https://doi.org/10.1007/s42979-022-01308-5

ORIGINAL RESEARCH

An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques

Mitushi Raj[1] · Samridhi Singh[1] · Kanishka Solanki[1] · Ramani Selvanambi[1]

Received: 25 April 2022 / Accepted: 1 July 2022 / Published online: 26 July 2022

Abstract

Nowadays, a lot of people indulge themselves in the world of social media. With the current pandemic scenario, this engagement has only increased as people often rely on social media platforms to express their emotions, find comfort, find like-minded individuals, and form communities. With this extensive use of social media comes many downsides and one of the downsides is cyberbully. Cyberbullying is a form of online harassment that is both unsettling and troubling. It can take many forms, but the most common is a textual format. Cyberbullying is common on social media, and people often end up in a mental breakdown state instead of taking action against the bully. On the majority of social networks, automated detection of these situations necessitates the use of intelligent systems. We have proposed a cyberbullying detection system to address this issue. In this work, we proposed a deep learning framework that will evaluate real-time twitter tweets or social media posts as well as correctly identify any cyberbullying content in them. Recent studies has shown that deep neural network-based approaches are more effective than conventional techniques at detecting cyberbullying texts. Additionally, our application can recognise cyberbullying posts which were written in English, Hindi, and Hinglish (Multilingual data).

Keywords Cyberbullying · Stack word embeddings · Deep learning model · Multilingual · Real-time tweets

Introduction

Many pieces of research work that are done in this area using various machine learning and deep learning techniques have yielded significant results in detecting and preventing cyberbully. However, most works have included mostly English data for training and testing purposes, while a few included native languages like Bangla, Arabic, and Urdu. As there is little to no work done in aiding the situation of increased cyberbullying in a country like India where most Hindi speaking people use English text, comprising of Hindi words written in Latin script, and many people using Hindi text written in Devanagari script, we plan to proceed to combat this problem by incorporating such data into our suggested learning algorithm so that cyberbullying can be detected in real-time tweets.

Data have been collected from three sources and then combined. One contains English texts, the other contains Hindi texts, and the last one contains a combination of Hindi and English texts. As we have acquired these three datasets from different sources, compiling them in their original form will not be compatible because of the difference in classification labels. To proceed with these datasets, we will first have to adopt one single classification technique, and for this purpose, 0-1 classifier was used, which will tell us if the text contains content of cyberbullying or not, thus making it a black and white area to train our model and eliminating any gray possibilities. Data cleaning is essential before classification to remove the symbols, spacy tokenizer URLs, emails, stopwords, white space, numbers, punctuation, stemming, lemmatization, and single tokens.

This article is part of the topical collection "Predictive Artificial Intelligence for Cyber Security and Privacy" guest edited by Hardik A. Gohel, S. Margret Anouncia and Anthoniraj Amalanathan.

✉ Ramani Selvanambi
ramani.s@vit.ac.in

Mitushi Raj
mitushiraj170@gmail.com

Samridhi Singh
samridhisingh270@gmail.com

Kanishka Solanki
kanishka.solanki2018@gmail.com

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamilnadu 632014, India

Word embedding is an natural language processing methodology for mapping words or phrases from a lexicon to a corresponding vector of real numbers, which is then used to find word predictions and semantics. These word embeddings are beneficial when training on data from contextual information because they indicate implicit associations between words. The main reason for choosing word embedding is that they do not require costly annotation and can be produced from vast, unannotated corpora that are freely available. After that, pre-trained embeddings can be used in tasks that need tiny amounts of labeled data.

In the collection of word embeddings, stacked embeddings are one of the most important topics. To combine distinct embeddings, a stack embedding method is utilized. A stack embedding strategy, for instance, is utilized to use both conventional and contextual string embeddings. Stack embeddings allow to mix and match and the combination of embeddings has been established to produce the best results [1].

Deep learning (DL) is a subset of Machine Learning (ML) that can be applied to a wide range of applications. Deep learning models are preferred in our work over traditional machine learning models because they have been shown to be more successful than machine learning (ML) or statistical methods and have a higher number of neuron layers than ML. Deep learning algorithms are proven to be highly effective at text classification, with state-of-the-art outcomes on a variety of classic academic benchmark issues. Deep learning networks have the advantage of improving their models as the size of the data grows. Because hybrid techniques have been demonstrated to be potential models for minimizing sentiment inaccuracies on more detailed training data, we explored utilizing a hybrid model instead of a single layer model [2].

On all types of datasets, hybrid models outperformed single models in terms of cyberbullying detection, especially when deep learning methods were combined. Our research looks at how our proposed hybrid model reacts to various forms obtained from multiple languages. The combination of multiple models was studied and validated in this study. We looked at the relationship between models and their increased abilities to extract traits, store past data and nodes, and classify text [3].

On multilingual Twitter datasets, we will use the CNN-BiLSTM model with stacked word embedding. The combined models boosted the accuracy of cyberbullying detection, according to the findings of our research. After training the model, we plan to create a Twitter-like app using python and the web tech stack [4], in which real-time data will be fed into the system, and whether the input text is cyberbullying or non-cyberbullying will be predicted.

This paper is organized as follows. The following section provides the aim and motivation of this research work. The subsequent section provides the highlights or the contribution made in this work. "Literature Review" provides an incisive review on the related works on existing cyberbully detection techniques based on machine leaning and deep learning, and the subsequent section provides the detailed explanation of the proposed model. "Results and Discussion" describes the results obtained and the last section is concluding in nature.

Aim and Motivation

During the time of COVID-19, people's activity on social media has increased tremendously, since social media platforms provide us with excellent communication platforms, but it also makes young people more vulnerable to online threats and harassment. Because of many active users on social media networks, cyberbullying has become a global problem. The pattern suggests that cyberbullying on social media is on the rise and according to recent studies, it is becoming more prevalent among teenagers. The ability to recognize potentially dangerous communications is critical to successful prevention, and the information overload on the Internet needs intelligent systems that can automatically identify potential threats. This research work aims to develop a model that can accurately detect cyberbully in real-time tweets.

Cyberbullying has been one of the main prevailing issues since the use of technology and internet services has been made accessible to the general public. After the COVID-19 pandemic hit, affecting the daily life of most people and forcing them to isolate themselves from social groups and communities, this usage of social media has only been increasing. With the increased use of digital platforms for educational reasons by most youths, an increase in cyberbully incidents seems unavoidable, as pupils who are bullied are more inclined to cyberbully. Bullying and discrimination based on race, religion, sex, caste, and creed can cause an adverse effect on the victim's mental health, eventually leading to anxiety, depression, and even an increase in suicide cases.

Cyberbullying has increased by 70% in just a few months, according to a group that records online bullying and harassment cases. Human monitoring becomes fruitless in the face of such huge amounts of data on the internet, as there is no scalable and effective means to trace out cyberbullies and tackle the problem. Thus, there is a need to tackle this problem on an automated level that is fast, efficient, and accurate for the sake of social well-being.

