Recognition of Hate Speech using Advanced Learning Model-based Multi-Layered Approach (MLA)

Puspendu Biswas[1], Donavalli Haritha[2]
Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India[1]
Professor and HOD, Dept. of Computer Science & Engg., Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India[2]

Abstract*—Hate speech becomes more complicated for the users of social media. Some users on online social networking sites (OSNS) create a lot of nonsense by uploading hate speech. OSNS applications developing many models to prevent this hate speech in terms of text and videos. However, these messages still need to be fixed for OSNS users. Sophisticated techniques must automatically identify and detect hate speech material to solve this problem. This paper proposes an advanced learning model-based Multi-Layered Approach (MLA) for hate speech recognition. The proposed model analyses textual data and finds hate speech patterns using multiple deep learning (DL) architectures. The algorithm can generalize well across settings and languages because it was trained on text datasets that include various hate speech types. The final step is an integrated model called Text Convolutional Neural Networks (TCNN), which combines hate text pattern detection with T-Convolutionals. Essential components of the model include the pre-trained model for DistilBERT, integrated pre-processing techniques like Text Cleaning, Lemmatization, and Stemming, and feature extraction techniques like GloVe and Bi-grams (2-grams) to capture contextual information and nuances within language. The model integrates continuous learning techniques to handle the dynamic nature of hate speech. It enables the model to update its comprehension of new language patterns and evolving forms of objectionable content. The evaluation of the proposed model involves benchmarking against existing hate speech detection methods, demonstrating superior precision, recall, and overall accuracy. Finally, the proposed MLA offers a practical and adaptable solution for recognizing hate speech, contributing to creating safer online environments.

Keywords—Multi-Layered Approach (MLA); Deep Learning (DL); DistilBERT; GloVe; Bi-grams (2-grams)

I. Introduction

With the increase in OSNS usage, many people use OSNS as a platform for expressing their views through text messages. Platforms like Twitter, Facebook, and Instagram have become more prevalent in India for expressing users' opinions through text, voice, and Videos. A huge amount of data is generated daily from these platforms, consisting of various messages [1]. Hate speech is harmful and can abuse the person personally and on public platforms [2]. Hate speech, defined as the communication between person to person or groups in OSNS based on topics such as religion, gender, nationality, and ethnicity, poses a significant challenge to maintaining a safe and inclusive OSNS.

Many existing hate speech detection models use keyword filters and manual detection, which causes inaccuracy in processing large amounts of data. There is a growing need for effective hate speech recognition systems to mitigate these negative consequences. This research explores developing and implementing hate speech recognition algorithms within OSNS [3]. The objective is to design intelligent systems capable of identifying and flagging instances of hate speech, allowing for timely intervention and moderation. The proposed approach involves utilizing natural language processing (NLP), machine learning (ML), and deep learning techniques to analyze textual content on social media platforms [4]. The impact of social networking sites on hate speech is a complex and multifaceted issue. On one hand, these platforms amplify the reach and speed of communication, allowing hate speech to spread rapidly and influence a large audience. On the other hand, social networking sites also provide opportunities for counter-speech, activism, and promoting tolerance of [5]. This essay will explore the various dimensions of the impact of social networking sites on hate speech, examining the challenges they present and the potential solutions and positive contributions they can make in mitigating the spread of hate online, as explained in [6] [7].

The techniques that are used in this work are as follows. In section 2, the literature is given with a performance analysis. The 3rd section describes the pre-trained model DistilBERT that helps train the two datasets, followed by integrated pre-processing and feature extraction techniques such as Global Vectors for Word Representation (GloVe) and Bi-Gram. Section 4 describes the proposed classification of Text Convolutional Neural Networks (TCNNs) for Hate Speech Detection with required layers, mathematical representation with datasets, Performance Metrics, and Evaluation Results. The final section explains the conclusion and future work of hate speech detection.

II. Literature Survey

With the aim of improving hate speech detection, Watanabe et al. [8] introduced a practical method for gathering and examining offensive and bigoted statements on Twitter.

The goal of the project is to create a comprehensive dataset that encompasses a variety of hate speech expressions while accounting for the dynamic nature of language and the changing methods in which people express negative opinions. In order to gather instances of hate speech, the suggested methodology combines supervised and unsupervised data gathering approaches in a multifaceted approach. In order to accomplish this, a hybrid strategy uses both manual annotation and machine learning algorithms to create a representative and diverse dataset. To ensure the robustness of the data gathered, special consideration is given to account for linguistic variances, emerging trends in hate speech, and contextual nuances. The outcomes show that the algorithm can detect and categorize hate speech on Twitter with an accuracy of 87.4%, indicating its potential for practical use in social media content moderation.

Al-Maatouk et al. [9] look into and assess how social media platforms are being adopted in academic settings using the Task-Technology Fit (TTF) framework and the Technology Acceptance Model (TAM). The purpose of the study is to determine how well social media fits in with the goals and duties of academic professionals and how user acceptance affects how it is used. The theoretical basis is provided by the Task-Technology Fit model, which highlights the significance of matching technology features to users' jobs in order to improve productivity and performance. The Technology Acceptance Model also sheds light on the attitudes, perceived utility, and ease of use of users—all of which are important variables that affect the adoption of new technologies. Using a mixed-methods approach, the technique includes surveys and interviews with academics, researchers, and students from a range of subject areas. To assess task-technology fit, perceived usefulness, and usability, and general acceptance, quantitative data will be collected. In order to obtain a greater understanding of users' experiences and perspectives of integrating social media into academic workflows, it also gathers information through interviews.

In order to determine if a certain person's health data is included in a dataset, Liu et al. [10] use machine learning techniques to explore the susceptibility of social media health data to membership inference attacks. By using data that the machine learning model leaked during training, an adversary can use membership inference attacks to determine whether a particular person's data is included in the training set. The ramifications of such attacks can be dire in the context of social media health data, since people may share private health information with the expectation of privacy. The results show the performance of proposed SocInf obtained the accuracy of 73% and precision of 84%. The investigation ends with recommendations and rules for protecting the privacy of people who post health-related content on social media platforms.

In the context of the big data era, Al-Garadi et al. [11] provided a thorough analysis of the body of research on cyberbullying prediction, with an emphasis on the use of ML techniques. The goal is to highlight unresolved issues in this field, identify important approaches, and offer insights into the state of the research at this time. It includes research that use machine learning methods to anticipate and identify instances of cyber bullying on social media sites. The effectiveness of several methods, such as sentiment analysis, network analysis, and natural language processing, in identifying and stopping cyber bullying is investigated. The review also looks at how big data analytics can be used to manage the enormous volumes of textual and multimedia data produced by social media. The literature now in publication identifies the main issues, which include the ever-changing landscape of cyber bullying, the dynamic character of online communication, and the moral ramifications of automated content moderation.

In order to meet the necessity for efficient and real-time angry emotion recognition in tweets, Roy et al. [12] concentrate on employing LSTM networks as a potential deep learning technique. In comparison to conventional machine learning techniques, the performance of LSTM-based models for hateful sentiment identification is compared in this study. Using this method, a representative and diversified dataset of current tweets with both hateful and non-hateful attitudes is gathered. We preprocess the data taking into account the distinct features of micro blogging sites, like the short text length and colloquial language. Next, put LSTM-based models into practice and assess how well they perform in comparison to other well-known machine learning algorithms, such as SVM and RF. LSTM achieves 0.98, 0.99, and 0.98 for precision, recall, and F1 score, respectively. LSTM was found to be more accurate than other models at 97% to find hateful sentiment.

