Hate Speech Detection using Deep Learning and Text Analysis

Prachi Patil
Computer Engineering
Vidyavardhinis College of Engg. And Tech., University of Mumbai, Vasai, India
prachi2patil2@gmail.com

Sakshi Raul
Computer Engineering
Vidyavardhinis' College of Engg. And Tech., University of Mumbai, Vasai, India
sakshiraul16@gmail.com

Dhanisha Raut
Computer Engineering
Vidyavardhinis' College of Engg. And Tech., University of Mumbai, Vasai, India
dhanisharaut07@gmail.com

Dr. Tatwadashi Nagarhalli
Associate Prof.
Computer Engineering
Vidyavardhinis' College of Engg. And Tech., University of Mumbai, Vasai, India
tatwadarshi.nagarhalli@vcet.edu.in

2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) | 979-8-3503-9725-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICICCS56967.2023.10142895

Abstract-

Hate speech is any form of speech, gesture, written or physical expression that threatens a person or a group based on their race, ethnicity, religion, gender, sexual orientation, nationality, disability, or any other characteristic that is protected by law. Hate speech can take many forms, ranging from verbal harassment to physical violence. Hate speech detection has become an important task in NLP due to the growing frequency of hate speech on online forums and social media. The proposed research work aims to improve hate speech detection by doing modification in standard i.e., Modified bi-LSTM model vs RCNN. The study examines how well the modified model performs on tasks involving the classification of hate speech when compared to a conventional LSTM model. The improved bi-LSTM model is intended to capture the context and relationships more accurately between the words in hate speech utterances.

The study uses a publicly accessible dataset of tweets containing hate speech and tweets without any hate speech. The proposed model is trained and tested with the help of various performance metrics such as F1-score, accuracy and precision, recall. The research outcomes show that the proposed model outperforms the standard LSTM model in detecting hate speech.

Keywords— CNN (Convolutional Neural Network), SVM (Support Vector Machine), BERT, LSTM (long short-term memory networks), Bi-LSTM (Bidirectional Long Short-Term Memory), Lexical Syntactical Feature (LSF), Deep Learning (DL), RNN (Recurrent Neural Network), ANN (Artificial Neural Network), DCNN (Deep Convolution Neural Network, GRU (Gated Recurrent Unit), Recurrent Convolutional Neural Network(RCNN)

I. INTRODUCTION

Hate speech is kind of expression that disparages, dehumanizes, or calls for retaliation against a specific person or group of people because of their identity, such as their race, ethnicity, religion, gender, sexual orientation, or nationality [15]. The prevalence of hate speech in this society has become a major concern for many individuals, organizations, and governments. A spike in cyberbullying, harassment, and discrimination has been caused by the propagation of hate speech made possible by the emergence of social media platforms [1]. So, in order to reduce the prevalence of hate speech in various platforms can be done by applying algorithms, machine learning techniques. The necessity for the detection of offensive content was driven by both the volume of online content produced, particularly on social media, and the psychological strain of manual moderation.

Hate speech identification is a challenging topic in natural language processing that has recently gained a lot of interest because of the growth in instances of online hate speech.

The study of natural language processing has yielded excellent outcomes in a variety of business fields. These successes have been aided by the adoption of ML and DL techniques in the field of NLP, not only in terms of processing of natural language [16], but also in the different applications of NLP [17].

With the purpose of detecting hate speech, researchers have created a variety of machine learning and deep learning models. A model that may simulate long-term dependencies in sequential data is the LSTM neural network, a subtype of recurrent neural network (RNN). [18].

LSTM has been shown to be effective in many NLP tasks [5], including sentiment analysis, machine translation [19, 20], and speech recognition or emotion recognition [21]. In the domain of hate speech identification, LSTMs can be trained to learn the linguistic characteristics and patterns of hate speech by sequentially analyzing text input. Hate speech can be found in many different languages. Many studies have been conducted on this subject for other languages, such as Portuguese, Urdu [2], and Greek, as well as multilingual approaches [7]. However, majority of papers and materials are in English. [10].

One of the key advantages of LSTM over traditional machine learning models is that it can capture the context and meaning of words, even when they are far apart in a sentence or document [6]. This is especially important for hate speech detection, as hate speech can be subtle and complex, and often requires a deep understanding of the language and context to be identified accurately. To demonstrate the value of the proposed system, it has been compared with Vanilla LSTM and Bi-Directional LSTM.

Both RCNNs (Recurrent Convolutional Neural Networks) and BiLSTMs (Bidirectional Long Short-Term Memory

Networks) are superior to conventional LSTMs (Long Short-Term Memory Networks) in the task of detecting hate speech. The following are some benefits of RCNN and BiLSTM over LSTM for the detection of hate speech:

The following are some benefits of RCNN over LSTM for detecting hate speech: it captures spatial characteristics, can handle inputs of different lengths, and trains and infers more quickly.