Highlights

The objective of this work is to build a CNN-BiLSTM deep learning detection model that can detect cyberbullying content in tweets posted by users in three different languages in real time data. Subsequently we have also developed a website much like a social media platform to portray the applicability of this model.

- A multilingual dataset is built, apart from English(Latin script) and Hindi(Devanagari script), Hinglish (Latin script- where Hindi words are written using English alphabets) is also included, which makes up most of the tweets from users who tweet in Hindi.
- A CNN-BILSTM model is proposed for cyberbullying detection because an ensemble deep learning model with multiple layers outperforms single-layer neural network models. To optimize the model even further, stacking of two-word embeddings (Glove + fastext) will be done to enhance the model's performance.
- This proposed model works for on real-time data and the web portal that is created will be like a clone of social media websites like Twitter, where one can post their tweets, and the change will be reflected in the feed. This site will assess if the posted tweet contains cyberbullying content by running it through our model.

Literature Review

Most research papers have extracted data from a single source and done a comparative study on various machine learning or deep learning techniques in combination with different word vectors or feature extraction techniques and drawn out the best combination. Only a handful of research was found where the work focused on optimizing the detection model by either building ensemble ml models or layering up different feature preprocessing techniques. Even in those researches, they focused on testing the model on the dataset and there was no real time detection involved. Most works done in this area have included mostly English data, while a few included native languages like Bangla, Arabic, and Urdu.

OCDD (Optimized Twitter Cyberbullying Detection based on Deep Learning) technique was used by [1] , an innovative solution to feature extraction difficulties. OCDD depicts a tweet as a series of word vectors rather than extracting features from tweets and feeding them to a classifier. Deep learning will be employed in the classification phase, together with a metaheuristic optimization technique for parameter adjustment. Using deep neural networks and word embeddings, methodology was proposed by [2] for detecting cyberbullying messages in text data. The classifier's performance is improved by stacking Bert and Glove embeddings together. As a result, the model outperforms most classic machine learning approaches, including Support Vector Machine and Logistic Regression. A single and double ensenble-based voting model was created by [3] that can divide items into two categories: offensive and non-offensive. On a dataset retrieved from Twitter, several machine learning classifiers, three ensemble models, two feature extraction algorithms, and countless n-gram evaluations were chosen. Logistic Regression and Bagging Ensemble Model Classifiers were shown to be the most effective in detecting cyberbullying in the study, however their proposed SLE and DLE voting classifiers outperformed them.

Substantial preprocessing was performed by [4] on Roman Urdu micro text, including the creation of a Roman Urdu slang-phrase dictionary and the mapping of slangs following tokenization. The unstructured data was then processed further to deal with encoded text formats and metadata/non-linguistic elements. Extensive tests using RNN-LSTM, RNN-BiLSTM, and CNN models were undertaken after the preprocessing stage. To give the comparison analysis, the performance and accuracy of models were assessed using several metrics. On Roman Urdu text, RNN-LSTM and RNN-BiLSTM performed best. For cyberbully detection, a BiGRU-CNN sentiment classification model was presented by [5] which consists of a BiGRU layer, attention mechanism layer, CNN layer, complete connection layer, and classification layer. The attention mechanism layer has a firmer grasp of representative words and can better allocate weight to them. To train and test the model, the Kaggle text data set is used, as well as the emoji data set gathered from social media. The model's classification accuracy is higher than that of the traditional model, according to the findings.

A pretrained BERT model was used by [6] which is built on a novel deep learning network with the technique of transformer to detect cyberbullying on social media platforms. For classification, the model employs a single linear layer of a neural network, which can be substituted by deep learning network models like CNN and RNN. The model has undergone extensive training based on two social media datasets, one of which is public. The first dataset is of small size (Formspring dataset), whereas the second is of greater size (Wikipedia dataset). The model produces better and more consistent results for the latter without the requirement for oversampling.

To detect cyberbullying in Bangla and Romanized Bangla literature, as well as to give a comparison of the two systems, Machine learning and deep learning algorithms were used by [7]. The detection was carried out using 5000 Bangla and 7000 Romanized Bangla texts, as well as a combined dataset. Before they were trained, the datasets were preprocessed with NLP techniques and features retrieved. Multinomial Naive Bayes, SVM, Logistic Regression, and XGBoost were among the machine learning models utilized. Deep Learning algorithms including CNN, LSTM, BLSTM, and GRU were used. According to the findings, CNN outperformed all other algorithms for the dataset containing Bangla texts, with an accuracy of 84%. In the other two datasets, the Multinomial Naive Bayes machine learning technique fared best, with 84% accuracy in the Romanized Bangla dataset and 80 percent accuracy in the combined dataset.

Using data from Twitter, Wikipedia, and Formspring, a working implementation of an application that detects cyberbullying across multiple social media platforms, was proposed by [8]. They have used LSTM layers to detect cyberbully. Using the backpropagation method, these models were trained. The cross-entropy loss function is used in combination with the Adam optimizer. These results were better than the traditional approaches.

Four machine learning models: LR, Gaussian Naive Bayes, RNN, and BERT were used by [9]. For final classification, these are combined with a neural network. The suggested model's and other accessory classification models' suitability was assessed using numerous assessment measures to determine how well the model can perform. Accuracy, Precision, and Recall were the most commonly utilized measures for measuring efficacy. The model works well regardless of the social media sentence input, according to the results of training and testing. Though the models' classification accuracy is good, they do have significant limits that might be overcome by including a variety of additional techniques. The effectiveness and efficacy of deep learning systems in detecting cyberbullying were discussed by [10]. They worked on four deep learning models: Bidirectional Long Short-Term Memory (BLSTM), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN). In comparison to the RNN, LSTM, and GRU models, the BLSTM model achieved good accuracy and F1-measure scores.

A cyberbullying detection framework was developed by [11] using reinforcement learning and combining various natural language processing techniques. The developed framework leverages human-like behavioral patterns, uses delayed rewards, and outperforms other models with a highly dynamic and populated dataset achieving 89.5% accuracy on the dataset. Different machine learning (Logistic Regression, Linear SVC, Multinomial Naive Bayes, and Bernoulli Naive Bayes) and deep learning techniques (CNN models incorporating) were applied by [12], by incorporating various n-gram ranges to detect cyberbullying on Twitter. Vector approaches included distributed bag of words and distributed memory, further extending to distributed memory concatenated and distributed memory means.

A neural network architecture with a self-attention model trained on a balanced dataset combining three different datasets from different sources was adopted by [13]. The self-attention model follows an overall standard encoder-decoder architecture that replaces recurrent layers with multi-headed self-attention and is tested on parameters like precision, recall, and F1 scores achieved state-of-the-art accuracy and even outperformed the BLSTM model with attention. An ensemble model involving feature analysis techniques was developed by [14] for Naive Bayes-SVM and XGBoost models and word embeddings for deep learning approaches like CNN, Bi-GRU, and attention networks using a majority voting-based ensemble where the prediction from individual models counts as votes for class labels with a fairly based dataset.

A classification model based on the attention techniques was proposed by [15] to analyze Arabic comments, including different Arabic dialects. Inspired by human-like learning, the proposed attention model dynamically pays attention to certain parts of the input that aids in achieving results and ignores everything irrelevant. The model is built with an embedding layer consisting of two LSTM layers with a dense layer and an output layer to compare impact with and without the recurrent networks.