Khan et al. [13] have developed a novel method to improve hate speech detection performance by combining the advantages of CNN, Bi-GRU, and capsule networks. To efficiently recognize and categorize hate speech in textual data, the suggested model makes use of the spatial hierarchies acquired by CNNs, the contextual knowledge offered by Bi-GRUs, and the dynamic routing mechanism of Capsule Networks. By combining these three elements, the model may extract intricate features, identify long-term relationships, and acquire hierarchical representations, all of which contribute to an increase in the overall accuracy of hate speech identification. While the Bi-GRU component enhances the model's comprehension of context by capturing sequential relationships in both forward and backward directions, the CNN component concentrates on extracting local features and patterns from the input text. The results reveal that the suggested strategy is effective, with superior metrics such as recall of 0.80, F1-score of 0.84, and precision of 0.90. Additionally, an analysis is conducted on the model's interpretation and resilience against different forms of hate speech, highlighting its possible practical applications.

A modified TF-IDF technique is presented by Almammary [14] for the classification of Arabic questions. Although it is a commonly used technique in text categorization and information retrieval, its efficacy may be limited in languages with intricate morphology, such as Arabic. The study's modified TF-IDF method tackles the difficulties caused by the linguistic subtleties of Arabic. The changes include taking language-specific elements into account, taking word roots into account, and taking question syntactic structure into account. Furthermore, a unique weighting technique is presented to help enhance classification accuracy by prioritizing essential phrases. According to preliminary findings, the modified TF-

IDF methodology performs better at correctly classifying Arabic questions than conventional methods. The proposed approach results obtained that accuracy is 0.597, recall obtained with 0.596, and precision with 0.636 which is high compared with existing models.

A thorough methodology for hate speech identification using a DCNN is proposed by Roy et al. [15]. In order to effectively identify hate speech, the framework makes use of deep learning to automatically learn from textual data and extract pertinent attributes. The suggested paradigm aims to tackle the difficulties created by hate speech's complex and context-dependent nature. To collect local and global contextual information in the text, the use an embedding layer, tokenization, and attention mechanisms pre-processing pipeline. Because of the deep CNN architecture's effective handling of language's hierarchical structure, the model is able to identify minute patterns that may be signs of hate speech. The usefulness of the proposed system is demonstrated in terms of specified performance measures through evaluation on benchmark datasets containing labeled instances of hate speech.

Oriola et al. [16] concentrate on the particular context of tweets from South Africa with the goal of assessing and contrasting different ML methods for the identification of hate and abusive speech in this distinct language and cultural environment. The research dataset is made up of a wide range of tweets that were contributed by people in South Africa, which represent the social subtleties and linguistic diversity of the country. The efficiency of various ML models, such as NLP, sentiment analysis, and NLP techniques, in precisely recognizing and classifying hate and offensive speech in this setting Pre-processing the twitter data using the methodology entails feature extraction, tokenization, and language normalization. The effectiveness of more sophisticated DL models, like RNN and BERT, and more conventional ML techniques, like SVM and RF, are compared.

SLMF-CNN architecture is put forth by Akhter et al. [17] for document-level text classification. By using a single convolutional layer with variable filter sizes, the SLMF-CNN model is able to extract a variety of n-gram features from the input text. The model can learn hierarchical representations by utilizing multisize filters, which capture both coarse- and fine-grained data. To further improve generalization and avoid over fitting, we also use dropout regularization. Using benchmark datasets for document-level text classification tasks, the proposed SLMF-CNN was evaluated against other models. The experimental findings show that our model performs as well as or better than the competition in terms of efficiency and accuracy.

Zheng et al. [18] introduce an attention-mechanism-equipped H-BRCNN as a novel text categorization method. The proposed method leverages the benefits of convolutional and bidirectional recurrent layers to extract sequential relationships as well as local patterns from textual input. Adding attention mechanisms increases the model's ability to focus on important parts of the input stream. The H-BRCNN's architecture consists of convolutional layers to identify local patterns, attention mechanisms to dynamically balance the significance of various input sequence segments, and bidirectional recurrent layers to efficiently record contextual information from both directions. Because of its hybrid nature, the model can effectively extract hierarchical patterns and relationships from the text, giving rise to a more thorough comprehension of the input data. The outcomes of our experiments show how well our strategy works in reaching competitive accuracy and surpassing current techniques in a range of text classification challenges. Furthermore, every element of the hybrid architecture is examined in this work, offering insights into the roles played by attention mechanisms, bidirectional recurrent layers, and convolutional layers.

The TACNN is a revolutionary technique presented by T. He et al. [19] that is intended to efficiently recognize text in complicated scenarios. TACNN uses convolutional neural networks and attention mechanisms to extract complex textual patterns in a variety of backgrounds. A text-specific attention mechanism that dynamically focuses on areas most likely to contain text is integrated into the suggested model. The network can process information selectively thanks to this attention mechanism, which improves its capacity to distinguish text from non-text elements. In order to learn discriminative features from different text appearances, sizes, and orientations, the convolutional layers are optimized. The trials show off state-of-the-art performance in terms of multiple parameters, demonstrating the effectiveness of TACNN on benchmark datasets.

An innovative architecture that makes use of 3D convolutional layers is put forth by Ouyang et al. [20] in an effort to more accurately represent the spatial correlations seen in words. The spatial pyramid pooling algorithm is integrated to handle different sentence lengths and efficiently capture multi-scale characteristics. As a result, the network is better able to adapt to a wider range of language patterns by processing sentences with varying durations and hierarchies. Benchmark datasets for sentence-level classification tasks, including sentiment analysis and topic categorization, are used in the trials. Our findings show that when compared to state-of-the-art techniques, the suggested 3D convolutional network with spatial pyramid pooling provides competitive or better performance. The model demonstrates resilience when dealing with different sentence lengths, demonstrating its efficacy in capturing complex spatial connections that are essential for precise sentence-level classification. To overcome the drawbacks of current designs, Y. Du et al. [21] present a novel CHNN for sentiment categorization. In order to increase overall sentiment analysis performance and the model's capacity to collect hierarchical features, the CHNN combines the best aspects of both classic neural networks and capsule networks. The CHNN's capsule network modules are made to effectively simulate the textual material's hierarchical structure. Capsules allow the network to better understand the intricate dependencies within phrases and documents by recording part-whole hierarchies and retaining spatial links. Furthermore, by utilizing the advantages of both convolutional and recurrent layers—which excel in feature extraction and sequential information processing, respectively—the hybrid design combine both advantages. The benchmark datasets for sentiment analysis are used in the experiments. The outcomes show that, when it comes to initialized metrics, the CHNN performs better than cutting-edge models.

Fazil et al. [22] offer a novel hybrid methodology that blends machine learning approaches with rule-based methods. The suggested system increases the precision and effectiveness of spam detection by utilizing the benefits of both rule-based and machine learning techniques. The ML component makes use of sophisticated algorithms, like ensemble methods and deep learning, to examine big datasets and identify complex patterns that point to automated spamming activity. Simultaneously, the rule-based component enhances the system's capacity to detect subtle spamming strategies by including predetermined rules and heuristics based on the unique characteristics of spam accounts. The proposed approach achieves high accuracy, scalability, and adaptability by fusing the advantages of rule-based and machine learning techniques. This helps to support continuous efforts to maintain a reliable and spam-free online social ecology.