BiLSTM has several advantages over LSTM for hate speech identification, including the ability to capture bidirectional context, superior long-term dependency modelling, and customizable architecture.

The findings of this research have important implications in order to create technologies that automatically recognize hate speech and the ongoing efforts to combat hate speech and promote a more inclusive and respectful online environment.

II. RELATED WORK

A. REVIEW

The reviews of recent papers published on hate speech detection have been put through in this section

Hate speech detection works on the dataset which is provided by the users. This system allows to perform the task of separating the abusive +

+or harmful words in the sentence. Chih-chien wang, et. al. [1] developed a system using BERT and Lexicon approach. Both the system worked very well and gave accurate result between 50-70 %. This system currently works for political hate text detection only.

Alternatively, Muhammad Pervez Akhter, et. al. [2] worked on hate speech detection for Urdu language. The technology used was n-gram to detect offensive words. The data was collected in the form of comments from a video-based application called YouTube. Using n-gram the collected Urdu dataset was segregated into offensive and non-offensive data.

Mondher Bouazizi, et. al. [3] proposed the use of machine learning features like sentimental and semantic. The classification was done in two methods binary and ternary. The data collected was very less, so in future a system can be built with greater and richer dictionary of data.

In above mentioned paper the technology is not elaborated. The language or method used for detection of text has not been explained. On the other side, the mentioned accuracy is excellent for detection of text.

Yanyan Yang, et. al. [4] experimented using ELMo, BERT and CNN. Among this CNN give more richer results. But there was one flaw: the level of integration was not substantial enough. This information was taken from Twitter. The information was then categorised, combined, and used to classify the hate speech.

Pradeep Kumar Roy, et. al. [5] experimented using different models and obtained various results. He used LSTM and DCNN models also. Among all of these SVM predicted 53 % of hate speech. Due to inaccurate data set, the predication was at lower rate. The text can be extracted from image and video as a future work. Also, various languages can be taken as dataset.

Hate speech detection utilising bi-directional and convolution gated recurrent units with capsule networks was the subject of a paper published by Pradeep Kumar Roy et al. [6]. Several contextual semantics might be taken into consideration to enhance application performance. Another area of research is to classify hostile multilingual and code-mixed content using extensions in HCovBi-Caps.

Marzieh Mozafari, et. al. [7], used XLM-R and MAML these technologies in their experiment. One of the main issues with XLM was that it needed parallel instances, which can be challenging to acquire at a large enough scale. This was not the case with XLM-R, which uses the self-supervised technique. One of the disadvantages of MAML is that it computationally heavy, as the gradients obtained from it are of a higher order.

Flor Miriam Plaza-Del-Arco ,et. al. [8], the computational cost is higher because of multitasking that leverages other corpora for classification, and it is not compatible in low resource languages.

Khubaib Ahmed Qureshi and Muhammad Sabih [9], the accuracy can be increased by reducing misclassifications and better understanding for classifiers, adding more dataset categories.

Axel Rodriguez, et. al. [10], proposed FADOHS which is based on emotion-analysis and clustering for facebook data. One of the technologies used by them which is LSTM is require more memory for train which is major drawback.

Analysis of Hate Speech Detection by Arum Sucia Saksei, et. al. [11] used DL algorithm with RNN algorithm. Firstly, given data is processed using Data mining to extract information into an understandable structure for further use. Next text Analysis is done. After that RNN is used where data is sorted into two section i.e hated data and not hated data. The tests' average precision, recall, and accuracy, which are presented in this paper were 91%, 90%, and 91%, respectively.

Automated Hate Text Against Women Detection in Twitter Data by Havvanur Sahi et al. [12]. A supervised learning model was developed by the author of this article to classify online harassment of women on Twitter. Among the five machine learning-based classification algorithms utilised were Support Vector Machines (SVM), J48, Naive Bayes, Random Forest, and Random Tree. The results showed that negative information can be accurately identified.

The author of Research Paper [13] by Zahid Hussain Khand, et al. examined the effectiveness of 3 feature engineering strategies and 8 ML algorithms. According to the experimental findings, the support vector machine method performed best when combined with bigram features, with an overall accuracy rate of 79%.

Offensive Language Detection using ANN by Meredita Susanty and four others[14]. In this study, an artificial neural network model is used to classify words as offensive or not while also taking into account the sentence structure to determine the context. Only by using the sigmoid activation function did the computer simulation results demonstrate exceptional accuracy of 99.18% training, 94.28% validation, and 96.8% testing.

Automatic Hate Detection of text was performed on social media data by Shivangi Modi, et. al. [15] In this paper different techniques of automatic hate speech detection and their comparisons are discussed.

Approaches based on Bag of Word (BOW) techniques have a few limitations, such as the fact that the semantics of the word are ignored by a BOW model. In comparison to BOW method, paragraph2vec representations are more insightful. The LSF technique is used to identify offensive content and predict a user's likelihood of transmitting offensive content. The LSF technique can easily adapt to all kinds of English composition styles while tolerating informal and incorrectly spelled contents.

