Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7699001

SOCIAL MEDIA HATE SPEECH DETECTION USING NLP AND DEEP LEARNING TECHNIQUE

Professor Loveleen Kaur Pabla[1], Dr. Prashant Kumar Jain[2] and Dr. Prabhat Patel[2]
[1]Department of Computer Science & Engineering, RGPV, Bhopal (M.P.) -462033, India
[2]Electronic Department, RGPV, Bhopal (M.P.) -462033, India
Corresponding Author E-mail: loveleenkaurpabla@gmail.com

Abstract: In a number of countries worldwide in normal communication people use offensive languages in reality both online and offline. But, all the abusive conversation between two parties is hate speech, it is the subject of investigation. Therefore, in this paper, the key area of study is the differentiation between hate speech and offensive language. The paper includes three parts of the work: the first study of the recent development in classifying hate speech in social media, the Second, proposed an algorithm for classifying hate speech text from normal and offensive language text, and the third provides an algorithm to identify the source of hate spreader. Therefore, first, a review of recent literature has been carried out which is divided into the review and surveys, hate speech classification as binary classification, and hate speech detection as a multi-class classification problem. Then a model for hate speech classification has been proposed, which includes the data pre-processing, natural language processing (NLP), and Term Frequency-Inverse Document Frequency (TF-IDF) based feature extraction. The features are used to train a 2D-Convolutional Neural Network (CNN) and Support Vector Machine (SVM) model. Finally, an algorithm is proposed to identify the source of hate spreader. The dataset available on Kaggle for hate speech, offensive language, and normal text is used for experimental analysis. According to finding social media text only with the NLP features are not providing good accuracy. On the other hand, only TF-IDF-based features demonstrate higher accuracy as compared to NLP-based features. Additionally, a combination of both features is providing more accurate results as compared to individual techniques.

Index Terms: Hate speech detection, Offensive language, Text mining, Natural language processing, Deep Learning.

I. Introduction

India is a republican country, which combines different cultures, communities, languages, religions, and food. India is an example of social harmony and unity. But, due to the size and diversity of the country, there is always a risk of social or communal tension. Because using different social media anti-social and anti-nation elements are involved in spreading hate. But, social media is raising an individual's voice and also providing freedom of expression. Therefore, we need a technique to deal with hate speech on social media. However, in order to classify hate from the social media text, a number of techniques are available but most of the techniques are aimed to classify only hate speech text and normal text as binary classification problems. But in reality, hate speech is different from other kinds of offensive language, because in real-life practice a number of people use abusive language in normal communication. That makes the classification of hate speech classification more complex.

In this paper, we provide the contribution for successfully classifying hate speech more accurately in presence of other offensive language and normal text. Therefore, first, we performed a literature review of recent techniques. In order to understand the problem correctly the review and surveys related to ML and social media hate speech is studied. Next the available solutions based on binary classification and multi-class classification has also been studied. Further, we have investigated how keywords are influencing the performance of classifying social media text. Therefore, we proposed a hate speech classification technique according to the ML application life cycle. Then, an algorithm is designed to retrieve the source of the hate speech spreader. Finally, we compare the performance of the proposed social media hate speech text classification techniques and the future extension plan has been shared.

2. Related Study

Social media is a large platform and engages a large number of audiences of almost all age groups. That enables us to connect the users around the world. On such platforms, anyone can create an account, publish posts and thoughts, share content, provide opinions or reviews, and others. But there is a category of malicious users also available. These users are intentionally sharing nonsocial or anti-social content to distribute hate. Therefore, detection and monitoring of hate speech content are essential. In literature, a significant amount of work is available based on machine learning for social media hate speech detection. Thus the conducted review includes:

1. Understanding the social media hate speech
2. Solutions based on binary classification
3. Solutions based on multi-class classification

2.1. Survey, Review and Comparisons

In order to understand the problem of hate speech detection we first consider the articles based on reviews and surveys. The most relevant articles, which help us for understanding the issue of hate speech classification using the ML technique, are included. Among these articles, N. S. Mullah et al [2] review machine learning (ML) algorithms for hate speech detection as text classification. The components of hate speech using ML life cycle were discussed. Authors (1) equip readers with the steps of hate speech detection (2) Weaknesses and strengths of the method (3) research gaps and challenges. S. Abro et al [4] compare the performance of three feature engineering techniques and eight ML algorithms. The results showed that the bigram features with SVM provide the best 79% accuracy. K. J. Madukwe et al [7] provide an overview of hate speech. They discuss how the pre-processing and data format influence results. They compared the attributes of datasets, outlined limitations, and recommended approaches.