Oma et al.'s [23] assessment centre on how well ML and DL algorithms detect hate speech, particularly in Arabic-language content on OSNs. We offer a comparative study of several ML and DL models, taking into account their computational efficiency, generalization potential, and performance measures. Our dataset is made up of a wide variety of Arabic text data that has been gathered from several OSNs and has been manually annotated for hate speech content. Both cutting-edge DL models like CNN and LSTM and conventional ML models like SVM, RF, and NB are included in the selection of methods to detect hate speech with precision and minimal false positives. Sajjad et al. [24] introduced a fusion approach to improve the precision and resilience of hate speech detection systems. The fusion approach builds a complete model for hate speech detection by combining data from various sources, including textual, visual, and contextual aspects. Analyzing the text's content, including its usage of slurs, derogatory language, and other discriminatory terms, is part of identifying its textual aspects. While contextual features take into account the communication's surrounding context, including past behavior and user interactions, visual features concentrate on identifying objectionable visuals or symbols. The results of this study enhance the field of hate speech recognition technology and offer a more complete and potent means of preventing the spread of damaging content online.

Zhang et al. [25] explore the challenges of recognizing and categorizing rare but essential cases of hate speech that have ramifications for creating a safer online community. A distinct set of challenges is brought about by the long tail phenomena in hate speech, such as the lack of labeled data for infrequent occurrences and the persistent adaptability of malevolent actors to evade detection systems. Ultimately, to overcome the difficulties the long tail presents, this research integrates machine learning methods with a sophisticated comprehension of linguistic patterns, contextual clues, and cultural variances. The cutting-edge algorithms are used to investigate new strategies to improve the detection precision for these infrequent but significant hate speech incidents.

TABLE I. DIFFERENT MACHINE LEARNING MODELS THAT APPLIED ON VARIOUS HATE SPEECH DATASETS

Author Name	Proposed Model	Dataset	Performance Analysis	Research Gaps
Ombui et al. [26]	SVM	Annotated Tweets	Accuracy-0.825	Low accuracy while performing on code switched language datasets
Plaza-Del-Arco et al. [27]	Multi-task approach	HatEval, MeX-A3T	Accuracy-91.92	Only limited hate speech text is detected
Sreelakshmi et al. [28]	SVM-RBF	Hindi Datasets DS1, DS2, DS3	DS1-Accuracy-64.15, DS2-75.11, DS3-85.81	Very limited Accuracy and can work on small datasets
Kapil et al. [29]	Deep-MTL	Five hate speech datasets	MacroF1 value for D1-89.30, D2-92.12, D3-86.12, D4-92.41, D5-86.05	More computation time.

Table I shows the several existing and proposed models that help to analyze performance of various algorithms. And also this gives the research gaps of the existing approaches with performance metrics. All the models belong to DL algorithms.

III. METHODOLOGY

A. DistilBERT Pre-trained Model

DistilBERT is a more compact and effective version of the BERT (Bidirectional Encoder Representations from Transformers) concept that nevertheless achieves competitive results. DistilBERT, created by Hugging Face, is better suited for contexts with limited resources because it requires less parameter while maintaining the majority of BERT's language understanding capabilities. An NLP task called "hate speech detection" looks for and classifies material that uses damaging or insulting language. This work is essential to preserving a welcoming and safe online community. Many terms that are used to discriminate, such as racism, sexism, homophobia, and more, can be considered hate speech. Using DistilBERT for hate speech identification, the model is trained on a labelled dataset that includes examples of both hateful and non-hateful words. The program gains the ability to predict whether a particular text contains hate speech by extracting contextual information from the input text. DistilBERT goes through pre-training on a sizable corpus of text data, just like BERT. The algorithm picks up contextual links between words and learns to predict missing words in phrases during pre-training. DistilBERT is refined on a particular hate speech detection dataset following pre-training. In order to create accurate predictions based on the training data, the model is exposed to labeled instances of both hate and non-hate speech, modifying its parameters accordingly. After training, new, unknown text can be classified by the DistilBERT model as either hate speech or not. After training, new, unknown text can be classified by the DistilBERT model as either hate speech or not. This is accomplished by running the model over the input text, and the model returns a probability score that indicates the possibility of hate speech. DistilBERT's lower size makes it more computationally efficient for use in real-world applications, which is useful for hate speech detection. Faster inference times are possible without noticeably compromising performance.

B. *Integrated Pre-processing*

The process of cleaning hate speech data entails preparing the text by eliminating superfluous material, standardizing the structure, and addressing any irregularities or noise. Initially, all text will be converted to lowercase in order to maintain consistency and simplify the data. Remove all special characters, stop words (such as "the," "and," "is," etc.), URLs, punctuation, and symbols that don't add anything useful to the analysis. This aids in preserving consistency. Divide the text into discrete words or phrases. This stage is crucial for feature extraction and additional analysis. This model also makes use of stemming and lemmatization as preprocessing methods. The process of reducing a word to its root or base form is called lemmatization. In contrast to stemming, which removes prefixes or suffixes in order to get at the word root, lemmatization takes the word's meaning into account and uses morphological analysis to determine the word's basic form, or lemma. It uses morphological analysis and language rules, and it frequently consults dictionaries. Part-of-speech tagging can be done in conjunction with lemmatization. In order to determine a word's root form, a set of rules is iteratively applied to it using the Lovins stemming algorithm (LSA), as described in this study. These criteria, which are based on linguistic patterns, are designed to capture frequent word form changes and suffixes. The two procedures pertaining to data cleansing.

C. *Apply Rules in Sequence*

- The input word is subjected to the algorithm's collection of rules, one after the other.
- Removing particular suffixes or making other changes in accordance with linguistic patterns are examples of rules.

D. *Iterative Process:*

- It is common practice to apply rules iteratively until no more can be applied.
- Until the word takes on a stable or reduced form, this iterative procedure is continued.

E. *Feature Extraction Technique*

An unsupervised learning approach called Global Vectors for Word Representation (GloVe) is used to generate vector representations of words. GloVe's primary goal is to identify word vectors by examining a corpus's global co-occurrence data for each word. The GloVe model's goal is to train word vectors so that the co-occurrence probabilities of words are reflected in the dot product of these vectors. Optimization aims to minimize the discrepancy between the logarithm of the observed co-occurrence probability and the dot product of word vectors. An international word-word co-occurrence matrix is used to train the model.

V: The vocabulary size (number of unique words in the corpus).

X: The word-word co-occurrence matrix, where Xab represents the number of times word a co-occurs with word b in the corpus.

W: The word vector matrix, where Wa represents the vector for word a.

The optimization objective of GloVe is to minimize the following cost function given in Eq(1).

$$J = \sum_{a=1}^{V} \sum_{b=1}^{V} f(X_{ab})(W_a^T \cdot W_b + y_a + y_b - \log(X_{ab}))^2 \quad (1)$$

$f(X_{ab})$ is a weighting function that can be used to down-weight the influence of very frequent word pairs.

$y_a + y_b$ are bias terms for words a and b.