B. RESEARCH GAP

One of the paper uses deep learning, lexicon based model and BERT algorithm, a method to help computers understand the ambiguous meaning in text. Another paper provides a solution to classify hate speech as offensive and non-offensive using CNN.

Few papers use simple Machine Learning techniques like Logistics Regression, SVM, Random Forest, CAT boost MLP and Decision tree algorithms to detect hate speech. Finally, few Research papers explores BiLSTM, paragraph2vec, LST framework approach to automatically detect offensive text in social networks. Some researchers have worked on various languages like Vietnamese language, urdu Bengali etc.

RCNN outperforms bi-direction LSTM and modified Bi-LSTM.

III. PROBLEM STATEMENT

In order to stop the spread of harmful content, lessen online harassment, and safeguard people from harm, DL and AI methods can be practiced to identify hate speech. Given a dataset of text data containing examples of hate and non-hate speech, the task is to train an LSTM framework to accurately divide text data as hate or non-hate speech. The evaluation of the hate text identification model will be based on metrics such as recall, F1 score, accuracy, precision. LSTMs are well-suited for these tasks because they can process sequential data and capture the context and meaning of words in a sentence. Proposed system consists of RCNN model to get more accurate results, as it performs faster and accurate text classification for hate speech detection than the modified Bi-LSTM model.

IV. PROPOSED SYSTEM

A. SYSTEM DESIGN AND DATASET

The architecture depicted in Fig. 1, consists of data collection, tokenization, word embedding, training and testing, RCNN, and model evaluation. These are the primary operations carried out by the system. RCNN (Recurrent Convolutional Neural Network) and other DL models like LSTM (Long Short Term Memory) were compared to one another. Many models use neural network layers to process input, with each layer sending a condensed version of the data to the one below it. The accuracy of such predictions can be enhanced and refined by the inclusion of several hidden layers, but a neural network with a single layer can still make approximations. The dataset used in this study is made up of English-language tweets. Twitter is used to gather/collect tweets. The dataset is used for training and testing after data cleaning. Dataset included details like the tweet id, username, and tweets, among other things. Dataset includes of tweets about racism, sexism, and other topics (No hate). The set of data is divided into two sections. These two sections are train and test. The remaining portion of the dataset is used for testing, while some portion is used for training.

Text classification issues include the identification of hate speech. ML algorithms can be used for handling the challenge of classifying many texts, however they have some significant drawbacks: 1. Weak performance on non-linear data—performs poorly when classes overlap. 2. It is slow and takes a long time to handle larger datasets.

