Cyberbullying detection in Twitter space using LSTM Neural Network

Thesis

Submitted In Partial Fulfilment of the Requirement for the Degree of

MASTER OF TECHNOLOGY

IN

COMPUTER TECHNOLOGY

BY

ARPITA DAFADAR

University Roll Number: 002010504008

Examination Roll Number: M6TCT23025

Registration Number: 154174 of 2020-21

Under The Guidance Of

PROF. DIGANTA SAHA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY
JADAVPUR UNIVERSITY, KOLKATA

132, Raja Subodh Chandra Mallick Road,

Jadavpur, Kolkata, West Bengal, 700032

JUNE, 2023

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

This is to certify that the dissertation titled **Cyberbullying detection in Twitter space using LSTM neural network** was completed by Arpita Dafadar, University RollNo: 002010504008, Examination Roll Number: M6TCT23025, University Registration No: 154174 of 2020-21, under the guidance and supervision of Prof. Diganta Saha, Department of Computer Science and Technology, Jadavpur University. The findings of the research detailed in the thesis have not been incorporated into any other work submitted to earn a degree at any other academic institution.

_____

Prof. Diganta Saha

Department of Computer Science & Engineering

Jadavpur University

COUNTERSIGNED BY

_____

Prof. Nandini Mukherjee

Head of the Department

Department of Computer Science And Engineering

Jadavpur University

COUNTERSIGNED BY

_____

Prof. Arthendu Ghosal

Dean, FET

Faculty of engineering and Technology

Jadavpur University

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled **Cyberbullying detection in Twitter space using LSTM neural network** is a bonafide record of work carried out by ARPITA DAFADAR in partial fulfilment of the requirements for the award of the degree Master of Technology in the Department of Computer Science and Engineering, Jadavpur University during the period of June 2022 to June 2023 (5^{th} & 6^{th} Semester). It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for the purpose for which it has been submitted.

_____

Signature of Examiner
Date:

_____

Signature of Supervisor
Date:

DECLARATION

I certify that,

(a) The work contained **Cyberbullying detection in Twitter space using LSTM neural network** in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them inthe text of the thesis and giving their details in the references. Further, I havetaken permission from the copyright owners of the sources, whenever necessary.

Arpita Dafadar

Master of Technology

Roll No: 002010504008

Exam Roll No: M6TCT23025

Registration No: 154174 of 2020-21

Department of Computer Science & Engineering

Jadavpur University, Kolkata

ACKNOWLEDGEMENT

First and foremost, I want to express my gratitude to God Almighty for provoiding me with the strength, wisdom, and capability to go on this amazing adventure and to continue and successfully finish the embodied research work. I'd like to thank Professor Diganta Saha of the Department of Computer Science and Engineering at Jadavpur University for his excellent assistance, consistentsupport, and inspiration during my dissertation. I owe Jadavpur University a great debt of gratitude for providing me with the chance and facilities to complete our thesis.

I am grateful to every one of the teaching and non-teaching personnel whose assistance has made our trip during my research time much easier. I would liketo thank my project partners Imanur Rahman and Rohan Biswas, my seniors, and my friends fo providing me with regular encouragement and mental support throughoutour effort.

Last, but not the least, my family deserves great recognition. There are no words to express my gratitude to my mother and father for all of the sacrificesyou've made on my behalf. Your prayers for me have kept me going thus far.

Arpita Dafadar

Master of Technology

Roll No: 002010504008

Exam Roll No: M6TCT23025

Registration No: 154174 of 2020-21

Department of Computer Science & Engineering

Jadavpur University, Kolkata

Table of Content

Declaration ……5

Acknowledgement……..6

1. Abstract…….……...…....8

2. Introduction……...……...……...10

2.1 Overview ……..…….10

2.2 Application……11

2.3 Challenges……..12

3. Literature Survey……..…….13-14

4. Proposed work…… 15

4.1 Algorithm…….15-18

4.2 Mathematical Notation……...…19

4.3 Schematic description of model…….....…….....……....……20

4.4 Flowchart description of model……...……...21

4.5 LSTM……...…….22-23

4.6 Fasttext embedding method…….…….24-25

5.1 Datasets …….……...…..26

5.2 Data pre-processing steps……..……27-32

5.3 Result and performance analysis……..……33-35

6 Conclusion and future work……..……...36-37

7. References……...……38-40

Chapter 1

Abstract

Cyberbullying is a troubling and disconcerting online misconduct. It seems in different forms and mostly in textual formats on various social networking sites.