Many hybrid approaches to the test on a variety of datasets from various disciplines were put by [16] to detect sentimental analysis. Eight textual tweets and review datasets from various fields are used to develop and test hybrid deep sentiment analysis learning models that integrate Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Support Vector Machine (SVM). Each technique was evaluated based on its dependability and calculation time. The CNN-BiLSTM model outperformed all the baseline deep learning models. A cyberbully detection model was proposed by [17] to improve manual monitoring for cyberbullying on social media. In this paper OCR was used to analyze image character to determine the impact of image-based cyberbullying on an individual basis, which was further tested on a dummy system. To create predictive models for cyberbullying detection, supervised learning-based techniques commonly use classifiers such as SVM and Nave Bayes.

A Deep Convolutional Neural Network (DCNN) was created by [18] to create an automated system as its nearly hard to physically filter any information from such a large amount of incoming traffic in the form of tweets With the use of convolution, the proposed DCNN model uses the Twitter text with GloVe embedding vector to capture the semantics of the tweets and outperformed existing models. a strategy for predicting hateful speech on social media networks using a hybrid of natural language processing and machine learning techniques was described by [19]. A powerful natural language processing optimization ensemble deep learning strategy is used to analyze the collected data (KNLPEDNN). The methodology employs an active learning environment to detect hate speech on social media platforms by classifying the text into neutral, offensive, and hate language.

A comparative study of pre-existing deep learning methods was presented by [20] to detect Hate Speech and Offensive Language in textual data. These methods include CNN, RNN, LSTM and BERT models. The authors also investigated the effect of class weighting technique on the enactment of the deep learning methods. It was found that the pre-trained BERT model outperformed other models in case of both unweighted and weighted classification, likely because of the property of BERT to measure the relation between sentences by treating them as whole units.

A model for detecting tweets that contain racist text was proposed by [21] by performing the sentiment analysis of tweets. A stacked ensemble deep learning model is assembled by combining GRU, CNN and RNN, called Gated Convolutional Recurrent- Neural Networks (GCR-NN). The performance of the model is optimized by setting different structures in terms of the number of layers, loss function, optimizer and number of neurons, etc. Proposed model showed substantially better performance than those of machine learning models. Embedding word representations and deep-learning approaches were employed by [22] to recognize and classify toxic speech. A Twitter corpora was used to conduct binary and multi-class classification, and two main approaches were investigated: extracting word embeddings accompanied by utilizing a DNN classifier and fine-tuning the pre-trained BERT classifier. BERT fine-tuning was found to be substantially more effective.

Numerous approaches with varying data ratios at about the same time were compared by [23]. As a result, when the data is small, machine learning produces good results. When they used more data for the trials, they got better outcomes by employing deep learning. When compared to the other approaches they examined, BiRNN produces the best results. They have used ML and DL models to detect hate speech using RNN. They have two datasets, one has more data than the other one. They have strived to extract hate speech from tweets to discover the best way for improving accuracy in several methods. Result shows that ML works well with small data while DL works well with large datasets. Even if their strategy outperforms previous models, they must consider the sort of data set that will be used in the future.

Efficiency of different neural network models such as CNN, BiLstm, BiLstm with attention mechanisms were combined with CNNLSTM models and evaluated by [24]. They trained these networks on a labeled dataset of YouTube comments. They employed Arabic word representations to depict the remarks after running the dataset through a variety of pre-processing procedures. They also used machine learning optimization techniques to fine tune the neural network model's parameters. They used 5 fold cross validation to train and evaluate each network. The CNN-LSTM had the highest recall of 83.46% followed by CNN than BiLSTM with attention and the last one was BiLSTM.

The problem of detection of hateful comments was solved by [25], using text mining and deep learning models built using LSTM to detect and classify cyberbullying speech and filter it out for us. The input layer has input in the form of sequences which are basically numbers that represent text. The embedding layer takes each word from the input layer and produces appropriate word vectors. This is fed to the LSTM model which is further connected to the dense layer. Each neuron in each layer is strongly linked to the layers above and below it. The final model has accuracy of 94.94% and is able to detect cyberbullying or not.

Proposed Methodology

This proposed model is a prototype for a cyberbullying detection system which can be used for social media platforms for automated checking and control of cyberbully. The data for training is cleansed and preprocessed before being fed into stacked word embeddings. Then the CNN-BiLSTM deep learning model is trained to perform better than regular deep learning models trained standalone. The model is saved for its use in the website. The website is similar to any social media platform where the user has access to many features. Admin will have privileges to view content status. Even though this work is a prototype, it is still a step towards getting a better result.

- Dataset Analysis - The acquired labeled data in 3 languages, i.e., Hindi, English and Hinglish, from numerous open-source dataset sources go towards the text preprocessing stage which involves Data Cleaning, Data Integration, Data Transformation, Data Reduction and Data discretization.
- Data Cleaning - Any irrelevant attributes, empty cells and NaN values are removed. The data is also formatted so that the data type across the dataset is uniform.
- Data Transformation - As the three datasets are acquired from different sources, compiling them in their original form will not be compatible because of the difference in classification labels. To proceed with these datasets, it is important to get rid of different label sets and using one single classification technique, 0-1 classifier, which will tell us if the text contains content of cyberbullying or not, thus making it a black and white area to train our model and eliminating any gray possibilities.

- Data Integration - All the datesets are integrated to one csv file that is used for further text preprocessing.
- Data Discretization - In this stage the data was tokenized, i.e., splitted the sentence into words for easy evaluation of data.
- Data Reduction - In this text preprocessing stage, certain things are removed such urls, special characters, '@' and stopped words from tweets and converted all the text into lower case. Further, stemming is performed, which is transforming a word to its root form, and lemmatization, which reduces the words to a word existing in the language. This stage helps in reducing data into its simplest possible form.

After the preprocessing of data is completed, we move towards the building and training of the model stage. As shown in Fig. 1, for building our CNN-BiLSTM model, Word Embedding approach is used as it solves various issues that the simple one-hot vector encodings have. Most crucial thing is that word embeddings boost generalization and performance. We will stack 2 word embeddings which are GloVe and FastText. A combination of embeddings has been established to produce the best results. After the stacking of word embedding, CNN-BiLSTM model is built. As a hybrid technique has shown the potential of reducing sentimental errors on increasingly complex data. An ensemble ML model is also built, in which feature extraction technique and unigram feature engineering are used. The proposed CNN-BiLSTM model is compared with an ensemble ML model to draw out a comparison on the accuracy.