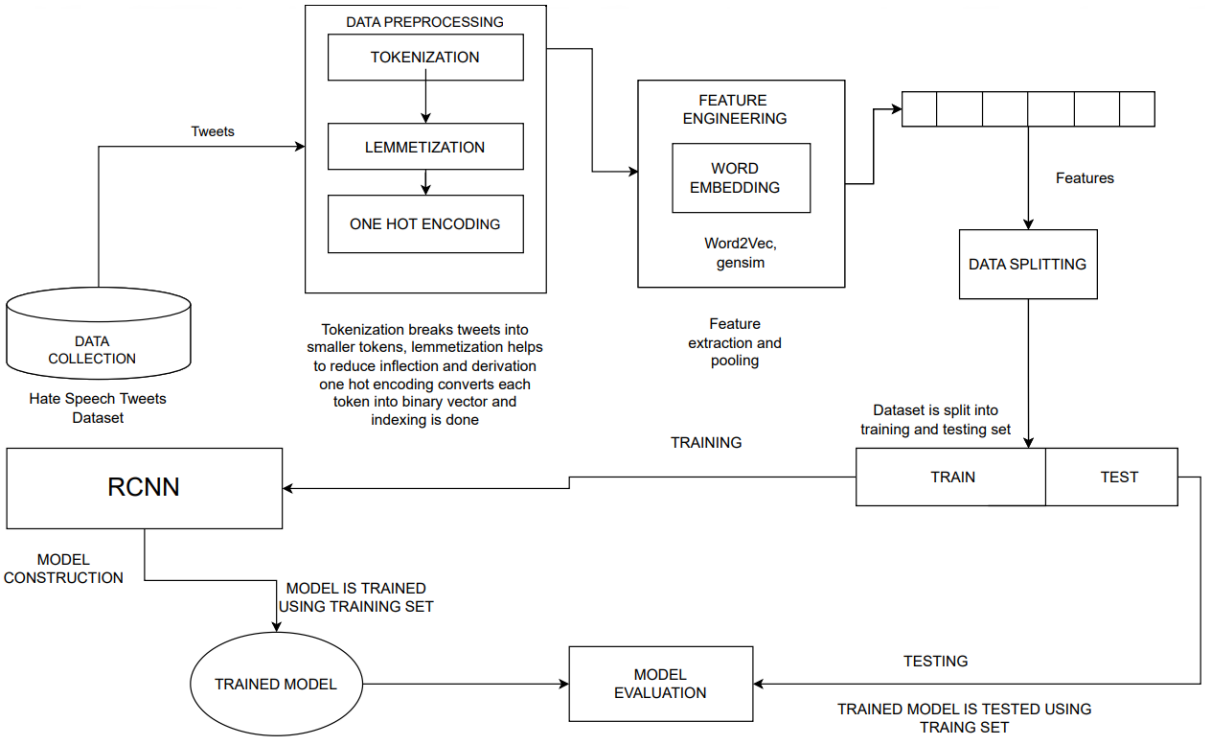

Fig 1. System Architecture

Deep learning models have the following benefit over machine learning models: they use trainable neural systems with multiple hidden layers and a network design to extract the hidden properties from tweets. The steps in the proposed deep learning model are as follows:

- Data collection-

Dataset is a collection of tweets. The data is stored as a csv file and as pickled pandas dataframe. Data file consists of 5 columns, in which there are three classes labelled as Hate speech (0), offensive language (1) and neither (2). The size of dataset is approximately 24000. Data split for training and testing is as follows-

The initial data is split into training and validation sets with a 75:25 ratio using the train_test_split() function. Then, the validation set is further split into validation and testing sets with a 60:40 ratio.

Training set size: 75% of the original data

Validation set size: 15% of the original data

Testing set size: 10% of the original data

- Tokenization-

Division of a lengthy piece of text into tokens is referred as tokenization. In this sense, a token may be a word, a character, or a subword. Tokenizing tweets from the collection into words allows for additional processing. For tokenization, a tokenizer (such as the NLTK tokenizer) is used to split the text into individual words or subwords . After applying the tokenizer, a list of tokens or words for each text data point is created, then use these tokens as input to model. For proposed model, there is need to convert these tokens into numerical representations, such as word embeddings or one-hot vectors.

- Lemmatization-

The use of particular words and their variations in hate speech can be recognized and examined via lemmatization. By breaking down words into their basic forms, it is feasible to spot patterns and trends in the vocabulary used in hate speech across many texts. This information can be used to improve hate speech detection algorithms.

- One-Hot Encoding-

To convert categorical data into numerical data, one-hot encoding divides the column into many columns (e.g., racism, sexism, and none). Depending on which column the data is in, the integers are converted to 1s or 0s. Identify the categories you want to encode. For example, you might have three categories: "Racism", "Sexism ", and "Neither". Next, assign a unique numerical value to each category. For example, assign the values 0, 1, and 2 to the categories: "Racism", "Sexism ", and "Neither. Then, for each text sample, create a one-hot encoded vector that represents the text's category. For example, if a text sample is classified as "Racism", the one-hot encoded vector would look like [1, 0, 0], with the 1 in the first position corresponding to the "hate speech" category. Repeat this process for each text sample in dataset, creating a one-hot encoded vector for each sample.Once you have created one-hot encoded vectors for all of text samples, use them as input to proposed model for training and classification.

- Word embedding-

Word mapping to real-number vectors is known as word embedding. Word embedding can record a word's relationship to other words, its context in a document, and its semantic and syntactic similarities.

The twitter matrix is created through word embedding, which maps the tweet's word. Word2vec and Gensim are used in word embedding. Tweet text is transformed into a numerical

vector shape using the embedding vector "Word2vec" programme.

- RCNN-

When it comes to feature extraction and sequence modelling, Recurrent Convolutional Neural Networks (RCNNs) combine the strengths of both recurrent neural networks (RNNs) and convolutional neural networks (CNNs). It has been applied to numerous natural language processing (NLP) tasks, such as the detection of hate speech.

B. ALGORITHM AND DESIGN

1. LSTM

For dealing with sequential data and preserving the context of lengthy sequences, the LSTM model performs well. [5], [6]. LSTM is a popular framework in deep learning for sequence modeling and prediction tasks which can be used for hate text classification as well. Here's LSTM architecture for hate speech detection:

- Input Layer: The input layer takes the input text data. Each text data is represented as a order of words or characters, and each word or is represented by a vector.
- Embedding Layer: The embedding layer maps each word or character vector to a dense vector of fixed dimension. This layer helps the model to learn semantic relationships between words or characters.
- LSTM Layer: The LSTM layer takes the embedded input sequence and processes it sequentially. The LSTM layer keeps track of the relevant data from the preceding words or characters in the input sequence in a cell state and a hidden state.
- Dropout Layer: The dropout layer is used to regularize the model and prevent overfitting.
- Dense Layer: To identify the input text as sexism, racism, or not hate speech, the dense layer uses the output of the LSTM layer and applies a fully connected layer with a ReLU activation function.
- Output Layer: It outputs the predicted class of the input text.

Training the model involves feeding it with a large dataset of text data labeled as hate or not hate speech. The framework trains to differentiate between the two classes by minimizing a loss function, such as binary cross-entropy, using gradient descent optimization. The effectiveness of the framework is assessed using text data that was not used in training the model. Fig 2 shows the LSTM architecture and its different layers.