Cyberbullying has become a common problem on online platforms, particularly social media platforms like Twitter. Detecting and contending cyberbullying is crucial to keeping users safe and healthy. Cyberbullying poses a serious risk to people's mental health and entails effective detection and prevention. Various algorithms like Gated Recurrent Units (GRU), Recurrent Neural Networks (RNN), and Bidirectional Long Short Term Memory (BLSTM) are used to define the experimental results.

This article focuses on developing a cyberbullying solution using Long Short Term Memory (LSTM), a deep learning technique for analyzing Twitter data and sequential data.

This research emphasizesexploiting LSTMs' ability to capture data content and excerpt relevant structures from the nature of Twitter data. The plan is comprising the collection of data, prioritization, LSTM model design, training, and evaluation. Experimental results prove that the theLSTMbased solution outperforms the basic method and demonstrates its accuracy and robustness in defining the nature of Twitter cyberbullying. The outcomes of this research help to meet the urgent need for cyberbullying detection technology and pave the way for future expansions in this field. Data pre-processing steps such as cleaning of text, tokenization, Stemming, Lemmatization, and removing stop words are used. After the pre-processing has been done, textual data that are already cleaned, have been passed to these algorithms of deep learning for forecasting purposes.

The process begins by collecting a large database of text tweets, including both cyberbullying and nonbullying ones**.** Using advanced techniques such as tokenization and rooting to convert the raw text into a format suitable for LSTM models.

The proposed model was evaluated on a diverse and representative Twitter dataset, taking into account various criteria such as precision, recall, and F1 score. The results demonstrate the effectiveness of the LSTMbased method in identifying cyberbullying incidents, outperforming traditional machine learning, and demonstrating the ability to detect cyberbullying on Twitter.

LIST OF ABBREVIATIONS

NLP: NATURAL LANGUAGE PROCESSING

LSTM: LONG SHORT-TERM MEMORY

BLSTM: BIDIRECTIONAL LONG SHORT-TERM MEMORY

GRU: GATED RECURRENT UNIT

TP: TRUE POSITIVE

TN: TRUE NEGATIVE

FP: FALSE POSITIVE

FN: FALSE NEGATIVE

RNN: RECURRENT NEURAL NETWORK

Chapter 2

Introduction

2.1 Overview

Social media platforms have revolutionized communication and networking by giving people authoritative tools to express themselves and involve in online conversations.

However,asidefrom the benefits**,** these platforms have also seen an alarming increase in cyber bullying. Cyberbullying **is** the use of digital communication to harass, threaten or annoy someone.

Twitter, one of the most prevalent social media platforms, is no exception. Detecting and addressing cyberbullying in real time is critical to protecting users from emotional, psychological, and possibly long-term consequences. Traditional cyberbullying methods often rely on benchmarks or guidelines that limit accuracy and efficiency. Therefore, there is a need for a technical system that can analyze big data and accurately detect cases of cyberbullying. This article aims to solve the problem of cyberbullying discovered on Twitter by proposing new techniques based on Long Short Term Memory networks. LSTMs are a type of recurrent neural network (RNN) that is excellent at modeling data connections and storing long-term prospects. Leveraging the nature of tweets, LSTMs can analyze the finer points and identify subtle patterns that point to cyberbull ying. In this work, I have tried out my experiment on English language datasets. This dataset contains 6595 tweets classified into two different classes Insulting and Non-Insulting. Various combinations are used to increase the efficiency and robustness of LSTM-based models. The word embedding is used to represent the meaning of language and allows the model to understand the deeper meaning. The monitoring function is used to focus on the impact of tweets by highlighting the main topics related to cyberbullying. Also, regular work processes are used to reduce intensity and improve generalization.