The weighting function $f(X_{ab})$ is applied to adjust the importance of each co-occurrence count. A logarithmic weighting function is given in Eq(2):

$$\widetilde{X}_{ab} = f(X_{ab}) = \log(1 + X_{ab}) \quad (2)$$

F. *Bi-Gram*

The other name for bi-grams is 2-grams, are groups of two neighboring elements in a particular text. These components are frequently words in the context of NLP. Applications for bi-grams include information retrieval, text processing, and language modeling. A bi-gram is made up of any one of n-1 possible pairings of neighboring elements in a sequence of n elements. Every component—aside from the final one—contributes to a bigram. You may come across the idea of conditional probability for bi-grams in the context of probability and language modeling. The following is the equation for the conditional probability of a word given the preceding word (a bi-gram):

$$P(w_i|w_{i=1}) = \frac{\mathrm{Count}_{(w_{i-1},w_i)}}{\mathrm{Count}(w_{i-1})} \quad (3)$$

w_i is the current word,

$w_{i=1}$ is the previous word,

$\mathrm{Count}_{(w_{i-1},w_i)}$ is the number of occurrences of the bi-gram.

$\mathrm{Count}(w_{i-1})$ is the number of occurrences of the word.

IV. TEXT CONVOLUTIONAL NEURAL NETWORKS (TCNNS) FOR HATE SPEECH DETECTION

The rise in online communication platforms in recent times has resulted in a rise in the occurrence of harmful content such as hate speech and cyberbullying. Detecting and mitigating such content is crucial for maintaining a safe and inclusive digital environment. Text Convolutional Neural Networks (TCNNs) have emerged as powerful tools for automated text analysis, particularly in the domain of hate speech detection. Hate speech is characterized by offensive language, discriminatory remarks, or expressions that incite violence or prejudice against specific individuals or groups. Manual moderation of online content is challenging due to the sheer volume of data generated daily [30]. Hence, there is a growing need for automated solutions that can efficiently identify and filter out hate speech. CNN include a variation called TCNN that are specifically designed to handle textual data. Initially intended for image identification, CNNs have demonstrated impressive performance across various NLP tasks. TCNNs are particularly well-suited for identifying hate speech because they take advantage of language's hierarchical and compositional nature to identify local patterns and relationships within the text. The proposed architecture is given in Fig. 1.

Multiple layers, including convolutional, pooling, and fully linked layers, are commonly found in TCNs. The convolutional layers filter local sections of the input text to extract features such as word embeddings and n-grams. Pooling layers help reduce dimensionality, and fully connected layers enable the model to learn global patterns and make predictions. Developing an effective hate speech detection system using TCNNs comes with challenges such as handling sarcasm, context dependence, and evolving language trends. Additionally, ethical considerations surrounding biases in training data and potential limitations in generalization must be addressed.

Fig. 2 describes the working of process of each and every layer present in the proposed T-CNN model. T-CNN mainly process the step-by-step given in this figure. The main step in this figure is extracting features from the given input text and this is carried out by convolution layer. The pooling layer and fully connected layer gives classification results.

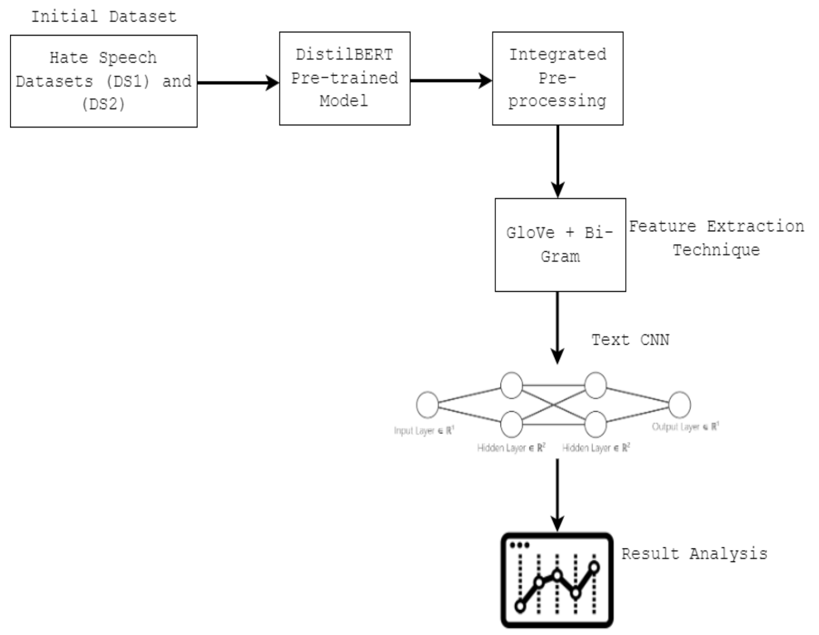


Fig. 1. The System srchitecture for Text-CNN.

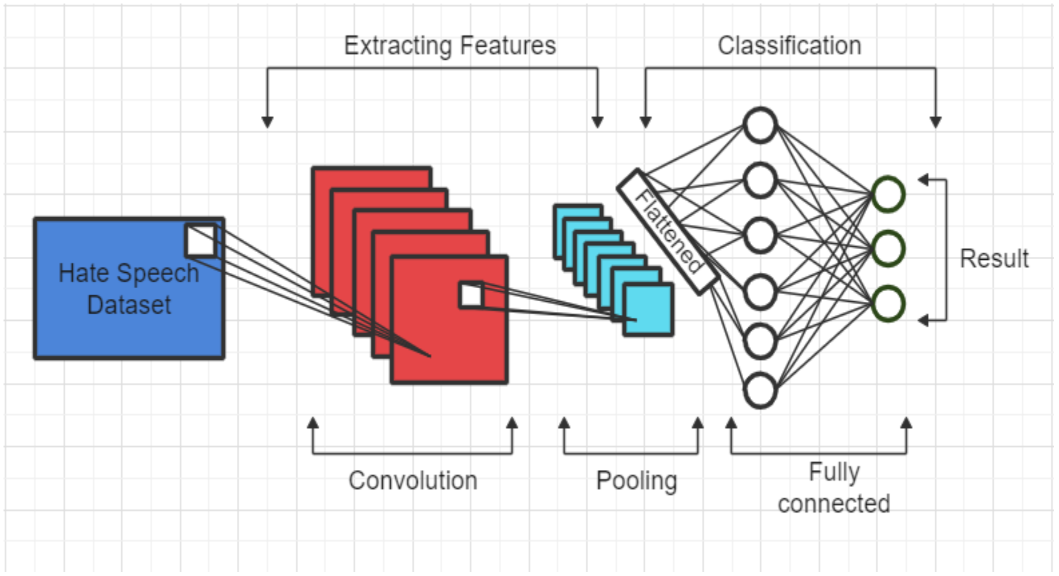


Fig. 2. The Architecture of text CNN approach for hate speech classification based on layers present in the T-CNN model.

The following layers that are used to classify the twitter dataset:

Input Layer: A sequence of word embeddings representing the input text. The input layer typically involves the conversion of text data into numerical representations that can be fed into the network.

Embedding Layer: The input sequence of words convert into dense vectors (word embeddings). Use pre-trained word embeddings to convert words into dense vectors, or create your own embeddings and train them.

Let m_i be the one-hot encoded vector for the word, ev_a be the embedding vector for the a^{th} word, and M be the embedding matrix. One way to depict the embedding layer is as follows:

$$ev_a = M \cdot m_a$$

Input Matrix: The input text sequence is converted into a 2D matrix.