CNN-BiLSTM Architecture

A single machine learning or deep learning model can predict the outcome rather accurately when applied to specific domains, but each has its own set of advantages and downsides. LSTM usually produces superior results, but it takes longer to process than CNN, and CNN has fewer hyperparameters and requires less supervision. In the meanwhile, the LSTM is more accurate for long sentences but takes longer to analyze. Because RNN has a major gradient loss issue when processing sequences, the perception of nodes in the front decreases as nodes get further back. To tackle the problem of gradient vanishing, BiLSTM is used. It solves the problem of fixed sequence to sequence prediction. RNN has a limitation where both input and output have the same size. So it fails in case of machine translation where input and output have different sizes or case of text summarization where input and output have a different length, which is not the case with BiLSTM. The concept of combining two (or more) methods is offered as a way of implementing the benefits of both while also addressing some of the drawbacks of existing techniques.

A CNN BiLSTM is a bidirectional LSTM and CNN framework that is concatenated. It trains both character-level and word-level characteristics in the initial formulation for classification and prediction. The character-level properties are induced using the CNN layer. To derive a new feature vector using per-character feature vectors such as character embeddings and (preferably) character type, the model includes a convolution and a max pooling layer for each word.

Combining different variation yields multiple hybrid approaches that we have tested:

1. $Glove + Fasttext \longrightarrow CNN \longrightarrow BiGRU \longrightarrow adam$
   $(dense, conv1d = relu; out = sigmoid), maxlen = 25$
2. $Glove + Fasttext \longrightarrow CNN \longrightarrow BiLSTM \longrightarrow adam$
   $(dense, conv1d = relu; out = sigmoid), maxlen = 25$
3. $Glove + Fasttext \longrightarrow BiLSTM \longrightarrow BiGRU \longrightarrow adam$
   $(dense, conv1d = relu; out = sigmoid), maxlen = 25$
4. $Glove + Fasttext \longrightarrow CNN \longrightarrow BiGRU \longrightarrow adam$
   $(dense, conv1d = relu; out = sigmoid), maxlen = 25,$
   $trainable = True$
5. $Glove + Fasttext \longrightarrow CNN \longrightarrow BiLSTM \longrightarrow adam$
   $(dense, conv1d = relu; out = sigmoid), maxlen = 25,$
   $trainable = True$
6. $Glove + Fasttext \longrightarrow BiLSTM \longrightarrow BiGRU \longrightarrow adam$
   $(dense, conv1d = relu; out = sigmoid), maxlen = 25$
   $\longrightarrow Spatialdropout1D, GlobalMaxpooling1D,$
   $GlobalAveragePooling1D$

The CNN-BiLSTM model that is to be used has the following features:

- Stacked Word Embedding: A distributed representation of words where different words that have a similar meaning (based on their usage) also have a similar representation. Two of such word embeddings are glove and fastext and stacking of these two embeddings provide better results
- Convolutional Model: A feature extraction model that learns to extract salient features from documents represented using a word embedding.
- Fully Connected Model: The interpretation of extracted features in terms of a predictive output.

Therefore, the model is comprised of the following elements as shown in Fig. 2:

Input layer t — The length of input sequences is defined by the input layer.

Embedding layer — 100-dimensional real-valued representations and an embedding layer set to the vocabulary's size.