Table 1 Survey and Reviews

Ref.	Issues	Contribution	Method	Results
[2]	Review ML algorithms for	Examined baseline components of hate	Equip with steps involved in HS detection. Weaknesses	ML techniques were reviewed like classical ML,

	hate speech detection	speech using ML algorithms.	and strengths of methods.	ensemble and deep learning.
[4]	Study to compare feature engineering and ML for HS.	Compare performance of three feature engineering techniques and eight ML algorithms.	Practical implication and use. The comparisons will be used for future researches for text classification.	Showed bigram features with the SVM algorithm provide 79% accuracy.
[7]	An overview of issues pertaining to the data that debilitate in this area.	How pre-processing and data format result in dataset, compare difficult and unfair.	Comparing the attributes of datasets for HS detection, outlining limitations and suggest approaches.	Fill the gap and become one-stop shop for information of hate speech datasets.
[13]	HS is a complex, and detection has gained traction in NLP, as reviews.	Annotated corpora and benchmarks are key resources, consider supervised learning.	Analyze resources available by the community, development method, topical focus, language, and others.	Highlight a heterogeneous, growing landscape, marked issues for improvement.
[14]	Existing methods are limited by lack of comparative evaluations.	Combining Convolutional and LSTM, and evaluation of the method on public datasets.	Feature selection to understand impact of the features. Findings show the importance of feature.	Outperforms on 6 out of 7 datasets. Feature selection reduces 90% feature space.
[15]	Anti-social behavior like harassment, bullying, and HS.	HS instigators and target users classification. Dataset for various types of hate.	Properties of instigators and targets. Instigators target popular and high profile users for visibility.	Both have personality facets differ. Understanding online HS engagement.
[17]	Offensive language and hate detection is important for understand social issue.	Exiting work suffer from inaccurate representation.	Feature set using language and intergroup threat theory. Classification with embedding.	Validated on experiments: (a) comparing method. (b) Tested on unseen datasets.

[19]	Review on hate speech with different classes and terrorism of online social networks.	-	-	Combined effort of ISPs, government and social networks, can frame policies to counter HS and terrorism.
[22]	Study of HS from computer science point of view.	Organizes and describes an overview of earlier approach, including algorithms, methods, and features used.	Complexity of HS, defined in platforms and contexts, and definition. It has social impact, online communities and media.	Development and systematization of resources, like guidelines, multiple language, and algorithms.

F. Poletto et al [13] analyze resources available by community, development method, topical focus, language, and others. They highlight a heterogeneous, growing landscape, marked by several issues. Z. Zhang et al [14] introduce a method based on a deep neural network combining Convolutional and long short-term memory. The method outperforms on 6 out of 7 datasets. Automatic feature selection reduces feature space by over 90%. M. ElSherief et al [15] compare hate speech instigators and target users. Through a multi-step classification, and create a hate speech dataset. They find hate instigators target popular and high-profile users, and result in greater visibility.

W. Alorainy et al [17] propose a feature set that utilizes language to use around the concept and intergroup threat theory to identify subtleties. Implement a range of classification methods using embedding learning. N. Chetty et al [19] review hate speech with different classes and terrorism. With the combined effort from the government, Internet Service Providers, and social networks, policies can be framed to counter both hate speech and terrorism. P. Fortuna et al [22] organizes and describes the overview of previous approaches, algorithms, methods, and features used. They also discuss the complexity of the platforms and contexts. The development and systematization of resources, such as guidelines, annotated datasets, and algorithms, is crucial.

2.2.Binary Classification

Next, we consider some articles where the hate speech classification problem is considered a binary classification problem. F. Balouchzahi et al [3] present Voting Classifier (VC) to Hate Speech Spreader Detection. They include profiling of HS spreaders. This task is modeled as a binary classification. The models utilize a combination of char and word n-grams as features. The features are used with, Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF). K. Miok et al [8] propose a Bayesian method using Monte Carlo dropout in the attention layers to provide reliable estimates. They test whether dimensions can enhance the information by the BERT model. P. Badjatiya et al [11] aim at bias mitigation from unstructured text data. First, design methods to quantify the bias and an

algorithm for identifying the set of words. Second, proposed a method based on generalizations for bias-free learning. It encodes knowledge and reduces imbalance. The experiments with the Wikipedia Talk Pages dataset and Twitter dataset were done. E. Ombui et al [12] address the identification of hate speech in code switched text. The words in different languages in a message are common among multilingual communities. They explored the performance of features across various ML algorithms and TF-IDF. Using a dataset of 25k tweets, using SVM provides better performance.

Table 2 Binary Classification

Ref.	Issues	Contribution	Method	Results
[3]	Identifying HS is temporary solution, so Develop system to detect content polluters.	HS Spreader Detection by profiling for two languages, English and Spanish.	Combination of char and n-grams as features. Use ML classifiers, SVM, LR, and RF with voting.	Models with accuracies of 73% and 83% for English and Spanish languages
[8]	Hate speech is an important problem in the management of user-generated content.	Deep neural networks, such as BERT, show good performance. So far, these methods have not been able to quantify their output reliability.	Bayesian method using Monte Carlo dropout in attention layers to provide reliable estimates. Evaluate and visualize results on hate speech detection.	Test whether it can enhance BERT in HS classification. The experiments show that Monte Carlo dropout provides a viable mechanism.
[11]	Critical to design abuse detection. Existing methods using stereotype words and suffer from biased training.	Quantify bias and algorithms for identifying set of words. Leveraging generalizations for bias-free learning.	Encode knowledge to generalizes content for classifier, and reduce imbalance. Multi generalization policies and analyze performance.	Two datasets, Wikipedia Talk Pages and Twitter dataset, used to show generalizations results in better performance.
[12]	Words in different languages are a common occurrence.	Performance of different features with ML algorithms and character level TF-IDF	SVM as compared to six other conventional and two deep learning algorithms.	Performed best given a code switched dataset of 25k annotated tweets

2.3.Multi-class classification

A key challenge for hate-speech detection is the separation of hate speech from offensive language. Lexical methods have failed because it classifies messages containing specific terms. T. Davidson et al [1] used a lexicon to collect tweets containing hate speech. The label of the sample is hate speech, offensive language, and neither. A multi-class classifier is used. They show when we can reliably separate hate speech. P. K. Roy et al [5] deal with hate speech and focuses on various aspects, like gender, religion, race, and disability. A system is developed using the Deep Convolutional Neural Network, which utilizes text with a GloVe embedding vector. Md. R. Karim et al [6] propose hate speech detection in the Bengali language, called DeepHateExplainer. Texts are pre-processed, to classify using a neural ensemble method. Important terms are identified using sensitivity and layer-wise relevance propagation (LRP), before explanations. H. Watanabe et al [9] detect hate expressions based on unigrams and patterns. The experiments show that the approach reaches accuracy equal to 87.4%.