For example the input text has N words and each word is represented by a dimensional embedding vector, then the input matrix X can be formed by stacking these embedding vectors.

$$X = [e_1, e_2, \ldots., e_N]$$

Convolutional Layer(s): Its main goal is to extract the local patterns and properties of the word embeddings. To extract local features from the input text, the convolution operation in NLP entails swiping a tiny filter, called a kernel, across the text. The convolution operation allows the model to identify specific combinations of words or phrases that may indicate hate speech. Filters are small windows that move across the input text during the convolution operation. Every filter picks up on a particular characteristic, like the existence of words or phrases connected to hate speech. A feature map, which shows where these features are present throughout the input, is the result of the convolution procedure. With a filter w of length F and an input sequence x of length N, the convolution operation's output y is calculated as follows:

$$y[i] = \sum_{j-0}^{F-1} x[i+j] \cdot w[j] + b$$

x[i] is the input at position i in the sequence.

W[j] is the weight of the filter at position j.

b is the bias term.

Max Pooling Layer(s): It performs the max pooling over the output of the convolutional layer to capture the most important features. Max pooling is a type of pooling layer commonly used in CNN for feature extraction. In the context of hate speech detection or any text classification task, it typically use 1D convolutional layers followed by max pooling to capture important features from the input text. In this scenario, a sequence of features from your previous layers, and you want to apply max pooling to obtain a fixed-size representation. A sequence of input vectors and you want to apply max pooling with a pool size of k. The output of the max pooling operation for each segment is given by:

$$y_i = \max(x_i, x_{i+1}, \ldots., x_{i+k-1})$$

Flatten Layer: The output of the max pooling layer is flattened into a one-dimensional vector by this layer.

$$X_{flat} = \text{Flatten}(X_{pool})$$

Fully Connected (Dense) Layer(s) (FCLs): It applies a linear transformation to the flattened vector to produce the final output.

$$X_{fc} = \text{ReLU}(\text{Dense}(X_{flat}))$$

Output Layer: The final output is generated, typically using softmax activation for classification tasks.

$$X_{output} = \text{Softmax}(\text{Dense}(X_{fc}))$$

A. *About Dataset*

The proposed model uses the two twitter datasets that DS1 and DS2. The DS1 contains 9484 tweets with four labels such as aggressive, bullying and spam and normal and DS2 contains 24802 tweets with three labels such as hateful, offensive, neither. Among these two datasets the training testing is divided into 70:30 ratio, 70% for training and 30% for testing. Table II shows the summary of all the datasets used in this context and Table III shows the sample hate and normal speech text.

TABLE II. SUMMARY OF DATASETS USED IN THIS WORK

Datasets	#Tweets	Labels
DS1 (Kaggle Twitter)	9484	Aggressive, Bullying, Spam, Normal
DS2 (Kaggle Twitter)	24802	Hateful, Offensive, Normal

TABLE III. SAMPLE HATE AND NORMAL SPEECH TEXT

Hate and Normal Speeches	Description
Hate-1	Females think dating a p***** is cute now? how does doing this stuff make him a p*****
Hate-2	Him s** me p***** wetter then a shower curtain....
Hate-3	How about them Cowboys!!!!" Shutup p****
Normal	Drakes new shoes that will be released by Nike/Jordan....

B. *Performance Metrics*

The performance of proposed MLA evaluated using the default data metrics such as Accuracy (ACC), Specificity (Spc), Precision (Pre), Recall (re) and F1-score (F1S). All these metrics are based on count values of true positive (TP), False positive (FP), True Negative, and False Negative. Here, the TP represents the hate tweets with accurate classification. FP represents the normal tweets classified as hate tweets. TN represents the accurately classified normal tweets. In the final stage, FN represents the hate tweets classified as normal tweets. The proposed MLA applied on DS1 datasets consists of 9484 tweets with four classes but the proposed approach consider all the classes such as Aggressive, Bullying, Spam, Normal. The DS2 consists of three classes such as Hateful, Offensive, Normal speech with 24802 tweets. After the training applied the testing is applied on 7700 tweet.

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Specificity (Spc)} = \frac{\text{No of TN}}{\text{No of TN} + \text{No of FP}}$$

$$\text{Recall (Re)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} - \text{Score (F1S)} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

$$\text{Precision (Pre)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

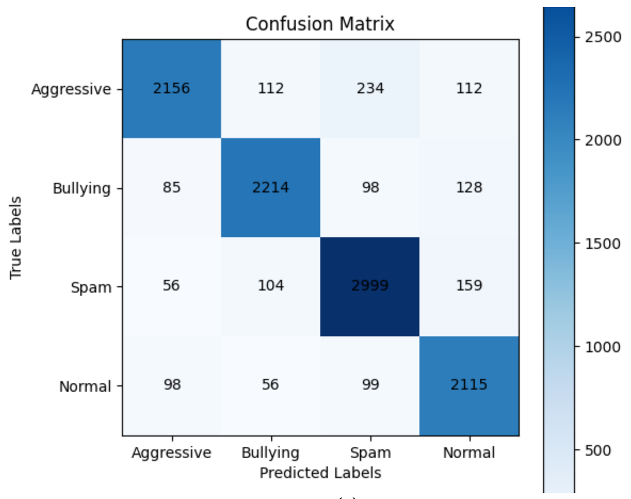
C. *Results and Discussions*

The proposed algorithm MLA is implemented by using Python programming language. All the experiments are performed by using the Windows 10 with I5 as the processer, 16 GB RAM, and 20GB hard drive. All the results are based on the count obtained from the confusion matrix attributes. Fig. 3(a) shows the performance of LSTM based on the count values obtained from the confusion matrix. LSTM is the existing model that shows the classification based on hate speech types.

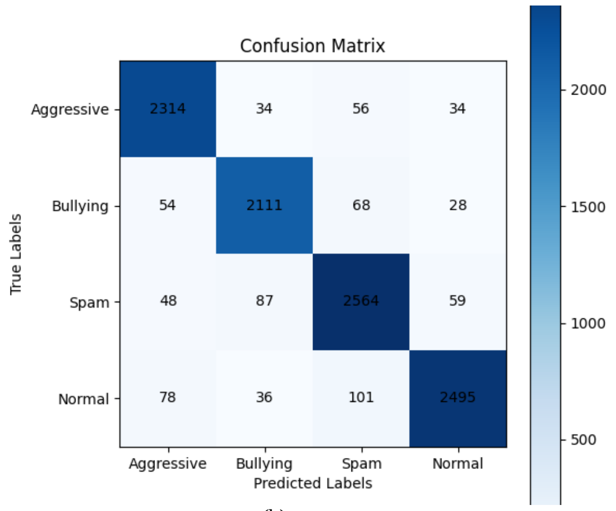
Fig. 3(b) shows the count values obtained from the confusion matrix using HCovBi-Caps. This is the multi-class classification based on four instances. These data was collected from DS1 and it is the labeled data which belongs to Hate speech.

Fig. 3(c) show the performance of MLA in terms of count values based on type of hate speech. All these data are text data collected from Kaggle. The count values show the confusion matrix based on predicted and actual values. It is also the multi-class classification count values obtained from the MLA. Table IV shows the performance of algorithms that classify hate speech on the DS1 dataset. The MLA obtained better classification results than existing models among all the algorithms. The lowest accuracy for the DS1 dataset was LSTM, with 0.88 accuracy for all the classes. The highest accuracy for DS1 was MLA, which achieved an accuracy of 0.95, the best performance.