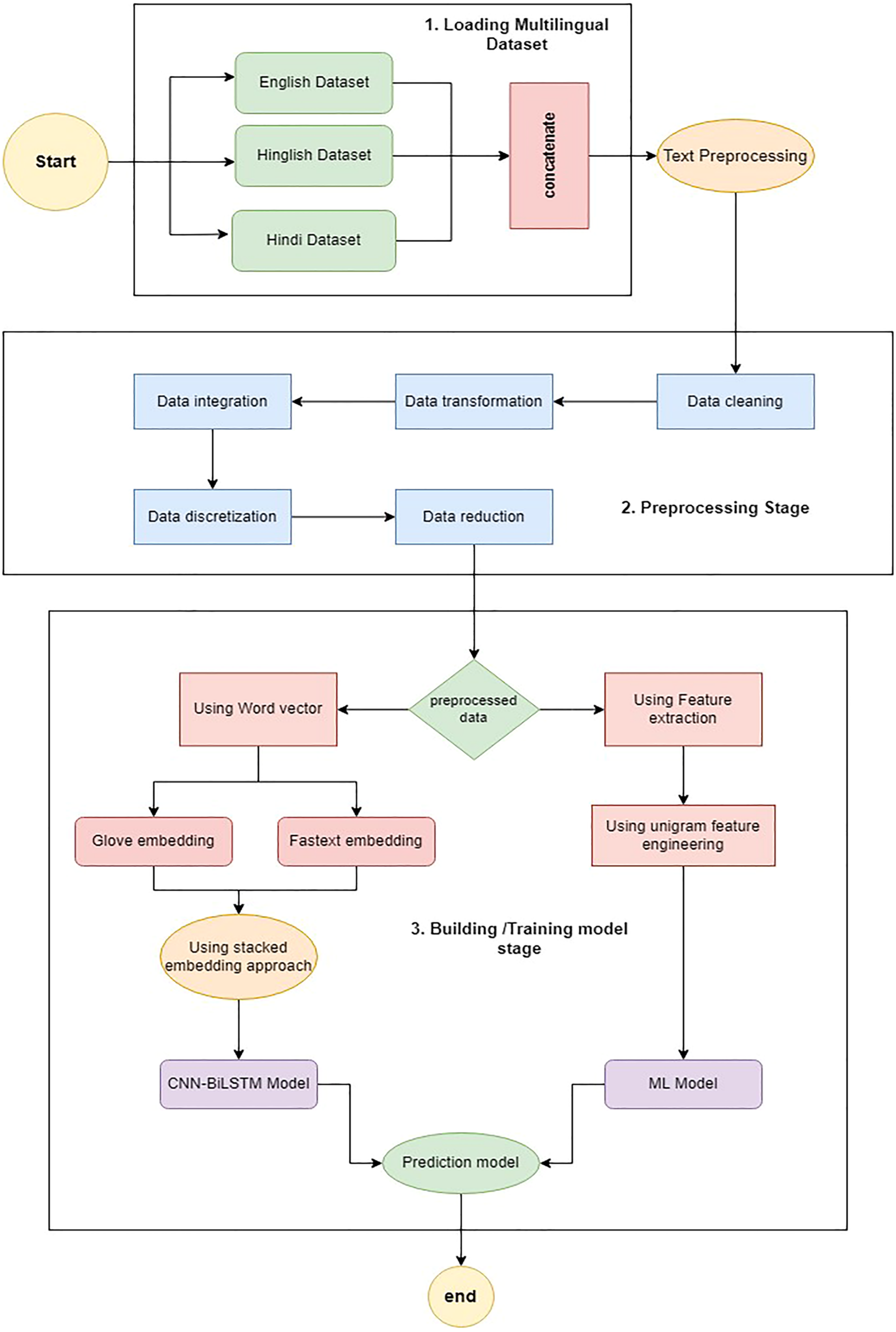


Fig. 1 Proposed model

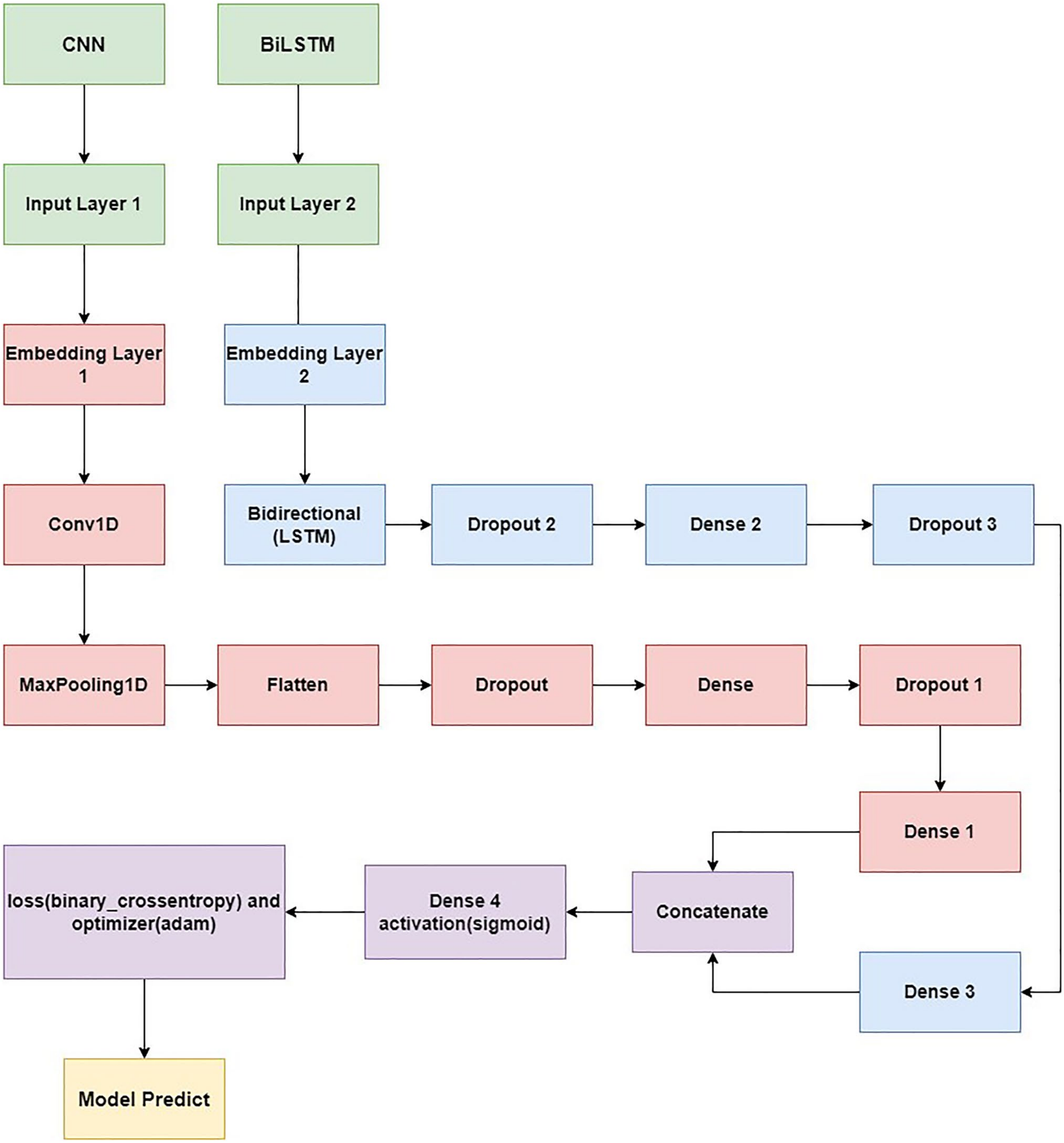


Fig. 2 CNN-BiLSTM architecture

Conv1D layer — Using 32 filters and a kernel size corresponding to the amount of words to read simultaneously.

MaxPooling1D — Merge the result of the convolutional layer with this layer.

Flatten layer — For concatenation and to convert the three-dimensional output to two-dimensional

Transfer function — Rectified Linear.

Kernel sizes— 3
Number of filters— 100
Dropout rate— 0.5
Weight regularization (L2) — 3
Batch Size — 128
Update Rule — Adam

The Adam optimizer is computationally more efficient, requires slight memory, is invariant to diagonal resizing of gradients, and it is well suited for problems with a lot of data/parameters. We will perform the best parameter using grid search and 10-fold cross validation. Now, Convolutional Neural Network (CNN) models are built to classify encoded documents as either cyberbullying or non-cyberbullying. Now, the CNN model can be defined as follows as shown in Fig. 2:

- One Conv layer with 100 filters, kernel size 3, and relu activation function;
- One MaxPool layer with pool size = 2;
- One Dropout layer after flattened;
- Optimizer: Adam
- Loss function: binary cross-entropy (suited for binary classification problem)
- Dropout layers are used to solve the problem of overfitting and bring generalization into the model. As a result, in hidden layers, it's best to keep the dropout value near 0.5.

WebAPP Architecture

As shown in Fig. 3, the web-based system is a social media prototype where users will be able to use the developed project like a social media platform where they can post the tweets, see real time updates of feed and chat with friends. The admin feature of the app allows the admin to trace cyberbullying comments and block the users for a certain amount of time.

Features User Register: The system allows new users to register themselves on the app.

User account: The system allows the user to create their accounts by providing their emails and setting a password. The user can also set a username for their account as well as view their profiles.

Admin account: The system allows admin to have separate login with which they can perform admin related activities.

Posting Feature: The system allows user to post their tweet and tag theirs friends.

View Feed: The system allows user to view their feed and gives admin privileges to see all kind of tweets.

Blocking Feature: The system has special admin features where they allow admin to suspend user accounts if they find their comments cyberbully.

Results and Discussion

The performance comparison of different configurations of activation's and optimizer for a baseline LSTM model on the basis of accuracy is done. Accuracy refers to the proportion of correct predictions made by the model.