P. Chiril et al [10] propose hate speech detection using annotated datasets, to investigate the problem of transferring knowledge. They contribute: (1) explore the ability of hate speech detection models; (2) models to detect topics and targets; and (3) study the impact of knowledge encoded. They show that: (1) training of topic-specific datasets is effective; (2) multi-task approach outperforms for hatefulness and topical focus; and (3) models incorporating EmoSenticNet emotions, SenticNet, and both emotions features. H. Liu et al [16] introduce a formulation of hate speech identification in multi-task learning through the fuzzy ensemble. A single-labeled data is used for semi-supervised multi-label learning. They report four types of hate speech, namely: religion, race, disability, and sexual, with a detection rate of 0.93. E. Pronoza et al [18] address the problems in the Russian language. This allows differentiating between attitudes. That comprise cases of toxic speech, the sample secures a realistic and, much higher proportion of negativity. Experiment with ML models, such as SVM, deep learning, BERT, and interpret predictions. The results are achieved by RuBERT with linguistic features and are acceptable on the two-class problem, and three-class detection. B. Pelzer et al [20] developed a method to measure hate directed at politicians using a combination of NLP and reasoning. They tested the method in a study that analyze hate directed at six Swedish politicians. Y. Zhou et al [21] focus on ML methods for text classification such as Embeddings from Language Models (ELMo), Bidirectional Encoder Representation from Transformers (BERT), and Convolutional Neural Network (CNN). Then adopt fusion strategies to combine the classifiers to improve classification performance.

Table 3 Multi-Class Classification For Hate Speech Recognition

Ref.	Issues	Contribution	Method	Results
[1]	Classify HS from offensive language	HS lexicon to collect tweets. Label samples into hate speech, offensive, and neither.	Train a multi-class classifier. Shows when we can separate HS and when it is difficult.	Homophobic and racist are classified as HS, sexist as offensive.

[5]	Cyberbullying, hate speech, and other.	Deals with the HS. Manually filtering traffic is impossible.	Automated system is developed using DCNN. Utilises tweet text with GloVe embedding vector.	Achieved the precision (0.97), recall (0.88) and F1-score (0.92).
[6]	Some languages are under-resourced, like Bengali, for accurate NLP.	Approach for HS detection from Bengali language, which called DeepHateExplainer.	Texts are pre-processed, classify using an ensemble. Terms are identified and explanations.	F1-scores of political 78%, personal 91%, geo-political 89%, and 84% for religious
[9]	Social networks forbid use of HS, due to size of networks	Propose an approach to detect hate expressions.	Based on unigrams and patterns of training set used, as features to train a ML algorithm.	Reaches accuracy 87.4% on offensive or not, and 78.4% on hateful.
[10]	Most of the HS detection approaches cast problem into classification without addressing topics or target.	Leverage annotated datasets, to investigate problem of knowledge transfer from different datasets and topics.	(1) HS detection from topic-generic datasets; (2) Models to detect topics and targets; (3) impact of encoded knowledge.	Topic-specific datasets is effective; multi-task performs better; features based Hurtlex, results best.
[16]	Instance can be assigned multiple labels but training is taken as single-task learning.	Multi-task learning using a multi-labelled data. Data Transformation result class imbalance.	Formulation of HS type identification. Data used with semi-supervised multi-label learning.	Types of HS, religion, race, disability and sexual. Result show fuzzy ensemble outperforms.
[18]	HS detection has issues of: reliable mark-up, informal and indirect way to express negativity, users' attitudes and, use	HS in Russian-language classifying between attitudes in text. Use a dataset of messages of 12K instances with annotation.	Previous dataset comprise cases of toxic speech; Sample secures a realistic and, higher proportion of subtle negativity. Used SVM, deep learning, BERT, and interpret.	Results are achieved by RuBERT with linguistic features, F1-hate = 0.760, on two-class, and F1-hate = 0.813, on three-class.

	of other languages.			
[20]	Detecting hate is a difficult task since hate can be expressed in many different ways.	A method to measure hate directed at politicians using a combination of NLP and automated reasoning.	The method is adapted to work on Swedish, it is language independent.	Method analyze hate directed at six Swedish politicians. Show method has a high precision but a low recall.
[21]	Literature on HS. Performances of methods, advantages and disadvantages.	Way to improve the results of classification by fusing the various classifiers results is a meaningful attempt.	Methods such as ELMo, BERT and CNN, and to data sets of SemEval 2019. Adopt fusion to combine the classifiers.	The results show that the accuracy and F1-score of the classification are significantly improved.