2.2 Application

Cyberbullying on Twitter is a concerning issue that can have different consequences for individuals targeted by online harassment. LSTM (Long Short-Term Memory) is a type of recurrent neural network that can be applied to analyze text data, including tweets, and detect patterns related to cyberbullying. Here's how LSTM can be used to address cyberbullying on Twitter:

Dataset Assortment: Collect a dataset of tweets containing examples of cyberbullying and non-cyberbullying tweets. The dataset should be labeled or annotated to indicate whether each tweet represents cyberbullying or not.

Data Pre-processing: Pre-process the collected dataset by removing unnecessary characters, converting text to lowercase, and tokenizing the text into individual words or subwords. You may also want to remove stop words and apply stemming or lemmatization techniques to normalize the text.

Word Embedding: Transform the pre-processed text into numerical representations using word embedding techniques like Word2Vec or GloVe. These techniques map each word to a high-dimensional vector, capturing semantic relationships between words.

LSTM Model Training: Train an LSTM model using pre-processed and embedded data. The LSTM model learns to analyze the sequential nature of the text, capturing dependencies and patterns across the tweet. The model can be trained using libraries like TensorFlow or PyTorch, specifying the appropriate architecture, hyperparameters, and loss function.

Model Evaluation: Evaluate the trained LSTM model on a separate test dataset to measure its performance. Common evaluation metrics for text classification tasks include accuracy, precision, recall, and F1 score. This step helps determine how well the model can differentiate between cyberbullying and non-cyberbullying tweets.

Deployment and Integration: Integrate the trained LSTM model into an application or platform that can analyze incoming tweets in real time. The model can process new tweets and classify them as cyberbullying or non-cyberbullying, flagging potentially harmful content for further action.

Post-processing and Intervention: Once cyberbullying tweets are identified, appropriate actions can be taken, such as notifying moderators, blocking or filtering content, or providing support to the affected users. The model's predictions can help facilitate intervention and prevent further harm.

It's important to note that the effectiveness of an LSTM-based approach for cyberbullying detection depends on the quality and diversity of the training dataset, as well as the model's ability to generalize to different types of cyberbullying instances. Ongoing monitoring and periodic updates to the model may be necessary to adapt to evolving forms of cyberbullying on Twitter.

2.3 Challenges

There are several challenges associated with using LSTM for detecting cyberbullying on Twitter. Here are some of the key challenges you might encounter:

Data Collection and Labeling: Collecting a well-balanced and diverse dataset that accurately represents cyberbullying instances on Twitter can be challenging. Labeling tweets as cyberbullying or non-cyberbullying can be subjective, leading to potential inconsistencies. Ensuring the dataset's quality and representativeness is crucial for training an effective LSTM model.

Contextual Understanding: Cyberbullying often involves subtle and context-dependent language usage, such as sarcasm or irony. LSTM models may struggle to capture the full context and interpret the intended meaning behind certain tweets accurately. This limitation can result in false positives or false negatives, where harmless tweets are misclassified as cyberbullying or vice versa.

Variability and Evolving Nature of Cyberbullying: Cyberbullying tactics can evolve rapidly as new platforms and communication trends emerge. LSTM models trained on existing datasets may struggle to generalize to new forms of cyberbullying that were not present in the training data. Continuous monitoring and updates to the model are necessary to address these challenges.

Imbalanced Data: The distribution of cyberbullying and non-cyberbullying instances in the dataset may be imbalanced, with cyberbullying instances being relatively rare compared to non-cyberbullying instances. This imbalance can affect the model's ability to learn effectively, leading to biased results and lower accuracy in detecting cyberbullying.

Noise and Irrelevant Information: Twitter data often contains noise, including unrelated content, spam, or unrelated hashtags. The presence of such noise can make it more difficult for the LSTM model to identify and focus on relevant features associated with cyberbullying.

Adversarial Attacks: Cyberbullies may intentionally manipulate their language or use tactics to evade detection by machine learning models. They may modify or obfuscate their messages to avoid being flagged, making it challenging for LSTM models to accurately identify such instances.

Ethical Considerations: Deploying a cyberbullying detection system on Twitter raises ethical concerns. It's crucial to balance the need for user protection with privacy considerations and ensure that the system does not inadvertently infringe upon users' rights or stifle freedom of expression.