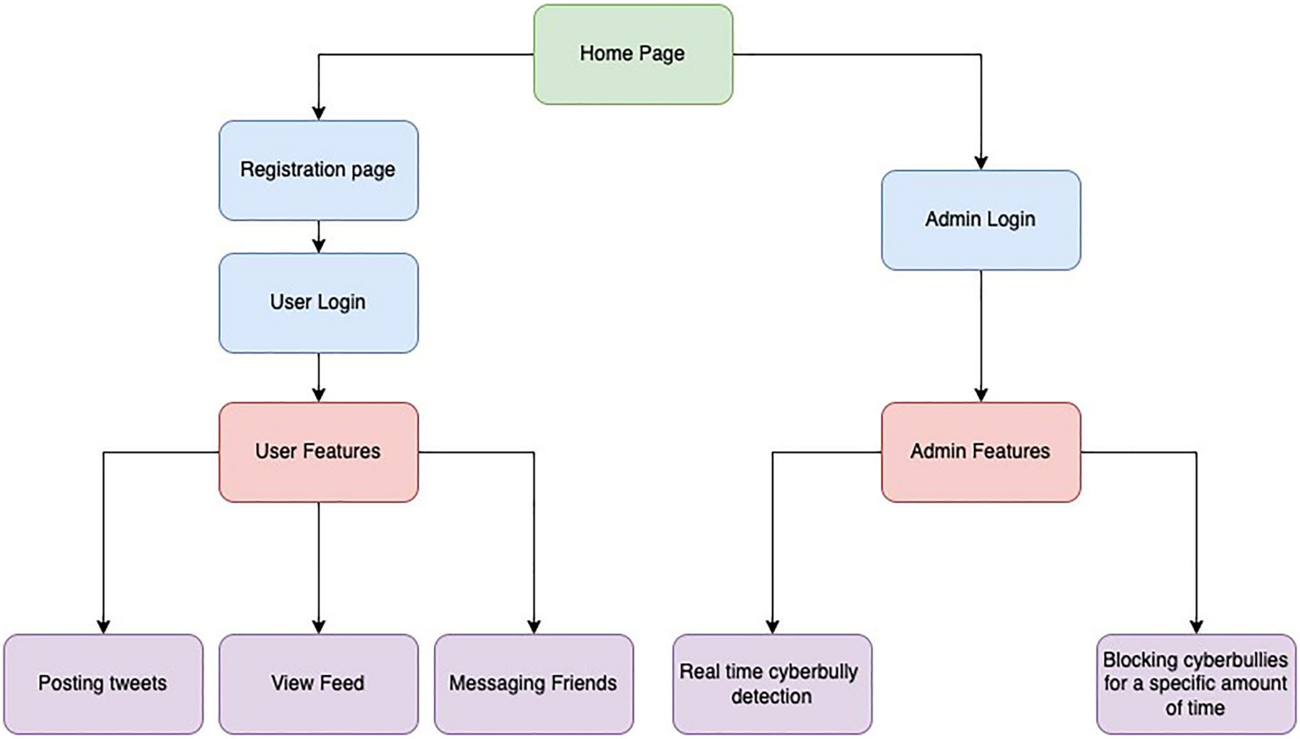


Fig. 3 WebAPP architecture

Table 1 Comparison of activations and optimizer on baseline models

Sl. no	Model Name	Hyperparameter	Accuracy
1	LSTM	Activation-sigmoid; Optimizer-Adam	0.87
2	LSTM	Activation-relu; Optimizer-Adam	0.85
3	LSTM	Activation-sigmoid; Optimizer-RMSProp	0.86
4	LSTM	Activation-relu; Optimizer-RMSProp	0.85

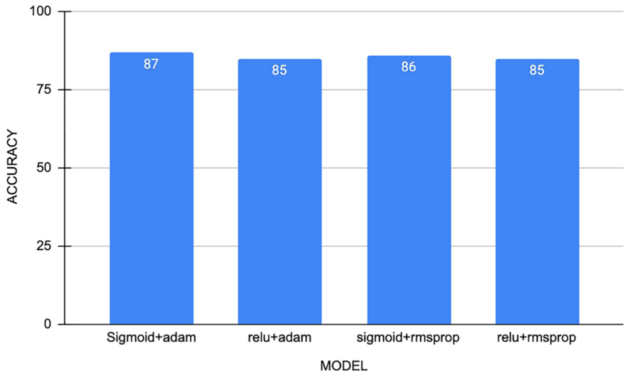


Fig. 4 Activation and optimizer Comparison on baseline models

Table 2 Comparison of activations and optimizer on baseline models

Sl. no	Model Name	Accuracy Before Hypertuning	Accuracy After Hyper-tuning
1	CNN+BIGRU	0.8905	0.9369
2	CNN+BILSTM	0.9135	0.9512
3	BILSTM+BIGRU	0.85330	0.8853

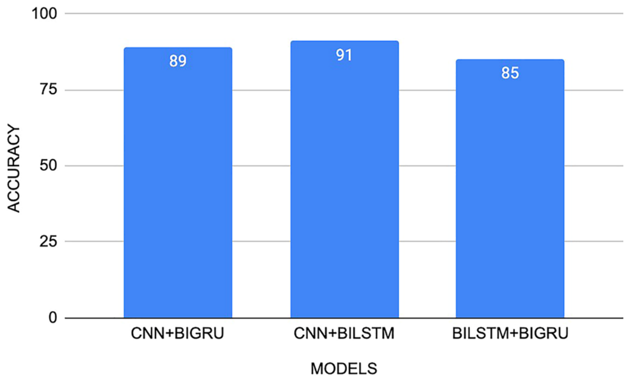


Fig. 5 Hybrid models before hyper-parameter tuning

$$\text{Accuracy} = \frac{\textit{Correct Prediction}}{\textit{Total Prediction}}.$$

Adam and RMSProp as optimizers are used and for activation layers ReLU and Sigmoid are used thus making it a total of 4 combinations.

The activation function chosen has a significant effect on the neural network's capabilities and performance, and various activation functions may be utilized in various portions of the model. The sigmoid function converts any input into a number between 0 and 1. The function sigmoid gives the result near to zero for small values, and a value close to one for high values. Sigmoid is the same as a two-element Softmax with the second element set to zero. As a result, the sigmoid is commonly used in binary classification.

The Adam optimizer is computationally more efficient, requires slight memory, is invariant to diagonal resizing of gradients, and it is well suited for problems with a lot of data/parameters, whereas the RMSProp optimization

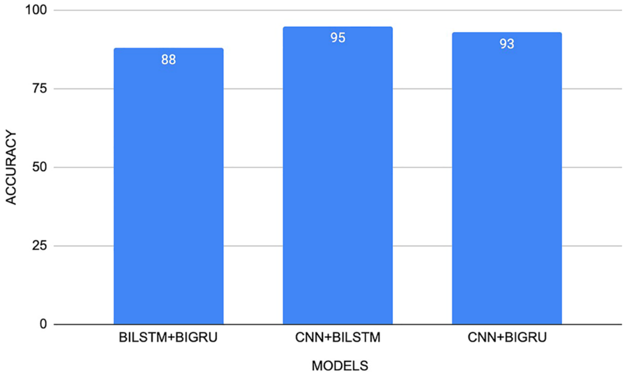
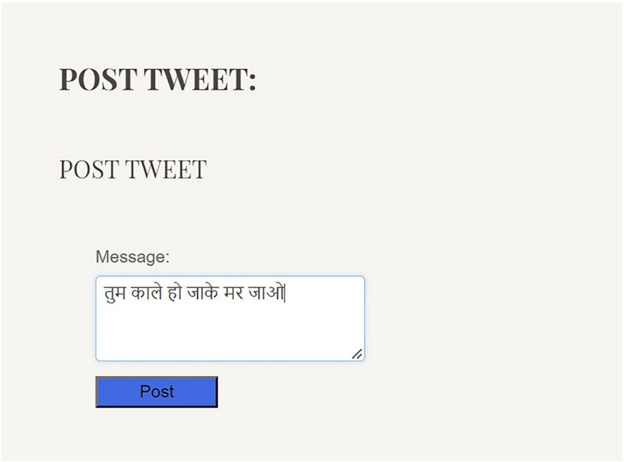


