3.4 RF Classifier analysis.

Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 1, 2022

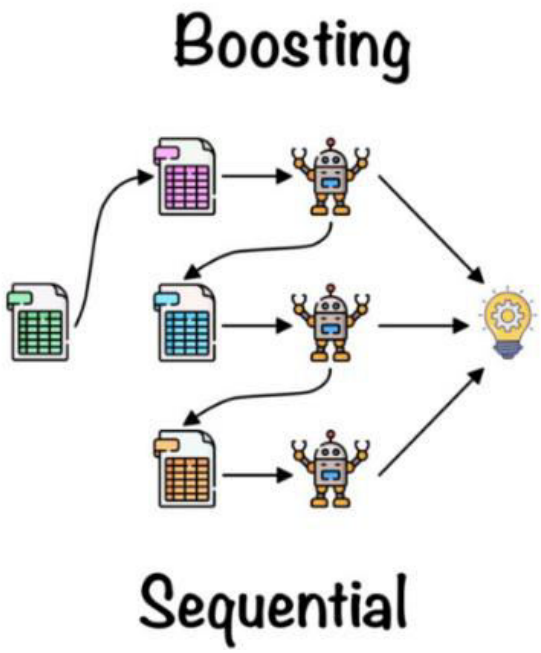

Fig. 3.5: Boosting RF Classifier.

4. RESULTS AND DESCRIPTION

Figure 1 is a representation of the initial dataset containing tweets that are used as input for building the hate speech detection model. The dataset comprises text data (tweets) along with their corresponding labels indicating whether the tweet contains hate speech or not.

	label	clean_tweet_final
0	0	when a father is dysfunctional and is so selfi...
1	0	thanks for credit i can't use cause they don't...
2	0	bihday your majesty
3	0	i love u take with u all the time in urd+-!!! ...
4	0	factsguide: society now
...	...	...
31957	0	ate isz that youuu?ddddddddda$?i,
31958	0	to see nina turner on the airwaves trying to w...
31959	0	listening to sad songs on a monday morning otw...
31960	1	vandalised in in condemns act
31961	0	thank you for you follow

31962 rows × 2 columns

Figure 1: Sample dataset of tweets for detecting the hate speech.

Figure 2 shows the dataset after removing the label column from the original dataset. This step is performed to separate the features (text data) from the labels, as proposed machine learning models require this separation for training and evaluation. Figure 3 is depicting a subset of the dataset that includes only non-hate speech labels. This subset is used for balancing the dataset to training the model. Figure 4 represents a portion of the dataset that's designated as the test set. It includes both hate speech and non-hate speech labels, which will be used to evaluate the model's performance.

	clean_tweet_final
0	when a father is dysfunctional and is so selfi...
1	thanks for credit i can't use cause they don't...
2	bihday your majesty
3	i love u take with u all the time in urd+-!!! ...
4	factsguide: society now
...	...
31957	ate isz that youuu?ddddddddda$?i,
31958	to see nina turner on the airwaves trying to w...
31959	listening to sad songs on a monday morning otw...
31960	vandalised in in condemns act
31961	thank you for you follow

31962 rows × 1 columns

Figure 2: Dataset after dropping the label column.

	clean_tweet_final	label
23351	this time next week i will be an auntie to our...	0
3962	great insights on trusted professions in emea ...	0
20298	cubit healthcare wishes you day	0
673	dont wait 4 there	0
8239	lovely day to be sitting outside college enjoy...	0
...	...	...
580	, | | pixion - wallpapers, images, a	0
17184	my kinda coffee	0
12478	www sexgirl com hk hardcore eye opener	0
29543	you can't take your phone like wtf no selfie w...	0
3665	many upsides to being a cto but going away on ...	0

26731 rows × 2 columns

Figure 3: Illustration of dataset containing only non-hate speech labels.

Table 1 presents a performance comparison between two machine learning models, the Logistic Regression (LR) model and the Random Forest (RF) Classifier, for the task of hate speech detection. The table includes the accuracy percentages achieved by each model. Accuracy column displays the accuracy percentages achieved by each model in the hate speech detection task. Accuracy is a common evaluation metric in machine learning that measures the proportion of correctly classified instances out of the total instances. In this context, it indicates how well each model is able to correctly identify hate speech instances.