To address these challenges, it is recommended to employ a combination of techniques, such as ensemble models, incorporating external knowledge sources, and regularly updating the training data to adapt to the evolving landscape of cyberbullying on Twitter. Additionally, human moderation and intervention should complement automated detection systems to ensure accurate and fair outcomes.

Chapter 3

Literature Survey

In the last few years, many people have worked and made a significant amount of progress in the field of cyberbullying detection. They have projected many methods to automatically detect cyberbullying with higher accuracy. These papers provide insights into the application of LSTM models for cyberbullying detection on Twitter and highlight different techniques, datasets, and evaluation methodologies used in the field. Exploring these works will give you a good starting point for understanding the advancements and challenges in cyberbullying detection on Twitter using LSTM.

In the paper, Hani et al. 2019 [1] have proposed a supervised machine-learning approach for detecting and averting cyberbullying. Through their work, they achieved 92.8% accuracy with the help of Neural Network (NN) and 90.3% accuracy with SVM on the dataset collected from Kaggle [2] which contains in general 12773 conversation messages collected from Formspring. They have also shown that NN outperforms other classifiers of related work on a similar dataset.

Another approach Talpur and O'Sullivan, 2020 [3] have proposed is a cyberbullying detection framework to produce features from Twitter content by leveraging a pointwise mutual information method. In this study, they applied Sentiment, Embedding, and Lexicon features along with PMI-semantic orientation. Random Forest, Naïve Bayes, KNN, and Support Vector Machine Learning algorithms were applied with the extracted features from the Twitter dataset collected by [4]. The dataset is publicly accessible on the git repository. The best overall classifier performance in the binary setting was accomplished by Random Forest for having an AUC of 0.971 and an f-measure of 0.929.

In the approach planned by Iwendi et al., 2020 [5] they have performed an empirical analysis to regulate the efficiency and performance of deep learning algorithms in detecting insults in Social Commentary on the dataset collected from Kaggle. They have used deep learning models like Gated Recurrent Units (GRU), Bidirectional Long Short-Term Memory (BLSTM), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) for experimental results. BLSTM on this dataset outpaces others in terms of Precision, Recall, F1-Measure, and AUC. Precision, Recall, and F1 Measure scores for the normal class using

the BLSTM model are 86%, 91%, and 88% whereas, for the Insult class, Precision is 71%, Recall is 60% and F1-Measure is 65%.

Furthermore, Al-Ajlan, and Ykhlef, 2018 [6] proposed a method, after fetching dataset from Twitter using Twitter streaming API they used a novel algorithm CNN-CB that eradicate the necessity for feature engineering. CNN-CB is a content-based convolutional neural network (CNN) and includes semantics using word embedding. They have shown experimentally that the CNN-CB algorithm outperforms traditional content-based cyberbullying detection with an accuracy of 95%, precision of 93%, and recall of 73%.

Moving on to another author, Murnion et al., 2018 [7] used War of Tanks game chat to get their dataset and manually categorized them, and then compared them to simple

Naïve classification that uses sentiment analysis as a feature. But their results were not up to the mark when compared to the manually classified results.

In another work by Ahmed et el. 2021 [8] we found they used Neural Networks (NN) for, multiclass classification, and binary classification on the dataset [9] containing comments from the Facebook platform. The classifier model for binary classification holds a precision of 90%, recall of 75%, and F1-score of 82%. Whereas for multiclass classification the classifier model holds a precision of 81%, recall of 74%, and F1-score of 76%.

Then another approach was proposed by Muneer and Fati, 2020 [10]. They got their dataset from Twitter and divided the dataset into a 7:3 ratio for training and testing. They have used seven machine learning classifiers, namely, Light Gradient Boosting Machine (LGBM), Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), Stochastic Gradient Descent (SGD), AdaBoost (ADB), and Support Vector Machine (SVM). And through experiments, they have shown Logistic Regression (LR) is the best among these classifiers in terms of median accuracy (around 90.57%). Among the classifiers, logistic regression achieved the best F1 score (0.928), SGD achieved the best precision (0.968), and SVM achieved the best recall (1.00).