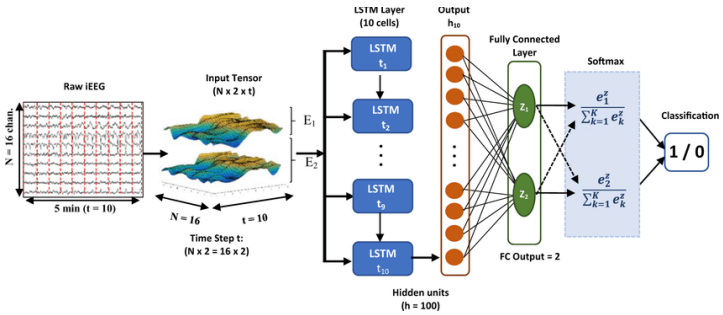

Fig 2. LSTM

2. Bi- Directional LSTM

Recurrent neural networks (RNNs) of the bidirectional long short-term memory (Bi-LSTM) type can be used to identify hate speech in natural language processing (NLP). In order to detect hate speech that may be spread out over numerous words, the Bi-LSTM architecture considers both the past and future context of a sequence of words.

Here's an architecture of a Bi-LSTM network for hate speech detection:

- Input Layer: The input layer consists of a sequence of words or tokens that are embedded into a dense vector representation using techniques such as word2vec.
- Bidirectional LSTM Memory Layer: One LSTM layer of the Bi-LSTM layer reads the sequence from left to right (forward LSTM), and the other reads it from right to left (backward LSTM). Each LSTM layer has a collection of LSTM cells that are appropriate for processing sequences since they can hold onto information throughout time.
- Dropout Layer: This layer is often attached after the Bi-LSTM layer to prevent overfitting by randomly dropping out some of the neurons in the network during training.
- Dense Layer: The result from the Bi-LSTM layer is fed into a dense layer, which applies a non-linear activation function to produce a probability score for each class (e.g., hate vs non-hate speech).
- Output Layer: The only neuron in this layer has a sigmoid activation function, which results in a binary output indicating whether or not there is hate speech.

By contrasting the anticipated output with the actual label, the Bi-LSTM network learns to alter the weights and biases of its neurons to minimise the loss function (such as binary cross-entropy).Once the network is trained, it can be used to classify new sequences of text as sexism or racism or non-hate speech. Fig 3 shows the Bi-LSTM architecture and its different layers.

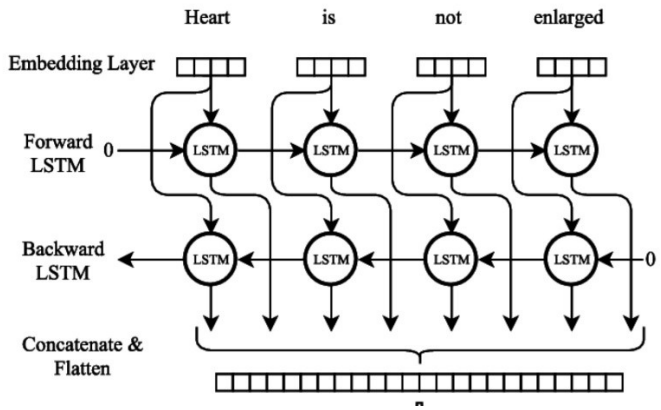

Fig 3. Bi-LSTM

3. Modified Bi-LSTM

An extra embedding layer, a dense layer, a layer of Bi-LSTM, and an activation function are added to the existing Bi-LSTM model in the proposed system.

A Modified Bi-LSTM network's architecture for detecting hate speech is shown here:

The input layer receives the text data that has been processed as input. The input sequence's texts are each represented as a numerical vector.

A dense vector space is created by mapping the numerical vectors from the input layer to the embedding layer. This aids the model's ability to represent the semantic connections between the words.

Bi-LSTM Layers: The Bi-LSTM layers are composed of two LSTM layers, one for forward processing and the other for backward processing of the input sequence. The forward LSTM layer processes the input sequence beginning to end whereas the reverse LSTM layer processes it ending to beginning.

Dropout Layer: To prevent overfitting, this layer is inserted. During training, it randomly removes a portion of the network's units, which serves to lessen the model's sensitivity to particular input features.

Fully Connected Layer: This layer provides a linear modification to the output from the Bi-LSTM layers to create the final output vector.

Output Layer: This layer determines whether or not the input text contains any sexism, racism, or both. It generates a probability value depending on whether the category is sexism, racism, or none using a leakyReLU activation function.

Loss Function: The loss function is used to calculate the difference between the output that was expected and the output that was actually produced. The binary cross-entropy loss function is commonly used for binary classification issues like the detection of hate speech.

Optimization: In order to minimise the loss function, the optimization algorithm is employed to update the framework's weight. It is common practise to train Bi-LSTM models using the Adam optimizer.