3. Proposed Work

According to the collected and studied recent literature around the development of hate speech recognition, we have found the classification of hate speech in presence of other offensive language is a complex task. Additionally, the methods developed using machine learning technologies are also not much accurate. Additionally, mostly the problem of hate speech detection has been formulated as binary and multi-class classification problems. Among them, the multi-class classification techniques are more realistic and practical in real-world problems. Thus in this paper, we have also considered this problem as a multi-class classification problem. This section focuses on these two objectives:

(1) Proposed an algorithm for classifying hate speech text from normal and offensive language text

(2) An algorithm to identify the source of hate spreader.

In this context, first design an algorithm for classifying the text into hate speech, offensive language, and normal. The aim is to apply natural language processing (NLP), keyword-based features, and the combined features into classifying the social media text. Then, the prepared features are utilized with two popular supervised learning algorithms namely Support Vector Machine (SVM), and Convolutional Neural Network (CNN). Second, we have developed an algorithm for identifying the source of hate speech spreader based on twit structural analysis. The details of both techniques are provided in this section.

3.1. Model for classifying social media text

The proposed model for classifying hate speech text from offensive and normal text is demonstrated in fig. 1. For experimentation, we need a dataset, which contains hate speech, offensive language, and normal text, such kind of dataset is available on Kaggle [23]. The

dataset consists of a total of 24783 instances of twits, which is subdivided into two sets of samples train and test. A total of 18587 samples are used for training and the remaining 6196 samples are used for testing. The obtained dataset is demonstrated in fig. 2.

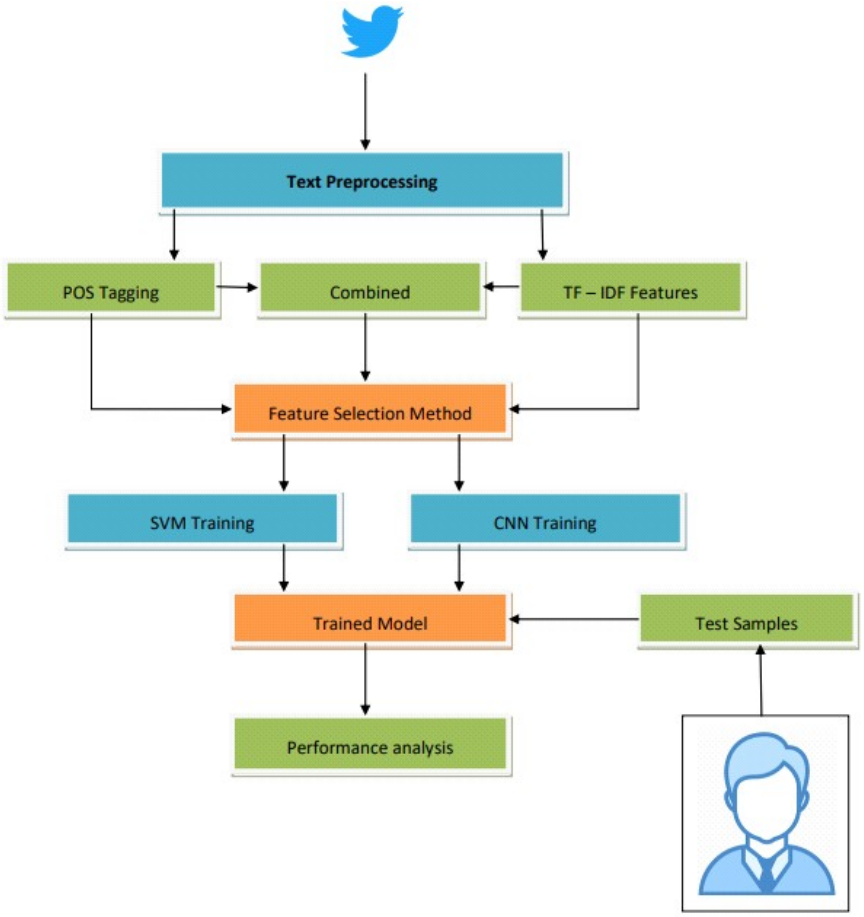


Fig. 1. Proposed Model for feature learning

The dataset contains a total of 7 attributes, among them we considered only 'tweet' and 'class' attributes. After separating the required attributes from the given dataset the pre-processing techniques are involved for cleansing the raw data. In this work the text data is being used for a classification task thus the available text pre-processing techniques are involved.

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	tweet
0	0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't...
1	1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2	2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3	3	3	0	2	1	1	!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4	4	6	0	6	0	1	!!!!!!!!!!! RT @ShenikaRoberts: The shit you...

Fig. 2: Initial Dataset format of Kaggle

These techniques are filtering stop words, removing special characters and specific character sequences. After these steps, we are converting text to lower case, tokenize the text, and finally use lemmatization. After pre-processing entire samples, we have performed feature selection. There are three variants of features involved:

A. **Part of Speech (POS) Tagging:** POS is a category of words that are used to describe the structure of a sentence. An example of a POS-tagged feature is demonstrated in fig. 3. To understand this we can take an example sentence:

"Fruit flies like a banana"

The sentence has been parsed and represented as a tree structure like fig. 3. In order to parse the sentence, we have used the Natural Language Toolkit (NLTK) library of python for implementation. After applying the POS tagging we have recovered 37 tags for each tweet in the dataset.