Chapter Summary:

Building upon these studies, the current project employs a combination of analysis on tweets using LSTM models and a memory based approach to detect whether it is bullying and non-bullying on tweets. This chapter presented a thorough examination of literature and related works of cyberbullying using different tweets.

Chapter 4
Proposed work

This chapter of this report addresses the problem statement and outlines the methodology employed for cyberbullying detection using tweets of prominent figures. Cyberbullying on social media platforms, such as Twitter, poses a significant threat to users' well-being and safety. This research aims to develop an effective cyberbullying detection system specifically designed for tweets. We propose a novel approach that combines LSTM models with contextual embedding to capture both the sequential nature of the text and the contextual meaning of words. The proposed system aims to improve the accuracy and robustness of cyberbullying detection on Twitter.

4.1 Algorithm

In this experiment, we have tried out the RNN-based LSTM Neural Network for our datasets. We have tried out the experiment in the English language. Our dataset consists of hate comments (bullying) which are mainly collected from Twitter.

These algorithms are elaborated in six different steps. They are

1. Data Pre-processing

2. Feature Removal

3. Test Train Validation split

4. Describe the model

5. Fit the model

6. Checking the accuracy of the model

1. Data Pre-processing

Input -> .csv file containing raw tweets along with their respective labels.

Output -> .csv file containing cleaned tweets along with their respective labels.

Step 1: Read the tweets of the dataset.

Step-2: Erase the username starting with @.

Step-3: Remove the URL from every single tweet.

Step-4: Swap multiple white spaces with a single white space.

Step-5: The pre-processed dataset is ready.

2. Feature Extraction

Input –> Cleaned Dataset

Output -> Feature extracted and moved for test train split

Step 1: Prepare the maximum sequence length to 250.

Step 2: Describe the embedding dimension to 100.

Step 3: From Keras. preprocessing we import Tokenizer and apply it to the column of 'Cleaned Tweets'.

Step 4: Now we import texts_to_sequences from keras. pre-processing and apply it to the column of 'Cleaned Tweets' and save it in a variable.

Step 5: At this time we import the pad sequence from Keras. preprocessing and applying it over the variable that we get from step 4 and passing the parameter value of padding length that we have initialized in step 1.

Step 6: We can encode the column that is associated with the labels of tweets.

Step 7: Now, Feature extraction is completed.

3. Test Train Validation split

Input –> Feature extracted dataset from part 2

Output–> Feature extracted test. train and validation of data

Step 1: We have done the test, the train split in a 60:40 ratio over both ‘Cleaned Tweets’ and 'Label’ and saved them in distinct variables.

Step 2: Test. The train and Validation split is now concluded.

4. Defining the model

Input –>None
Output–> A sequential model with an LSTM layer

Step 1: We import Sequential () from Keras and save it as a variable named model.

Step 2: We add an Embedding layer at first with parameters MAX_NB_WORDS, and EMBEDDING_DIM that we have defined in part-2.

Step 3: We have then added a special drop-out layer with a minimum rate = 0.2 to reduce overfitting.

Step 4: At the moment we add an LSTM layer with 100 memory units with a dropout rate of 0.2 for the input layer, and recurrent dropout rate = 0.2. This is to learn the temporal dependencies among words in the sequences.

Step 5: At present, we add a dense () layer with ‘n’ output units and the activation function as ‘softmax’ or ‘sigmoid’ (according to binary or multiclass classification) which will give the probability distribution over ‘n’ classes.

Step 6: Now our model is equipped to compile.

Step 7: At this instant, we compile the model with categorical cross-entropy loss or binary cross-entropy, Adam optimizer, and accuracy metric for evaluation during training.

5. Fitting the model

Input –>Pre-processed dataset from part 3 and compiled model from part 4.

Output–>Training of LSTM model using the pre-processed dataset and a trained model.

Step 1: We have fixed our model with the pre-processed dataset and predefined

Epoch number and batch size as our choice.

Step 2: Make the validation split 50 % over the test data.

Step 3: Our Model is now trained.

6. Check the accuracy of the model

Input –>Trained model from part 5 and test dataset from part 3.

Output ->Overall Accuracy score of the model, F1 score of each class, and confusion matrix from each class.