Training: A labelled dataset of texts expressing hate and non-hate speech is used to train the model. By modifying the weights of the model using backpropagation, the objective is to minimise the loss function.

Testing: A different test dataset is used to evaluate the trained model's performance. Performance is widely measured using metrics like F1-score, recall, accuracy, and precision.

The layers that are added to the current Bi-LSTM model are shown in figure 4. Leakyrelu activation function, dense layer, and one additional embedding layer are included in this model to determine a probability score for each class. The layers that are added to the current Bi-LSTM model are shown in Fig. 4. It performs better and produces better results as a result than the current model.

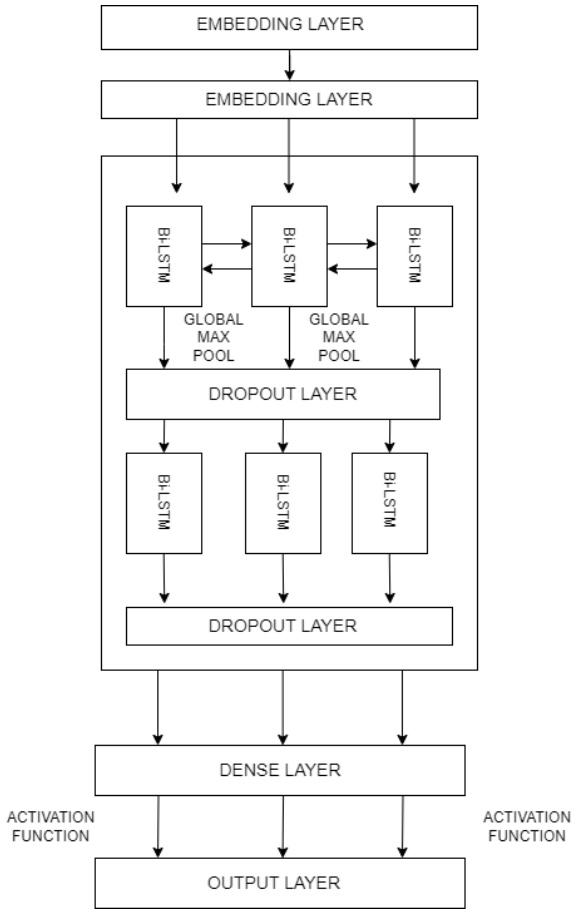

Fig 4. Modified Bi-LSTM

4. RCNN

When it comes to feature extraction and sequence modelling, Recurrent Convolutional Neural Networks (RCNNs) combine the strengths of both recurrent neural networks (RNNs) and Convolutional Neural Networks (CNNs). It has been applied to numerous Natural Language Processing (NLP) tasks, such as the detection of hate speech.

An RCNN's architecture for detecting hate speech typically consists of the following elements:

Text data is entered into the RCNN's input layer as a list of words or characters.

Each word or character in the input sequence is transformed into a high-dimensional vector representation by the embedding layer. This layer aids in capturing the semantic

significance of the words and the links between them in context.

Convolutional Layer: The convolutional layer uses a number of filters to extract features at various degrees of abstraction from the embedded sequences. Each filter applies a convolution over the sequence to produce a feature map that illustrates the presence or absence of a specific feature.

Pooling Layer: The pooling layer aggregates the features that are most pertinent to the downstream job, hence reducing the dimensionality of the feature maps. Max-pooling is a popular pooling method that chooses the highest value from each feature map.

Recurrent Layer: By processing the pooling layer's output sequentially, the recurrent layer takes into consideration the temporal dependencies between the words in the sequence. This layer is in charge of capturing long-term dependencies in the context and modelling it.

The output layer determines the input sequence's final classification, including whether or not it contains hate speech. This layer may be fully linked, followed by a sigmoid or softmax activation function for binary or multiclass classification, respectively.

Flatten layer used after the output layer, essentially convert the output tensor of the fully connected layer into a one-dimensional vector, which would discard the spatial and temporal information of the data. This is detrimental to the performance of the model, as the spatial and temporal information may be important in making accurate predictions, especially in tasks such as natural language processing.

In conclusion, an RCNN is a good architecture for detecting hate speech because it combines the advantages of CNNs and RNNs to efficiently capture the spatial and temporal aspects of the input text data.