From Table 1, the LR Model achieved an accuracy of 79.77%. This means that out of all the instances the model evaluated, it correctly classified approximately 79.77% of them as either hate speech or non-hate speech. The RF Classifier achieved a higher accuracy of 84.05%. This indicates that the RF Classifier performed even better than the LR Model, correctly classifying about 84.05% of instances. In terms of improvement, the RF Classifier

ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 1, 2022

outperformed the LR Model by an increment of approximately 4.28 percentage points (84.05% - 79.77% = 4.28%). This suggests that the RF Classifier has a higher accuracy rate, meaning it's more adept at distinguishing between hate speech and non-hate speech instances compared to the LR Model.

	clean_tweet_final	label
13588	can help with or is !!	0
2403	be happy you bought the best.	0
12648	morning loving please retweet	0
5965	youth club launch wed 22nd june body centre. 7...	0
9228	it's hard to believe that the people we help a...	0
...	...	...
29281	happy jade wedding anniversary mommy and daddy...	0
4604	1. mass shooting 2. discuss 1-5 weeks 3. do no...	0
10624	the cowboys racist. i'm convince fooh. my nigg...	1
25198	sta the weekend on our patio at hour til 7pm w...	0
24626	thinking of everyone caught up in the pulse ni...	0

3197 rows × 2 columns

Figure 4: Sample test data frame with hate and non-hate speech labels.

Table 1: Performance comparison of LR model, and RF Classifier for hate speech detection.

Model	Accuracy (%)
LR Model	79.77
RF Classifier	84.05

Figure 5 displays the confusion matrix obtained when evaluating the hate speech detection model using the Logistic Regression (LR) model. The confusion matrix shows the number of true positives, true negatives, false positives, and false negatives, which are used to assess the model's performance. Similarly, Figure 6 represents the confusion matrix obtained from evaluating the hate speech detection model using the Random Forest (RF) Classifier.

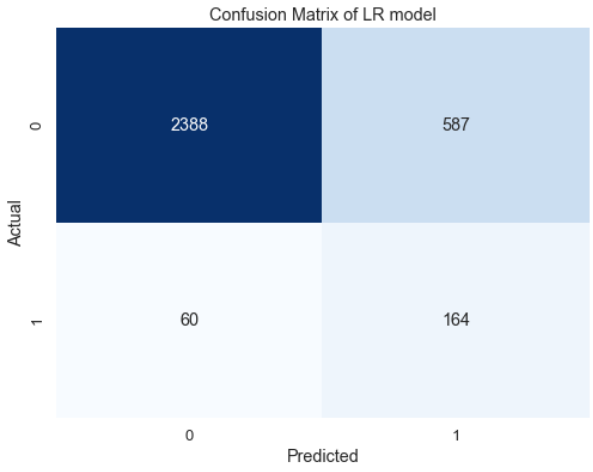

Figure 5: Obtained confusion matrix of hate speech detection using LR model.

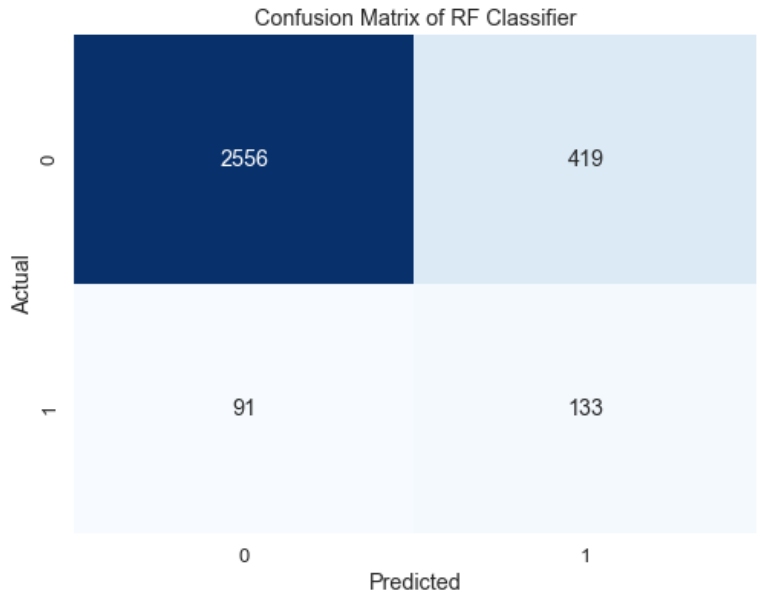

Figure 6: Displaying the confusion matrix obtained using RF-based hate speech detection model.