Step 1: Using the trained model we predicted the labels of each tweet from a test data set and store them in a variable.

Step 2: From sklearn.metrics we have imported the predefined objects and using the output from step 1 and test data from part 3 we have got our overall accuracy, F1 score, and confusion matrix.

Step 3: Record accuracy, F1 score, and confusion matrix in a Word document.

Step 4: Completed.

4.2 Detection of Cyberbullying in Twitter Space Using LSTMNeural Network

MATHEMATICAL NOTATION:

H = f (T, L), Where

- T is input in string format. This is a sentence in which we will try to predict whether it is a Hateful comment or a Non Hateful one.
- L is our LSTM neural network model which we will train to predict the class of Input T.
- f is a function that takes a string input T and LSTM neural network model L and shows the class of input T.

We will train our LSTM neural network model L using both hate and non-hate tweets and then we will try to predict the class of tweet T.

The model L can be implemented as a neural network model that takes the word embedding of the input T as input and outputs the most probable class of that input.

The function f can be trained using supervised learning, where the training data contains a set of messages that are already labeled with either bullying or non-bullying. The function will be trained to predict the correct class of input T using the LSTM model.

We can also attach different types of word embedding models such as *Fasttex*t with model M. Respective embedding model trained on a particular language can also help increase the accuracy of our model M for a particular language's tweet.

4.3 METHODOLOGY

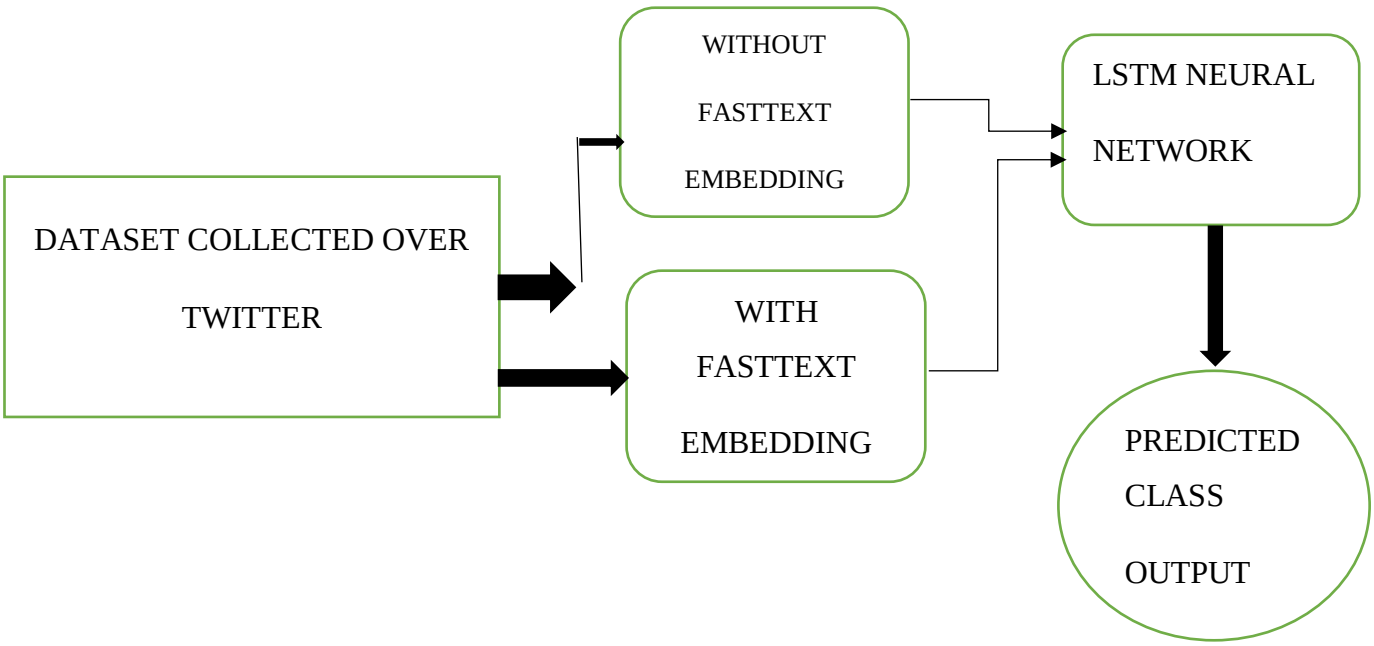
Schematic Description of Model

In this segment, we are discussing the comprehensive approaches of our work and discussing how we congregate to our final result.