Fig. 6 Hybrid models after hyper-parameter tuning



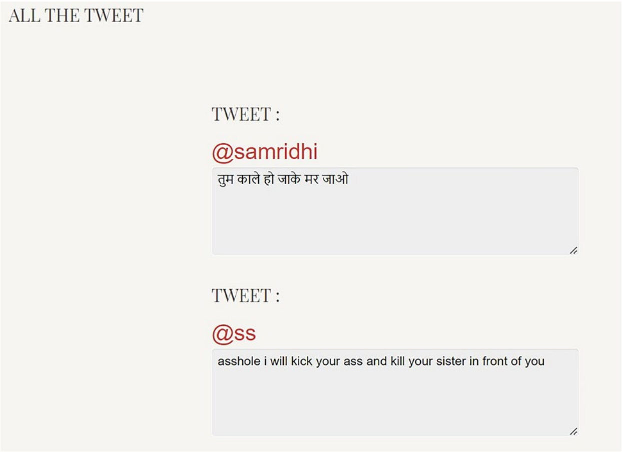
POST TWEET:

POST TWEET

Message:

तुम काले हो जाके मर जाओ|

Fig. 7 Posting tweets



ALL THE TWEET

TWEET :

@samridhi

तुम काले हो जाके मर जाओ

TWEET :

@ss

asshole i will kick your ass and kill your sister in front of you

Fig. 8 Updated feed

algorithm keeps the sections under control the entire time because of the decay rate, which makes RMSProp faster than Adam. Adam obtains his speed from momentum, while RMSProp gives him the capability to adjust gradients in various directions. It's powerful because of the mix of the two. Whereas RMSProp just uses the second moment and speeds it up with a decay rate, Adam employs both first and second moments and is usually the best option (Figs. 4, 5).

The combination of Sigmoid activation with Adam optimizer provided best results among the four configurations as seen in Table 1.

A neural network's design is incomplete without activation functions. The hidden layer's activation function determines how well the model understands the training data. The kind of predictions the model can produce is determined by the activation function used in the output layer.

Other than Sigmoid, ReLU is also employed as the activation layer for our CNN-BiLSTM model's hidden layer. The main reason for this is that the Sigmoid function and its derivative are not complex and therefore help to reduce the amount of time needed to design models; nevertheless, there is a substantial disadvantage of information loss because of the derivative's small range. As a result, the more layers (or perhaps the deeper the Learning Algorithm is), the more information is condensed and lost at every layer, and this magnifies at each level, resulting in significant data loss throughout.

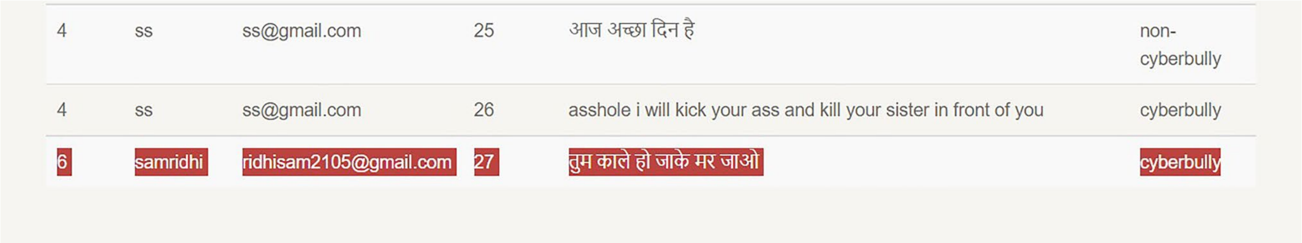
ReLU is non-linear and, unlike the sigmoid function, has no back - propagation algorithm problems. Additionally, for a bigger artificial neural network, the speed of creating models based on ReLU is much faster than using Sigmoids. This is why ReLU is utilized instead of sigmoid for the hidden layer activation.

After the experimentation with various models like CNN and RNN models like LSTM and GRU the result obtained is shown in Table 2. We also tweaked with hyperparameters and did a comparative analysis on what worked well for the dataset. After the comparison, it is evident that the CNN-BiLSTM model has the best performance out of all the various models tested as shown in Fig 6. After putting all the layers together, the model is fitted over our data for 10 epochs and achieved an accuracy of about 98%.

Figure 7 shows that the user can post a tweet and it will automatically get updated in the feed as shown in Fig. 8. Admin can see tweet status as shown in Fig. 9 and also has the privilege to block users who post cyberbullying tweets as it can be seen in Fig. 10.

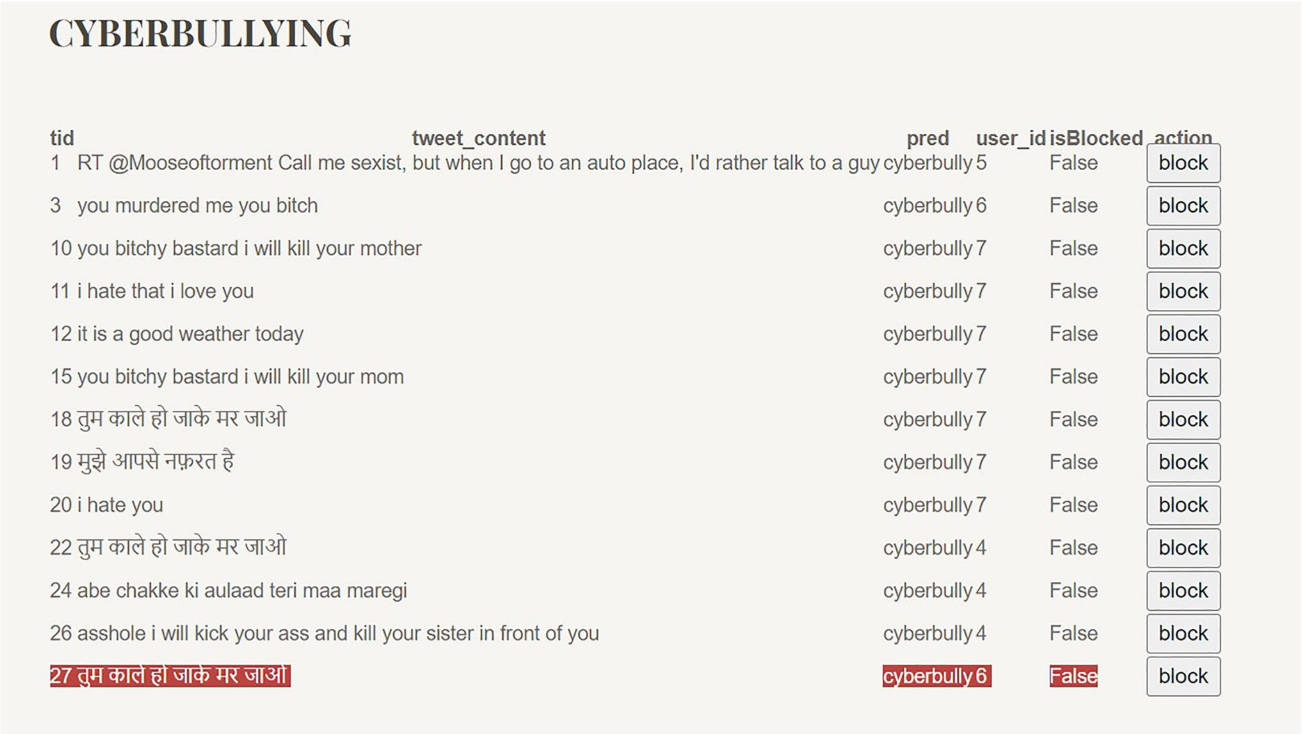
Conclusion

The model for automatically detecting cyberbullying text on multilingual data is addressed and proposed in this work. Solving this issue is critical for controlling social media material in multiple languages and protecting users from the negative impacts of toxic comments like verbal assaults and offensive language. The performance of our various models of neural networks is examined. The CNN-BiLSTM network has the best accuracy. While the CNN alone can only train local characteristics from word n-grams, with its LSTM layer, the CNN-BiLSTM can also learn global features and long-term dependencies. Future research will look at both picture and video elements to see if cyberbullying can be detected automatically.