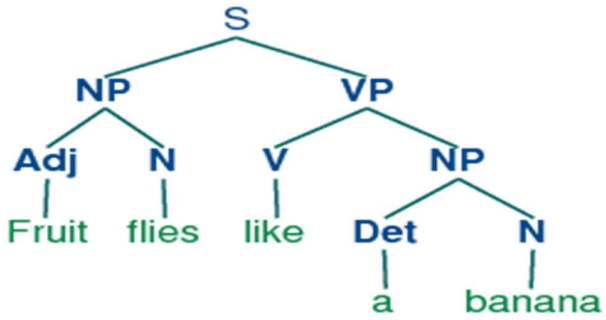


Fig. 3. POS Tagging Example [24]

B. **Term Frequency and Inverted Document Frequency (TF-IDF):** it is one of the popular text feature selection techniques in text mining. It is used for identifying essential keywords as a feature, which represents the entire domain knowledge. The TF-IDF can be calculated using the equation (1), (2), and (3).

$$TF = \frac{total\ times\ a\ keyword\ appeared}{total\ words} \ldots\ldots\ldots (1)$$

And

$$IDF = log\frac{N}{|\{d \in D : t \in d\}|} \ldots\ldots\ldots (2)$$

Where, N is the number of documents, and $\{d \in D : t \in d\}$ is the number of documents where the term t appeared.

Further the weights are computed as:

$$W = TF * IDF \ldots\ldots\ldots\ldots (3)$$

The weights are used for selecting the essential keywords for utilizing with the classifier. A total of 5000 keywords are used for feature vector representation.

C. **Combined Features:** The obtained features from both the techniques i.e. POS tagging and TF-IDF has been combined to prepare a new dataset. The aim is to take advantage of both features. This feature vector contains a total of 5037 attributes.

In order to experiment with the implemented feature extraction techniques, a provision has been implemented to select a specific feature selection approach. The extracted features are used with two different classification algorithms namely Support Vector Machine (SVM) and Convolutional Neural Network (CNN).

SVM Classifier: The SVM is one of the classical classifiers, which is used in various kinds of applications. Additionally have competitive performance with the neural network. The SVM classifier is mainly creating a boundary condition to distinguish between two classes of samples. These boundaries are known as the hyper-planes. It maximizes the chances to classify the data samples more accurately. Here, we have used a simple linear version of the SVM classifier.

CNN classifier: CNN is a variant of Artificial Neural Network (ANN), and has the great potential to deal with a large amount of data. Therefore it is used for large dataset processing. Initially, it is designed for image data, but now it is being used with different data formats. It supports various different kinds of architectures to best fit the problem to find a solution. However, in this experiment, we have used a simple or basic architecture of CNN. The used CNN has four main layers such that:

1. Input layer is a dense layer, which consists of a similar number of neurons as the input feature set size. This layer includes the "ReLu" activation function.
2. The Second layer has 128 neurons and is configured with the "ReLu" activation function.
3. Third Layer contains 64 Neurons and used the "ReLu" activation function.
4. Final layer is an output layer and contains the three neurons as the class labels. It is developed with the "SoftMax" activation function.

Both the implemented machine learning algorithms are trained on the prepared feature vectors. After training the algorithms, the trained algorithms are used for identifying the class labels of text in terms of normal, offensive, and hate speech. Therefore a user can also provide the input for validating the trained model. Thus, by using real samples or prepared test samples can be utilized to classify them. Further, the performances of the models have been measured using different performance parameters such as accuracy, loss, and training time.

3.2. Tracking the Source of Hate

Spreading hate on social media is a criminal offense. Therefore we need to identify the hate spreaders accurately. In this context, the proposed model is extended to locate the source of hate speech tweets. This process is initiated when the classifier classifies a tweet as a hate speech text. The aim of this algorithm is to find out the victim or target of hate i.e. person or a community. Therefore, we first define the scope of source and target.

1. **Source:** The term source is used for describing a Twitter user, who initiated the twit. It is necessary for identifying the source of hate speech.
2. **Target:** The term target is used to denote an individual or community who is the victim of the tweet.

The proposed algorithm for finding source and target, the algorithm is using POS tagging. Here we are considering a limited number of POS attributes. These NLP tags are:

1. Noun

2. Pronoun
3. Verb
4. Conjunction
5. Ad verb
6. Adjective
7. Negation

Additionally, we have considered the sentence formation as demonstrated in table 3 for finding the source and target.

Table 3 Sentence structure

Source	To	Destination
↑	↑	↑
Split 1	Conjunction	Split 2

However, always a sentence is not similar to what someone is saying about someone, because in the passive voice the sentence structure is just opposite to this structure. In order to find whether a sentence is active or passive, we use the method described in [25]. Using the given set of rules first we identify whether a sentence is active or passive. Then tweet is tagged using POS tagger. After tagging the tweet has split into two parts i.e. Split 1 and Split 2. Both the splits are further used to locate noun and noun phrases. Finally, we follow the following two rules to describe the source and target of the tweet:

- If Sentence = = Active Then
    - Split 1 Noun or Noun phrase is Source of hate
    - Split 2 Noun or Noun phrase is Target of hate
- Else if Sentence = = Passive Then
    - Split 1 Noun or Noun phrase is Target of hate
    - Split 2 Noun or Noun phrase is Source of hate

The process of identifying the hate speech source and target is demonstrated in fig. 4.