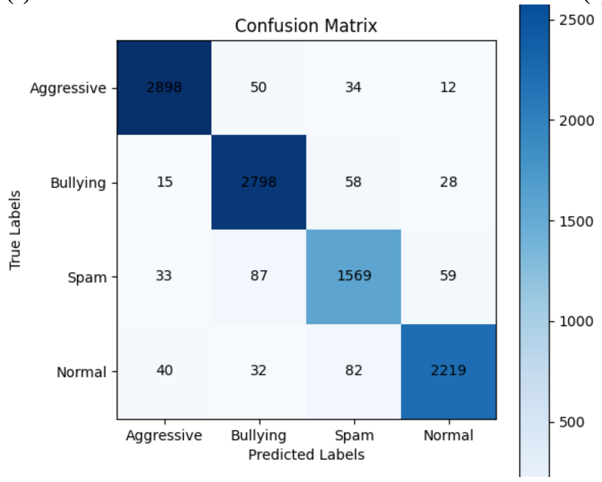
Fig. 4 shows the performances of DL algorithms compared with the proposed MLA, which shows high performance in terms of given parameters for DS1—all these values and performances were obtained using the different labeled data types.



(a)



(b)



(c)

Fig. 3. (a) Count Values of LSTM, (b) Count Values of HCovBi-Caps, (c) Count Values of MLA.

TABLE IV. THE PERFORMANCE OF DL ALGORITHMS BASED ON CLASSIFICATION OF HATE SPEECH FOR DS2

Parameters		**LSTM**	**HCovBi-Caps**	**MLA**
Accuracy	Aggressive	0.88	0.93	0.95
	Bullying	0.88	0.93	0.95
	Spam,	0.88	0.93	0.95
	Normal	0.88	0.93	0.95
Recall	Aggressive	0.82	0.96	0.96
	Bullying	0.87	0.94	0.96
	Spam,	0.90	0.91	0.89
	Normal	0.89	0.91	0.93
F1S	Aggressive	0.86	0.95	0.96
	Bullying	0.88	0.92	0.95
	Spam,	0.88	0.92	0.89
	Normal	0.86	0.93	0.94
Precision	Aggressive	0.90	0.94	0.97
	Bullying	0.89	0.90	0.94
	Spam,	0.87	0.92	0.90
	Normal	0.84	0.95	0.95

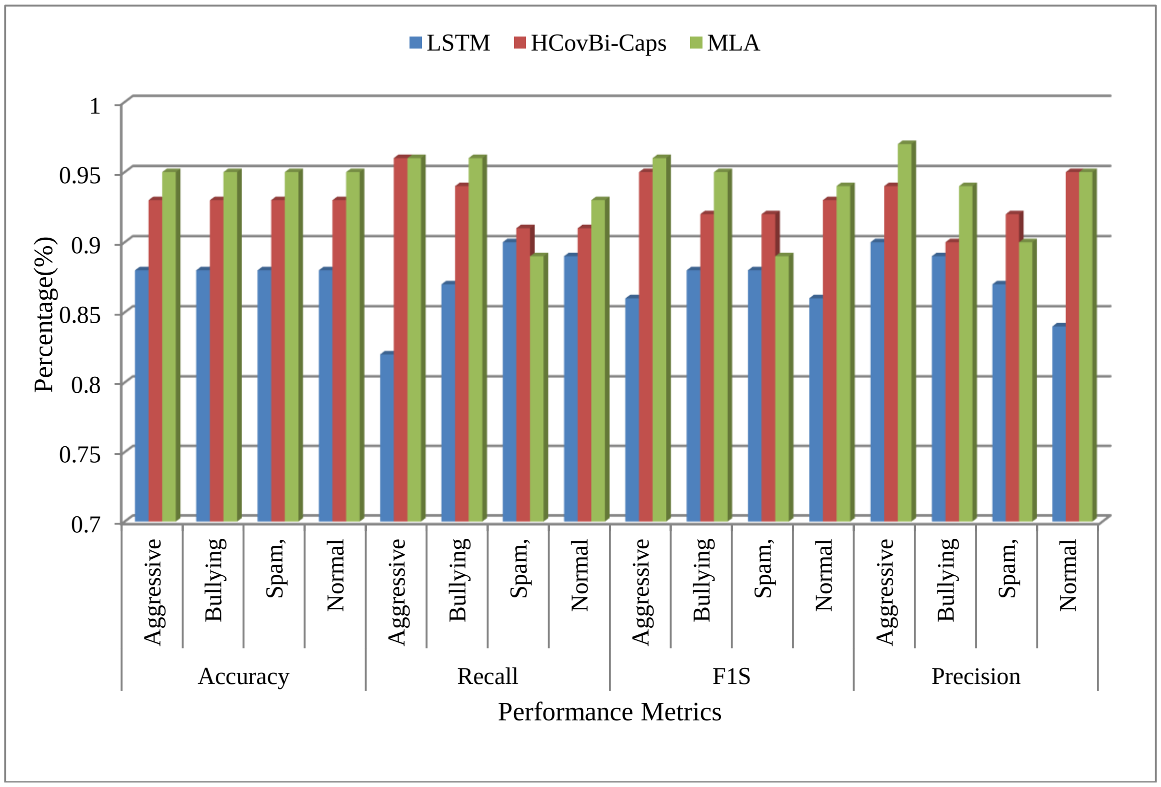
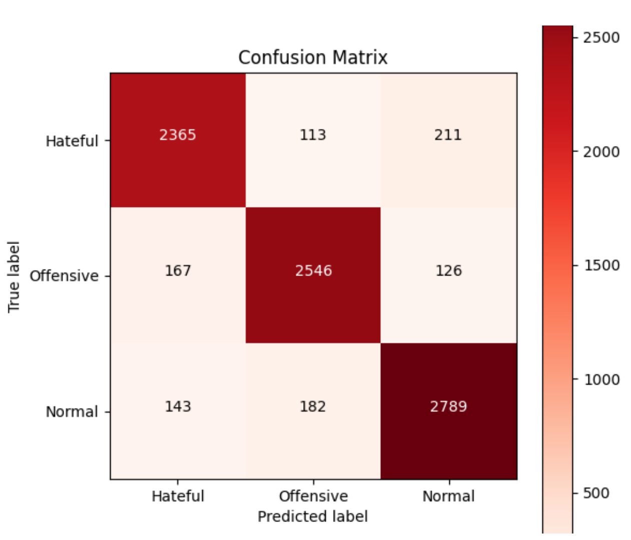
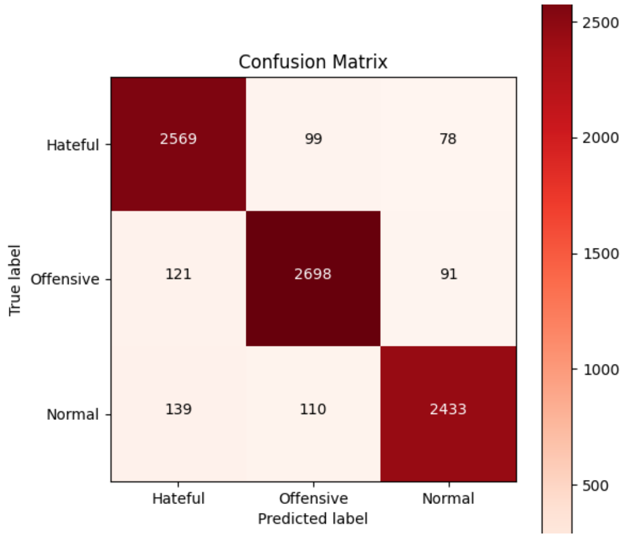


Fig. 4. Performance of various algorithms applied on DS1.