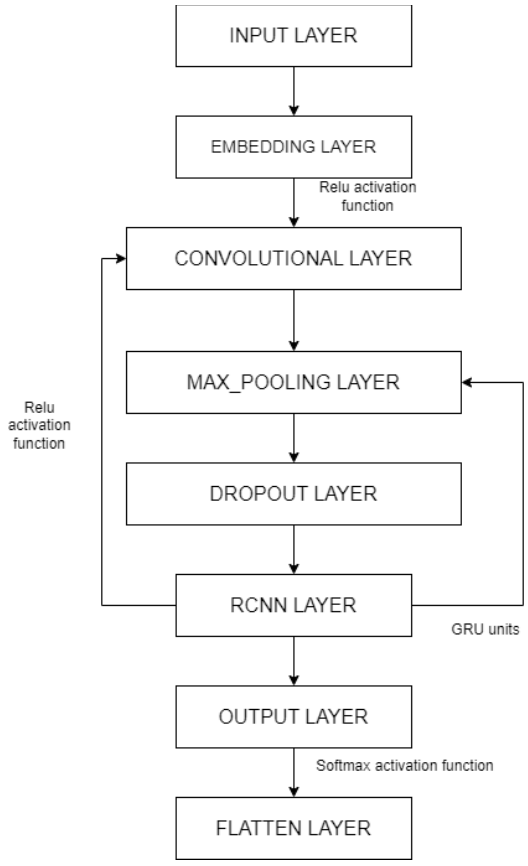

Fig 5. RCNN

V. EVALUATION TECHNIQUES

A hate speech detection model's evaluation process can be similar to that of any classification model. The following are some typical evaluation criteria and methods applied to hate speech detection models:

1. Accuracy: To calculate the accuracy based on a validation split, compare the predicted labels with the actual labels of the validation set. Here are the steps to calculate accuracy of trained model [11].

i. Use trained model to make predictions on the validation set.
ii. Compare the predicted labels with the actual labels of the validation set.
iii. Calculate the accuracy as the number of correct predictions divided by the total number of predictions.

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

2. Precision and Recall: Precision evaluates the percentage of correctly categorised instances among those that are predicted to be positive, whereas recall measures the percentage of correctly classified cases among all actually positive occurrences [11].

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

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

3. F1-Score: This weighted harmonic mean of recall and precision is frequently used as a single summary metric to evaluate a model's performance.

$$F1\ Score = 2 * Recall * \frac{Precision}{Precision + Recall} \quad (4)$$

The dataset's quality and any potential biases in the data should also be considered in addition to these parameters. A representative and varied dataset with a variety of hate speech kinds. To prevent adding biases to the model, the dataset must to be labelled appropriately and consistently. Finally, in order to prevent sustaining or increasing pre-existing biases in the data, the model's output should be carefully examined and analysed. Overall, assessing a hate speech detection algorithm necessitates the use of both quantitative metrics and qualitative analysis, as well as careful consideration of the biases and quality of the data

VI. RESULTS AND ANALYSIS

The following findings were acquired after conducting experiments on two different models. Table I and Table II presents the results after evaluation of model.

Table I. EVALUATION RESULT

Frameworks	Recall	F1-score	Precision
LSTM	0.9210	0.8388	0.8652
Bi-LSTM	0.9237	0.8901	0.8588
Modified Bi-LSTM	**0.9268**	**0.8916**	**0.8600**
RCNN	**0.9424**	**0.9000**	**0.8800**

Table II. ACCURACY

Model	Accuracy
LSTM	0.9210
Bi-LSTM	0.9237
Modified Bi-LSTM	**0.9283**
RCNN	**0.9424**

From the Tables I and II it can be seen that the proposed modified Bi-LSTM and RCNN produces very good results and outperforms standard LSTM and Bi-LSTM in almost all of the evaluation parameters. But between modified Bi-LSTM and RCNN, RCNN gives best results. Modified Bi-LSTM has an accuracy score of 92.83 % and RCNN has an accuracy score of 94.24 %, so according to the outcomes RCNN is better for hate speech detection.

RCNNs are particularly effective because they can capture both the sequential and spatial features of text. This is because RCNNs use RNNs to capture the sequential dependencies within a text and CNNs to capture local features of the text, such as word or n-gram level patterns, this fact reflect in accuracy score.

VII. CONCLUSION, LIMITATIONS AND FUTURE SCOPE

Bi-LSTM and LSTM are both effective DL models for offensive/ hate text detection. The proposed system has been contrasted with Vanilla LSTM and Bidirectional LSTM in order to demonstrate its effectiveness. Bi-LSTM tends to outperform LSTM by F1 score of 0.89. Comparing Bi-LSTM, Modified Bi-LSTM and RCNN, RCNN give better results. The accuracy score of 0.9424 and tends to work well among all the frameworks used.

Many different machine learning or deep learning methods can be explored in order make the Hate speech detection systems more accurate. In future the existing models can use Recurrent convolutional neural network for faster text categorization and detect the hate text in form of sarcasm and irony. These limitations can be taken into consideration when developing the system. The current data has imbalanced class distribution. The framework can be modified by increasing the dataset and balancing the classes.

REFERENCES

[1] Wang Chih-Chien, Min-Yuh Day, and Chun-Lian Wu. "Political Hate Speech Detection and Lexicon Building: A Study in Taiwan." IEEE Access 10 (2022): 44337-44346.

[2] Akhter Muhammad Pervez, Zheng Jiangbin, Irfan Raza Naqvi, Mohammed Abdelmajeed, and Muhammad Tariq Sadiq. "Automatic detection of offensive language for urdu and roman urdu." IEEE Access 8 (2020): 91213-91226.