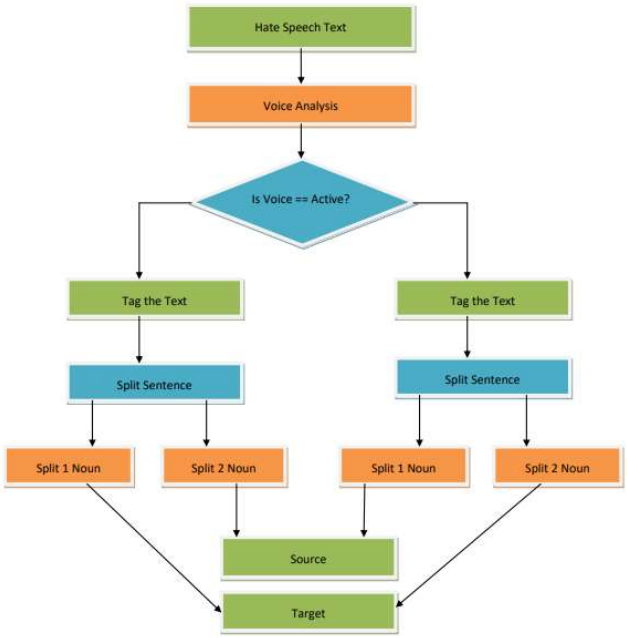


Fig. 4. Flow chart for identifying hate speech victim

4. Results Analysis

This section provides the experimental evaluation of the implemented models for identifying hate speech. The experiments are conducted in the following scenarios:

1. Evaluation of the influence of the feature extraction techniques for classifying hate speech, offensive language, and normal text.
2. Comparing the performance of SVM and CNN model for classification
3. Evaluation of performance for discovering twit source and target of hate speech

Scenario 1: In this experiment, we involve the three feature sets based on POS Tagging, TF-IDF, and a combination of both with the CNN and SVM classifiers. The aim of this experiment is to estimate which kind of features effectively work for classifying hate speech from other offensive text or normal text. Thus, the first performance of all three types of feature sets with the CNN classifier is given in fig. 5. Fig. 5 consists of four line graphs to demonstrate the performance of the CNN classifier with three different feature sets. Fig. 5(A) demonstrates the performance of the CNN classifier in terms of training accuracy. According to the training accuracy, we can see the feature based on POS is providing very less performance as compared to the TF-IDF and combined features. On the other hand, the performance of the model during validation, which is demonstrated in fig. 5(B), is demonstrating the effective performance of the combined feature as compared to the other two features. Similarly, performance in terms of training and validation loss is given in Fig. 5(C) and 5(D). The loss demonstrates how effectively we are approaching the required solution. According to the measured loss for both i.e. training and validation, we found the combined features are performing better than the other two individual features.