4	ss	ss@gmail.com	25	आज अच्छा दिन है	non-cyberbully
4	ss	ss@gmail.com	26	asshole i will kick your ass and kill your sister in front of you	cyberbully
6	samridhi	ridhisam2105@gmail.com	27	तुम काले हो जाके मर जाओ	cyberbully

Fig. 9 Tweet status



CYBERBULLYING

tid	tweet_content	pred	user_id	isBlocked	action
1	RT @Mooseoftorment Call me sexist, but when I go to an auto place, I'd rather talk to a guy	cyberbully	5	False	block
3	you murdered me you bitch	cyberbully	6	False	block
10	you bitchy bastard i will kill your mother	cyberbully	7	False	block
11	i hate that i love you	cyberbully	7	False	block
12	it is a good weather today	cyberbully	7	False	block
15	you bitchy bastard i will kill your mom	cyberbully	7	False	block
18	तुम काले हो जाके मर जाओ	cyberbully	7	False	block
19	मुझे आपसे नफ़रत है	cyberbully	7	False	block
20	i hate you	cyberbully	7	False	block
22	तुम काले हो जाके मर जाओ	cyberbully	4	False	block
24	abe chakke ki aulaad teri maa maregi	cyberbully	4	False	block
26	asshole i will kick your ass and kill your sister in front of you	cyberbully	4	False	block
27	तुम काले हो जाके मर जाओ	cyberbully	6	False	block

Fig. 10 Admin feature to block users

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Al-Ajlan, M.A., Ykhlef, M.: Optimized twitter cyberbullying detection based on deep learning. In: 2018 21st Saudi Computer Society National Computer Conference (NCC), pp. 1–5 (2018). IEEE
2. Mahlangu, T., Tu, C.: Deep learning cyberbullying detection using stacked embbedings approach. In: 2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI), pp. 45–49 (2019). IEEE
3. Alam, K.S., Bhowmik, S., Prosun, P.R.K.: Cyberbullying detection: an ensemble based machine learning approach. In: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 710–715 (2021). IEEE
4. Dewani A, Memon MA, Bhatti S. Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for roman urdu data. J Big Data. 2021;8(1):1–20.
5. Luo, Y., Zhang, X., Hua, J., Shen, W.: Multi-featured cyberbullying detection based on deep learning. In: 2021 16th International Conference on Computer Science & Education (ICCSE), pp. 746–751 (2021). IEEE
6. Yadav, J., Kumar, D., Chauhan, D.: Cyberbullying detection using pre-trained bert model. In: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 1096–1100 (2020). IEEE
7. Ahmed, M.T., Rahman, M., Nur, S., Islam, A., Das, D.: Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. In: 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), pp. 1–10 (2021). IEEE
8. Mahat, M.: Detecting cyberbullying across multiple social media platforms using deep learning. In: 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 299–301 (2021). IEEE
9. Jain, N., Hegde, A., Jain, A., Joshi, A., Madake, J.: Pseudo-conventional approach for cyberbullying and hate-speech detection. In: 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), pp. 1–8 (2021). IEEE
10. Iwendi, C., Srivastava, G., Khan, S., Maddikunta, P.K.R.: Cyberbullying detection solutions based on deep learning architectures. Multimedia Systems, 1–14 (2020)
11. Aind, A.T., Ramnaney, A., Sethia, D.: Q-bully: a reinforcement learning based cyberbullying detection framework. In: 2020 International Conference for Emerging Technology (INCET), pp. 1–6 (2020). IEEE
12. Ketsbaia, L., Issac, B., Chen, X.: Detection of hate tweets using machine learning and deep learning. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 751–758 (2020). IEEE
13. Pradhan, A., Yatam, V.M., Bera, P.: Self-attention for cyberbullying detection. In: 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), pp. 1–6 (2020). IEEE
14. Sahana, B., Sandhya, G., Tanuja, R., Ellur, S., Ajina, A.: Towards a safer conversation space: Detection of toxic content in social

media (student consortium). In: 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), pp. 297–301 (2020). IEEE

15. Berrimi, M., Moussaoui, A., Oussalah, M., Saidi, M.: Attention-based networks for analyzing inappropriate speech in arabic text. In: 2020 4th International Symposium on Informatics and Its Applications (ISIA), pp. 1–6 (2020). IEEE
16. Dang, C.N., Moreno-García, M.N., De la Prieta, F.: Hybrid deep learning models for sentiment analysis. Complexity **2021** (2021)
17. Yuvaraj N, Chang V, Gobinathan B, Pinagapani A, Kannan S, Dhiman G, Rajan AR. Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. Comput Electr Eng. 2021;92: 107186.
18. Roy PK, Tripathy AK, Das TK, Gao X-Z. A framework for hate speech detection using deep convolutional neural network. IEEE Access. 2020;8:204951–62.
19. Al-Makhadmeh Z, Tolba A. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing. 2020;102(2):501–22.
20. Yadav, Y., Bajaj, P., Gupta, R.K., Sinha, R.: A comparative study of deep learning methods for hate speech and offensive language detection in textual data. In: 2021 IEEE 18th India Council International Conference (INDICON), pp. 1–6 (2021). IEEE
21. Lee E, Rustam F, Washington PB, El Barakaz F, Aljedaani W, Ashraf I. Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble gcr-nn model. IEEE Access. 2022;10:9717–28.
22. d'Sa, A.G., Illina, I., Fohr, D.: Bert and fasttext embeddings for automatic detection of toxic speech. In: 2020 International Multi-Conference on:“Organization of Knowledge and Advanced Technologies”(OCTA), pp. 1–5 (2020). IEEE
23. Jiang, L., Suzuki, Y.: Detecting hate speech from tweets for sentiment analysis. In: 2019 6th International Conference on Systems and Informatics (ICSAI), pp. 671–676 (2019). IEEE
24. Mohaouchane, H., Mourhir, A., Nikolov, N.S.: Detecting offensive language on arabic social media using deep learning. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 466–471 (2019). IEEE
25. Dubey, K., Nair, R., Khan, M.U., Shaikh, S.: Toxic comment detection using lstm. In: 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), pp. 1–8 (2020). IEEE

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.