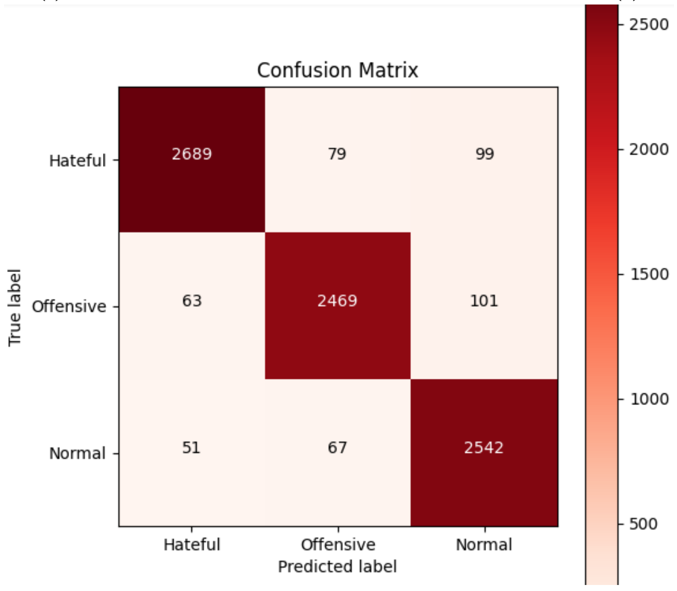
Fig. 5(a) obtained the multi-class classification of several types of Hate speech text messages from DS1 dataset using LSTM. It is the existing approach that classifies the hate speech messages based on the preprocessing, feature extraction and word classification. Fig. 5(b) shows the performance of multi-class classification by using the HCovBi-Caps model. It is the model that classifies the text messages with Hateful, Offensive and normal messages. Fig. 5(c) shows the count values of hate speech with improved classification results. Table V shows the comparative performance of various algorithms that performed on DS2 dataset. The proposed approach shows the high values with accuracy 0.94% for all the classes, recall of 0.91% (average), F1S of 91.5% (average of all the classes) and precision with the 0.93% (average of all the classes). Fig. 6 shows the overall performances of all the algorithms with multi-class classification.



(a)



(b)



(c)

Fig. 5. (a): Count Values of Hate Speech types based on LSTM, (b): Count Values of Hate Speech types based on HCovBi-Caps, (c): Count Values of Hate Speech types based on MLA

TABLE V. THE PERFORMANCE OF DL ALGORITHMS BASED ON CLASSIFICATION OF HATE SPEECH FOR DS2

Parameters		**LSTM**	**HCovBi-Caps**	**MLA**
Accuracy	Hateful	0.89	0.91	0.94
	Offensive	0.89	0.91	0.94
	Normal	0.89	0.91	0.94
Recall	Hateful	0.87	0.93	0.93
	Offensive	0.89	0.91	0.93
	Normal	0.89	0.90	0.95
F1S	Hateful	0.88	0.91	0.94
	Offensive	0.89	0.92	0.94
	Normal	0.89	0.92	0.94
Precision	Hateful	0.88	0.89	0.95
	Offensive	0.89	0.92	0.94
	Normal	0.89	0.93	0.92

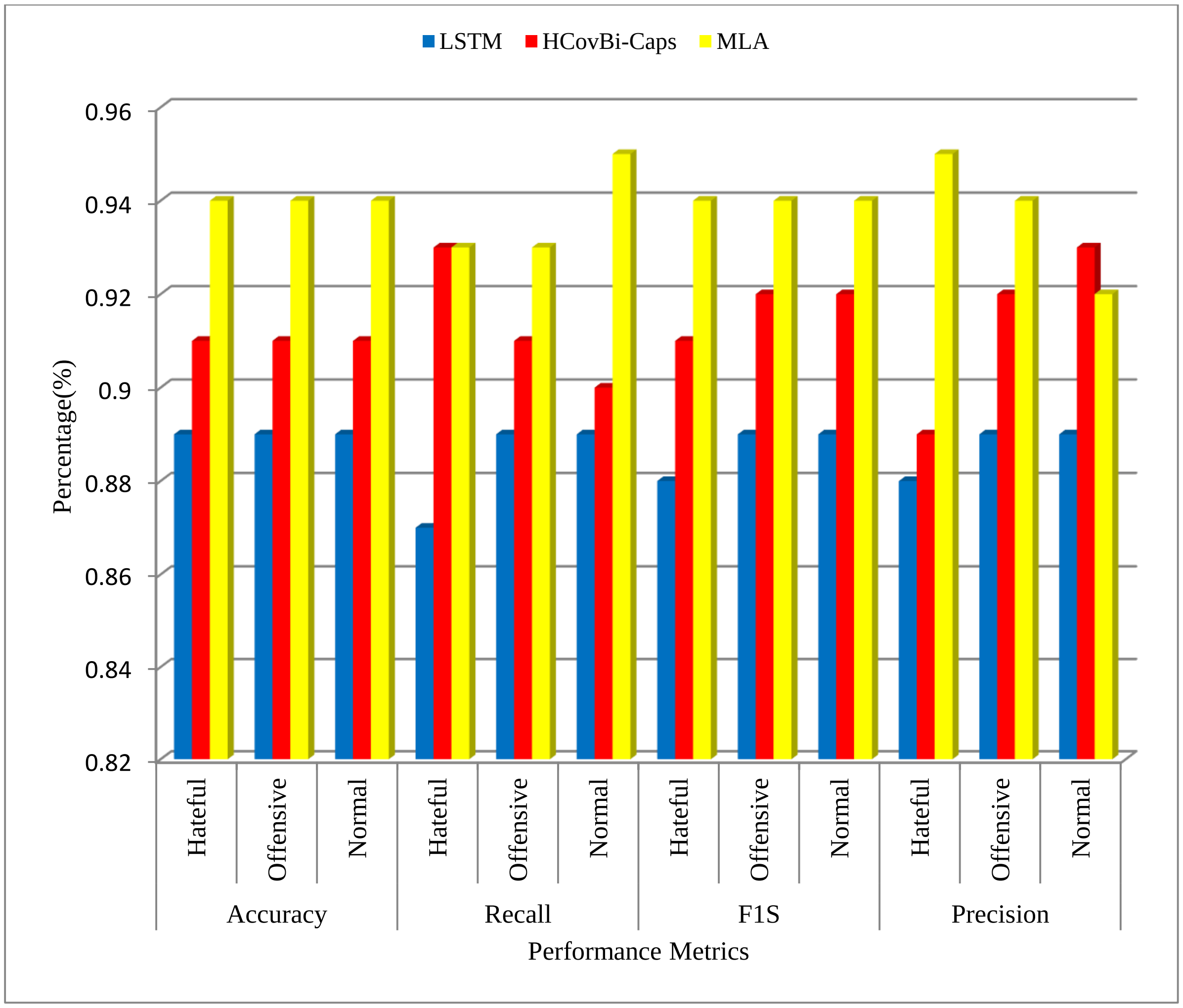


Fig. 6. Classification of hate speech using several algorithms.