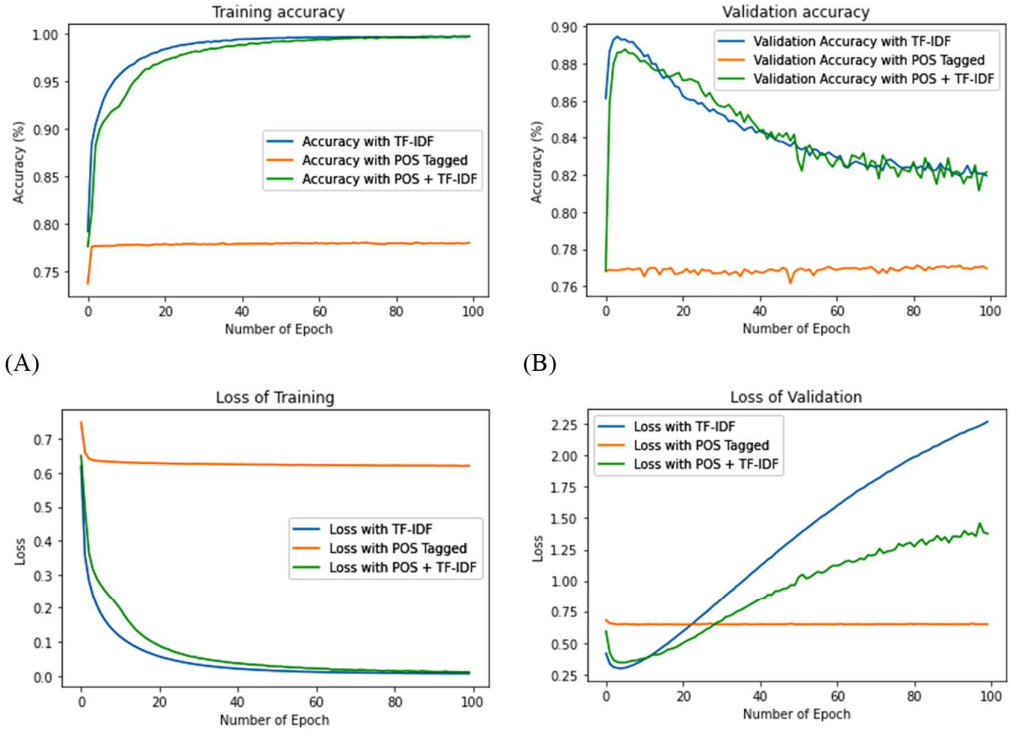


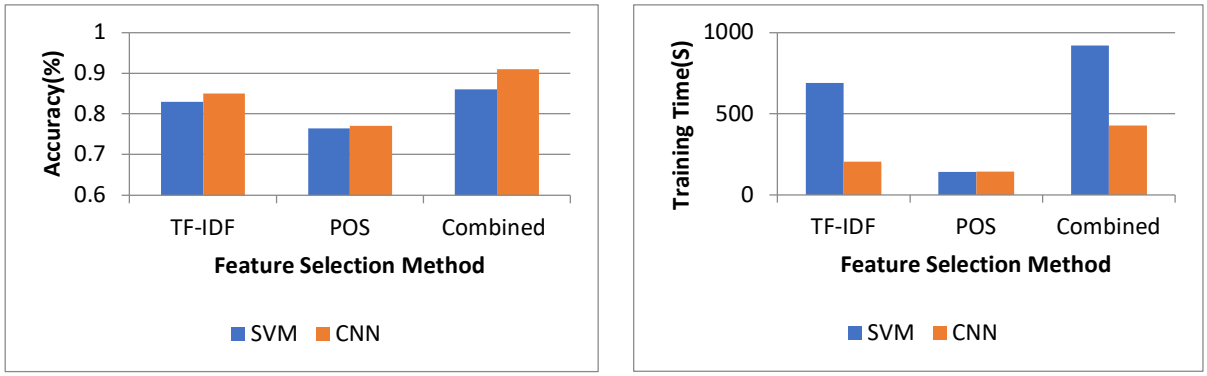
Fig. 5. Performance analysis of 2D-CNN for classifying the different feature set based on POS Tags, TF-IDF and combination of both in terms of (A) Training Accuracy (B) Validation Accuracy (C) Training Loss (D) Validation Loss

On the other hand, we also measure the performance of the features with the SVM classifier. In this context, we have measured the accuracy of the SVM with extracted features, as demonstrated in fig. 6. In fig. 6 we demonstrate the performance of the SVM with the extracted features. In this diagram, the accuracy is given on the Y axis in terms of percentage. According to the results, the combined features are providing high accurate classification as compared to individual TF-IDF and POS features.

Fig. 6. Classification accuracy of features using SVM

Scenario 2: In this experiment, we compare the performance of both the classifier i.e. SVM and CNN. The comparative performance in terms of accuracy and Training Time has been

measured and demonstrated in fig. 7. Fig. 7(A) demonstrates the accuracy of the SVM and CNN classifier for three feature extraction techniques. According to the obtained results, CNN has providing a more accurate classification as compared to SVM. Additionally, we have also provided a comparison of the time consumption for training. Fig. 7(B) demonstrates the training time of both algorithms for three feature extraction techniques. According to the training time, we found the SVM will perform well with a small size of data but when the dimensions of data have increased the SVM takes a significant amount of time for training. Additionally, the training time is also increased for CNN but it is acceptable as compared to the SVM classifier.



(A) (B)

Fig. 7. Comparison of SVM and CNN performance for classifying the three type of social media text features in terms of (A) accuracy in percentage (%) and (B) Training Time in seconds (S)

Scenario 3: Finally we have evaluated the performance of the proposed algorithm for identifying hate speech source and target (Victim). In order to demonstrate this phenomenon, we prepared three sets of labeled dataset samples, which are identified as hate speech. This set of data has the size of 100, 200, and 350 instances. The accuracy of the proposed algorithm for identifying the source and target of hate speech is demonstrated in fig. 8. The accuracy of the source and target identification technique has been found acceptable but requires more improvement in the near future.

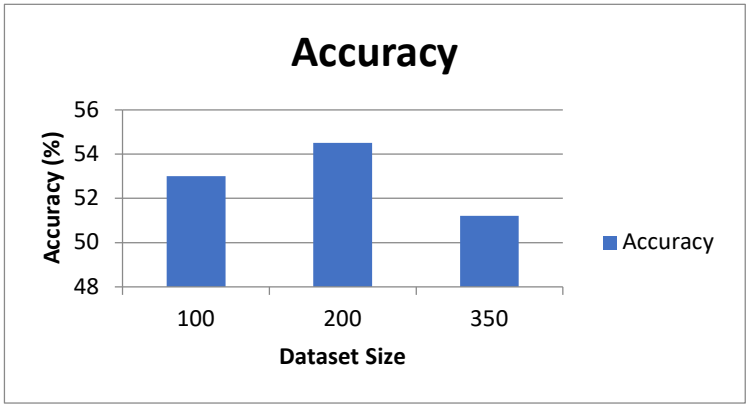


Fig. 8 Accuracy of hate speech source and target identification

5. Conclusion and Future Work

The proposed work is intended to study the classification of hate speech text from offensive and normal text. This task is much more complicated as compared to classifying the text into their sentiment classes. Therefore, this paper describes the following investigational consequences:

1. Influence of text feature selection techniques for hate speech classification: In this experiment, we have utilized three sets of features, first contains the features based on POS tags. Second, considered the use of TF-IDF-based features. The third feature set combines both features. The experimental analysis demonstrates the following:
    a. For identification of hate speech and offensive text from normal text is easier than the classification of hate speech from offensive and normal text.
    b. Both i.e. hate speech and offensive language contain various common keywords and structures which make this issue of classification more complex.
    c. Features which contain the properties of keywords and sentence structure can provide accurate results.
    d. The computational cost of the combined feature for feature extraction and training is higher as compared to using individual features but provides higher accuracy.
2. Performance analysis of classical and deep learning approach: in this experiment, we found both the models (SVM and CNN) work faster when we use less dimensional data. But when the data size is increased then deep learning techniques are providing better yield as compared to classical machine learning techniques in terms of both accuracy and time.
3. Investigation of hate speech source and target: For this purpose, we have designed an algorithm, which identifies the hate speech spreader and victim at the sentence level. However, we are getting an average of 50% of accurate identification of victims and spreaders.