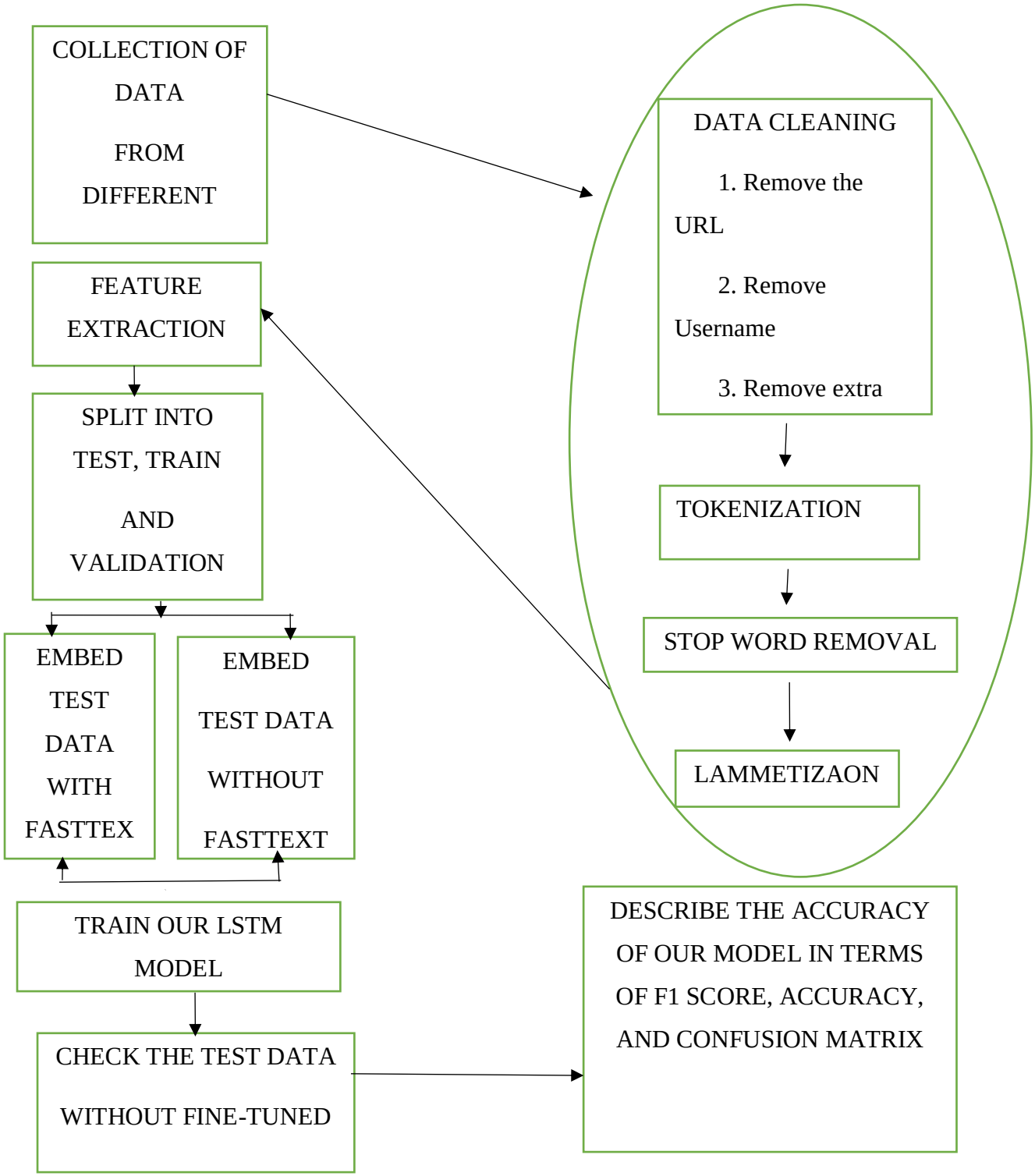
We are also describing how we pre-process the accessible dataset to make a more precise training dataset for our model and help us to get a nearly better result.