[3] Watanabe Hajime, Mondher Bouazizi, and Tomoaki Ohtsuki. "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection." IEEE access 6 (2018): 13825-13835.

[4] Zhou Yanling, Yanyan Yang, Han Liu, Xiufeng Liu, and Nick Savage. "Deep learning based fusion approach for hate speech detection." IEEE Access 8 (2020): 128923-128929. [5] Roy, Pradeep Kumar, et al. "A framework for hate speech detection using deep convolutional neural network." IEEE Access 8 (2020): 204951-204962.

[5] Roy Pradeep Kumar, Asis Kumar Tripathy, Tapan Kumar Das, and Xiao-Zhi Gao. "A framework for hate speech detection using deep convolutional neural network." IEEE Access 8 (2020): 204951-204962.

[6] Khan Shakir, Ashraf Kamal, Mohd Fazil, Mohammed Ali Alshara, Vineet Kumar Sejwal, Reemiah Muneer Alotaibi, Abdul Rauf Baig, and Salihah Alqahtani. "HCovBi-caps: Hate speech detection using convolutional and Bi-directional gated recurrent unit with Capsule network." IEEE Access 10 (2022): 7881-7894.

[7] Mozafari Marzieh, Reza Farahbakhsh, and Noel Crespi. "Cross-lingual few-shot hate speech and offensive language detection using meta learning." IEEE Access 10 (2022): 14880-14896.

[8] Plaza-Del-Arco, Flor Miriam, M. Dolores Molina-González, L. Alfonso Ureña-López, and María Teresa Martín-Valdivia. "A multi-task learning approach to hate speech detection leveraging sentiment analysis." IEEE Access 9 (2021): 112478-112489.

[9] Qureshi Khubaib Ahmed, and Muhammad Sabih. "Un-compromised credibility: Social media based multi-class hate speech classification for text." IEEE Access 9 (2021): 109465-109477.

[10] Rodriguez Axel, Yi-Ling Chen, and Carlos Argueta. "FADOHS: framework for detection and integration of unstructured data of hate speech on facebook using sentiment and emotion analysis." IEEE Access 10 (2022): 22400-22419.

[11] Saksesi Arum Sucia, Muhammad Nasrun, and Casi Setianingsih. "Analysis text of hate speech detection using recurrent neural network." In 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), pp. 242-248. IEEE, 2018.

[12] Abro Sindhu, Sarang Shaikh, Zahid Hussain Khand, Ali Zafar, Sajid Khan, and Ghulam Mujtaba. "Automatic hate speech detection using machine learning: A comparative study." International Journal of Advanced Computer Science and Applications 11, no. 8 (2020).

[13] Şahi Havvanur, Yasemin Kılıç, and Rahime Belen Sağlam. "Automated detection of hate speech towards woman on Twitter." In 2018 3rd international conference on computer science and engineering (UBMK), pp. 533-536. IEEE, 2018.

[14] Susanty Meredita, Ahmad Fauzan Rahman, Muhammad Dzaky Normansyah, and Ade Irawan. "Offensive language detection using artificial neural network." In 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), pp. 350-353. IEEE, 2019.

[15] Modi Shivangi. "AHTDT-Automatic Hate Text Detection Techniques in Social Media." In 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), pp. 1-3. IEEE, 2018.

[16] Tatwadarshi P Nagarhalli, Vinod Vaze, NK Rana, "Impact of machine learning in natural language processing: A review", IEEE third international conference on intelligent communication technologies and virtual mobile networks (ICICV), 2021, pp. 1529-1534.

[17] Tatwadarshi P Nagarhalli, Sneha Mhatre, Sanket Patil, Prafulla Patil, "The Review of Natural Language Processing Applications with Emphasis on Machine Learning Implementations", IEEE International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 1353-1358.

[18] Tatwadarshi P. Nagarhalli, Ashwini Save, and Narendra Shekokar, "Fundamental Models in Machine Learning and Deep Learning", Design of Intelligent Applications using Machine Learning and Deep Learning Techniques, edited By Ramchandra Sharad Mangrulkar, Antonis Michalas, Narendra Shekokar, Meera Narvekar, Pallavi Vijay Chavan, Chapman and Hall/CRC, 2021.

[19] Tatwadarshi P Nagarhalli, Vinod Vaze, NK Rana, "A Novel Framework for Neural Machine Translation of Indian-English Languages", IEEE International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 676-682.

[20] Raj Vyas, Kirti Joshi, Hitesh Sutar, Tatwadarshi P Nagarhalli, "Real time machine translation system for english to indian language", IEEE 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 838-842.

[21] Ved Kokane, Prasad Nijai, Vikas Jamge, Tatwadarshi P Nagarhalli, "Speech Emotion Recognition using Convolutional Neural Networks and Long Short-Term Memory", IEEE 6th International Conference on Trends in Electronics and Informatics (ICOEI), 2022, pp. 1-8.