During the experiments and the proposed system design, we have located some essential facts that may help to improve the proposed model. Therefore, in near future the following improvements are proposed:

1. We found that the TF-IDF-based features contain various keywords which are not much appropriate for representing the features. Therefore we need more refinement of features extracted.
2. We have found the deep learning techniques are better performing with a large set of data, thus in near future, we have explored more deep learning models that will provide more accurate classification.
3. The hate speech source and target identification technique offers limited classification accuracy and has limited scope. Thus, in near future, we are trying to improve the performance of identification in terms of accuracy and scope.

References

[1] T. Davidson, D. Warmsley, M. Macy, I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language", Proceedings of the Eleventh International AAAI Conference on Web and Social Media, 2017

[2] N. S. Mullah, W. M. N. W. Zainon, "Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review", IEEE Access, Vol. 9, 2021

[3] F. Balouchzahi, H. L. Shashirekha, G. Sidorov, "HSSD: Hate Speech Spreader Detection using N-grams and Voting Classifier", Conference and Labs of the Evaluation Forum, Bucharest, Romania, Sep. 2021

[4] S. Abro, S. Shaikh, Z. Ali, S. Khan, G. Mujtaba, "Automatic Hate Speech Detection using Machine Learning: A Comparative Study", International Journal of Advanced Computer Science and Applications, Vol. 11, No. 8, 2020

[5] P. K. Roy, A. K. Tripathy, T. K. Das, X. Z. Gao, "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network", IEEE access, Vol. 8, 2020

[6] Md. R. Karim, S. K. Dey, T. Islam, S. Sarker, M. H. Menon, K. Hossain, B. R. Chakravarthi, Md. A. Hossain, S. Decker, "DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language", arXiv:2012.14353v4 [cs.CL], Aug 2021

[7] K. J. Madukwe, X. Gao, B. Xue, "In Data We Trust: A Critical Analysis of Hate Speech Detection Datasets", Proceedings of the Fourth Workshop on Online Abuse and Harms, pages 150–161, Nov 2020.

[8] K. Miok, B. Škrlj, D. Zaharie, M. R. Šikonja, "To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection", Cognitive Computation, https://doi.org/10.1007/s12559-021-09826-9

[9] H. Watanabe, M. Bouazizi, T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection", IEEE, Vol. 6, 2018

[10] P. Chiril, E. W. Pamungkas, F. Benamara, V. Moriceau, V. Patti, "Emotionally Informed Hate Speech Detection: A Multi-target Perspective", Cognitive Computation, https://doi.org/10.1007/s12559-021-09862-5

[11] P. Badjatiya, M. Gupta, V. Varma, "Stereotypical Bias Removal for Hate Speech Detection Task using Knowledge-based Generalizations", ACM, San Francisco, CA, USA, IW3C2, 2019

[12] E. Ombui, L. Muchemi, P. Wagacha, "Hate Speech Detection in Code-switched Text Messages", 2019 IEEE

[13] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, V. Patti, "Resources and benchmark corpora for hate speech detection: a systematic review", Lang Resources & Evaluation, 55, 477–523, 2021

[14] Z. Zhang, D. Robinson, J. Tepper, "Hate Speech Detection Using a Convolution-LSTM Based Deep Neural Network", Lyon, France, ACM, WWW'2018

[15] M. ElSherief, S. Nilizadeh, D. Nguyen, G. Vigna, E. Belding, "Peer to Peer Hate: Hate Speech Instigators and Their Targets", Proceedings of the Twelfth International AAAI Conference on Web and Social Media, 2018

[16] H. Liu, P. Burnap, W. Alorainy, M. L. Williams, "Fuzzy Multi-task Learning for Hate Speech Type Identification", San Francisco, CA, USA, International World Wide Web Conference Committee, ACM , 2019

[17] W. Alorainy, P. Burnap, H. Liu, M. L. Williams, "'The Enemy Among Us': Detecting Cyber Hate Speech with Threats-based Othering Language Embeddings", ACM Transactions on the Web, Vol. 9, No. 4, Article 39, March 2018.

[18] E. Pronoza, P. Panicheva, O. Koltsova, P. Rosso, "Detecting ethnicity-targeted hate speech in Russian social media texts", Information Processing and Management, 58, 102674, 2021

[19] N. Chetty, S. Alathur, "Hate speech review in the context of online social networks", Aggression and Violent Behavior, 40, 108–118, 2018

[20] B. Pelzer, L. Kaati, N. Akrami, "Directed Digital Hate", 2018 IEEE

[21] Y. Zhou, Y. Yang, H. Liu, X. Liu, N. Savage, "Deep Learning Based Fusion Approach for Hate Speech Detection", IEEE Access, Vol. 8, 2020

[22] P. Fortuna, S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text", ACM Computing Surveys, Vol. 51, No. 4, Article 85, July 2018

[23] https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset

[24] http://www.armando.ws/2014/02/parts-of-speech-tagging-n-grams/

[25] https://gist.github.com/armsp/30c2c1e19a0f1660944303cf079f831a