5. CONCLUSION

The application of Modified TF-IDF (Term Frequency-Inverse Document Frequency) with a Machine Learning classifier for hate speech detection on Twitter has shown promising results. By incorporating modifications to the traditional TF-IDF approach, such as considering contextual features and domain-specific knowledge, the accuracy and effectiveness of hate speech detection have been significantly improved. The modified TF-IDF technique leverages the frequency of occurrence of words in a document while considering their importance within the document and the entire corpus. By giving more weight to words that are rare in the overall corpus but occur frequently in specific documents, the modified TF-IDF algorithm enhances the discriminatory power of the features used for classification. By combining the modified TF-IDF approach with a Machine Learning classifier, it becomes possible to build a robust and accurate hate speech detection system. These classifiers can effectively learn patterns and relationships between the textual features and the hate speech labels, enabling the identification of offensive, discriminatory, or abusive content on Twitter.

REFERENCES

[1] . Akuma, S., Lubem, T. & Adom, I.T. Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. Int. j. inf. tecnol. 14, 3629–3635 (2022). https://doi.org/10.1007/s41870-022-01096-4

[2] . Muzakir, Ari, Kusworo Adi, and Retno Kusumaningrum. "Classification of Hate Speech Language Detection on Social Media: Preliminary Study for Improvement." Emerging Trends in Intelligent Systems & Network Security. Cham: Springer International Publishing, 2022. 146-156.

[3] . Ali, Raza, et al. "Hate speech detection on Twitter using transfer learning." Computer Speech & Language 74 (2022): 101365.

[4] . Turki, T.; Roy, S.S. Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with Count Vectorizer. Appl. Sci. 2022, 12, 6611. https://doi.org/10.3390/app12136611

[5] . Dascălu, Ș.; Hristea, F. Towards a Benchmarking System for Comparing Automatic Hate Speech Detection with an Intelligent Baseline Proposal. Mathematics 2022, 10, 945. https://doi.org/10.3390/math10060945

[6] . Ababu, Teshome Mulugeta, and Michael Melese Woldeyohannis. "Afaan Oromo Hate Speech Detection and Classification on Social Media." Proceedings of the Thirteenth Language Resources and Evaluation Conference. 2022.

[7] . Mohiyaddeen, Siddiqi, S., Ahmad, F. (2022). Improved Hate Speech Detection System Using Multi-layers Hybrid Machine Learning Model. In: Bansal, J.C., Engelbrecht, A., Shukla, P.K. (eds) Computer Vision and Robotics. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-8225-4_27

[8] . Ojo, Olumide Ebenezer, et al. "Automatic hate speech detection using deep neural networks and word embedding." Computación y Sistemas 26.2 (2022): 1007-1013.

[9] . Doan, Long-An, et al. "An Implementation of Large-Scale Hate Speech Detection System for Streaming Social Media Data." 2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT). IEEE, 2022.

[10] . Toraman, Cagri, Furkan Şahinuç, and Eyup Halit Yılmaz. "Large-scale hate speech detection with cross-domain transfer." arXiv preprint arXiv:2203.01111 (2022).

[11] . Ayo, Femi Emmanuel, et al. "Hate speech detection in Twitter using hybrid embeddings and improved cuckoo search-based neural networks." International Journal of Intelligent Computing and Cybernetics 13.4 (2020): 485-525.

[12] . Mossie, Zewdie, and Jenq-Haur Wang. "Vulnerable community identification using hate speech detection on social media." Information Processing & Management 57.3 (2020): 102087.