V. Conclusion

The use of a Multi-Layered Approach (MLA) in hate speech detection and classification has shown to be a reliable and successful tactic for handling the intricate problems involved in locating and classifying hate speech on a variety of online platforms. This multifaceted approach increases the precision and effectiveness of hate speech detection systems by utilizing linguistic, contextual, and machine learning techniques. The accuracy of hate speech detection is greatly increased by combining several layers, such as machine learning models, semantic analysis, and lexical analysis. The technology can distinguish between non-hateful statements and offensive language more accurately by looking at contextual details and linguistic subtleties. The ethical ramifications of developing and deploying an MLA for hate speech identification must be carefully considered. It is imperative to strike a balance between defending free expression and fighting hate speech, and ongoing efforts should be made to prevent biases and unexpected repercussions in the process of detection. Future iterations of these systems will require constant development, cooperation, and ethical considerations in addition to ongoing study.

References

[1] A. Sharma and R. Bhalla, "Automatic and Advance Techniques for Hate Speech Detection on Social Media: A Review," 2022 Algorithms, Computing and Mathematics Conference (ACM), Chennai, India, 2022, pp. 54-61, doi: 10.1109/ACM57404.2022.00017.

[2] N. S. Mullah and W. M. N. W. Zainon, "Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review," in IEEE Access, vol. 9, pp. 88364-88376, 2021, doi: 10.1109/ACCESS.2021.3089515.

[3] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text", ACM Comput. Surv., vol. 51, no. 4, pp. 1-30, Sep. 2018.

[4] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter", Proc. NAACL Student Res. Workshop, pp. 88-93, 2016.

[5] R. Alshalan and H. Al-Khalifa, "A deep learning approach for automatic hate speech detection in the Saudi Twittersphere", Appl. Sci., vol. 10, no. 23, pp. 1-16, 2020.

[6] A. Alrehili, "Automatic hate speech detection on social media: A brief survey", Proc. IEEE/ACS 16th Int. Conf. Comput. Syst. Appl. (AICCSA), pp. 1-6, Nov. 2019.

[7] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco and V. Patti, "Resources and benchmark corpora for hate speech detection: A systematic review", Lang. Resour. Eval., vol. 55, no. 2, pp. 477-523, Jun. 2021.

[8] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and

perform hate speech detection," IEEE Access, vol. 6, pp. 13825–13835, 2018.

[9] Q. Al-Maatouk, M. S. Othman, A. Aldraiweesh, U. Alturki, W. M. Al-Rahmi, and A. A. Aljeraiwi, "Task-technology fit and technology acceptance model application to structure and evaluate the adoption of social media in academia," IEEE Access, vol. 8, pp. 78427–78440, 2020.

[10] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, "SocInf: Membership inference attacks on social media health data with machine learning," IEEE Trans. Comput. Social Syst., vol. 6, no. 5, pp. 907–921, Oct. 2019.

[11] M. A. Al-Garadi, M. R. Hussain, N. Khan, G. Murtaza, H. F. Nweke, I. Ali, G. Mujtaba, H. Chiroma, H. A. Khattak, and A. Gani, "Predicting cyberbullying on social media in the big data era using machine learning algorithms: Review of literature and open challenges," IEEE Access, vol. 7, pp. 70701–70718, 2019.

[12] S. S. Roy, A. Roy, P. Samui, M. Gandomi and A. H. Gandomi, "Hateful Sentiment Detection in Real-Time Tweets: An LSTM-Based Comparative Approach," in IEEE Transactions on Computational Social Systems, doi: 10.1109/TCSS.2023.3260217.

[13] S. Khan et al., "HCovBi-caps: Hate speech detection using convolutional and Bi-directional gated recurrent unit with Capsule network," IEEE Access, vol. 10, pp. 7881–7894, 2022.

[14] A. S. Alammary, "Arabic questions classification using modified TFIDF," IEEE Access, vol. 9, pp. 95109–95122, 2021.

[15] P. K. Roy, A. K. Tripathy, T. K. Das, and X.-Z. Gao, "A framework for hate speech detection using deep convolutional neural network," IEEE Access, vol. 8, pp. 204951–204962, 2020.

[16] O. Oriola and E. Kotze, "Evaluating machine learning techniques for detecting offensive and hate speech in South African tweets," IEEE Access, vol. 8, pp. 21496–21509, 2020.

[17] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-level text classification using single-layer multisize filters convolutional neural network," IEEE Access, vol. 8, pp. 42689–42707, 2020.

[18] J. Zheng and L. Zheng, "A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification," IEEE Access, vol. 7, pp. 106673–106685, 2019.

[19] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," IEEE Trans. Image Process., vol. 25, no. 6, pp. 2529–2541, Jun. 2016.

[20] X. Ouyang, K. Gu, and P. Zhou, "Spatial pyramid pooling mechanism in 3D convolutional network for sentence-level classification," IEEE/ACM Trans. Audio, Speech, Language Process., vol. 26, no. 11, pp. 2167–2179, Nov. 2018.

[21] Y. Du, X. Zhao, M. He and W. Guo, "A novel capsule based hybrid neural network for sentiment classification", IEEE Access, vol. 7, pp. 39321-39328, 2019.

[22] M. Fazil and M. Abulaish, "A hybrid approach for detecting automated spammers in Twitter", IEEE Trans. Inf. Forensics Security, vol. 13, no. 11, pp. 2707-2719, Nov. 2018.

[23] A. Oma, T. A. El-Hafeez and T. M. Mahmoud, Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs, Cham, Switzerland:Springer, no. 1153, 2020.

[24] M. Sajjad, F. Zulifqar, M. U. G. Khan and M. Azeem, "Hate speech detection using fusion approach", Proc. Int. Conf. Appl. Eng. Math. (ICAEM), pp. 251-255, Aug. 2019.

[25] Z. Zhang and L. Luo, "Hate speech detection: A solved problem? The challenging case of long tail on Twitter", Semantic Web, vol. 10, no. 5, pp. 925-945, Sep. 2019.

[26] E. Ombui, L. Muchemi and P. Wagacha, "Hate speech detection in code-switched text messages", Proc. 3rd Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT), pp. 1-6, Oct. 2019.

[27] F. M. Plaza-Del-Arco, M. D. Molina-Gonzalez, L. A. Urena-Lopez and M. T. Martin-Valdivia, "A multi-task learning approach to hate speech detection leveraging sentiment analysis", IEEE Access, vol. 9, pp. 112478-112489, 2021.

[28] K. Sreelakshmi, B. Premjith and K. P. Soman, "Detection of Hate Speech Text in Hindi-English Code-mixed Data", Procedia Computer Science, vol. 171, pp. 737-744, 2020.

[29] P. Kapil and A. Ekbal, "A deep neural network based multi-task learning approach to hate speech detection", Knowl Based Syst, vol. 210, Dec. 2020.

[30] M. Z. Ali, S. Rauf Ehsan-Ul-Haq, K. Javed and S. Hussain, "Improving Hate Speech Detection of Urdu Tweets Using Sentiment Analysis", IEEE Access, vol. 9, pp. 84296-84305, 2021.