This project depends on the results from the experiment results, which can be assessed as a function of the use of suitable inputs and technologies used on the input data. We have deliberated numerous approaches and experimented on datasets collected from social media in the related work section.

Here, we have used LSTM deep neural network with *fast text* embedding and without *fast text* embedding and documented our results.



4.4 Block Diagram of the Model



4.5 LSTM Deep Neural Network

LSTM is a recurrent neural network (RNN) architecture gaining popularity in natural language processing (NLP) and data analysis for sequential tasks. It has been shown to overcome the limitations of traditional RNNs in capturing long-lived populations.
The main innovation of the LSTM is the ability to selectively remember and forget information in the next step. It does this using different cells of memory and a special technique. This technique allows LSTMs to learn and retain important information over time while selecting unimportant or unreliable information.

The LSTM architecture consists of a memory chain where each cell is responsible for capturing and storing information at a given time. Each memory cell has three main compone nts: an input gate, a forget gate, and an output gate.

This table
determines how often new data should be stored in cell memory over a given period**.** It takes into account current ideas and previously hidden situations. On the other hand, the
forget gate determines what information to discard from the memory cell based on the current input and the previous hidden state.

Memory storage and forget selection ensures long-term storage of LSTMs. The output gate controls how much data from the memory cell should be accessible to the next layer or used for prediction. It is based on existing ideas and preceding hidden situations. The latent state often referred to as the output of the LSTM, is calculated from the current input, the previous hidden state, and the contents of the current memory. It carries important information learned from the system that can be used for subsequent tasks such as classification.

The ability of LSTM architectures to capture long-
term dependencies makes them particularly useful for tasks involving continuous data such as text analysis, recognition of speech, and
estimated time. In the context of Twitter cyberbullying detection, LSTM networks can effectively analyze the tweet order and capture the content of the data, making sure of the cyb erbullying events.

Using the power of LSTM networks, researchers have developed effective models for making connections and understanding data in data, ultimately contributing to the development of va rious NLP techniques and providing insight into global challenges such as Cyberbullying.

Input gate:

The input gate in the LSTM network determines the amount of new data to be stored in the memory cell at a time. In the context of cyberbullying detection, the input gate takes into account the current login (for example, a tweet) and the previous confidential state to det ermine what information is relevant and should be stored in the memory unit**.**

For example, an input gate may learn to target specific words, expressions, or outlines in tweets that may specify cyberbullying.
By choosing to collect this crucial information, LSTMs can better capture and understand the key features needed to detect cyberbullying.

Forget Gate:

Forget gate control is the information that must be erased from the cell of the memory or can be discarded. It takes into account the current input and the previous hidden state to regulate which data is irrelevant and should be removed from the memory cell.

In the context of cyberbullying detection, forget gates help LSTM networks remove irrelevant or outdated information from previous steps. This allows the model to change the tweet content from time to time and avert inappropriate information from affecting the cyberbullying procedure. By selectively fo rgetting less vital information, LSTMs can focus on the most important features that are affecting the detection of cyberbullying.

Output gate:

The output gate in LSTM networks controls the flow of data from the memory cell to the next layer or as the final output. It helps the current strategy and previous confidentiality to classify information that needs to be released for further processing or forecasting.

In the context of cyberbullying detection, the output gate determines how significant information in cell'smemory should be used to categorize tweets as cyberbullying or not cyberbullying. It allows LSTMs to extract learned representations from the memory of the cell and create outputs that summarize relevant information to be true.

All these gates allow LSTMs to capture selectively store, forget, and apply in tweets for necessary investigation of cyberbullying patterns.

4.6 FastText Embedding

FastText embedding is a standard method for creating word embedding that can be used in cyberbullying recognition solutions on tweets based on different LSTM models. FastText is an addition of Word2Vec, an extensively used word embedding technique that presents subword material for capturing contextual and morphological texts more effectively.

Now, FastText embedding has been applied in the perspective of the detection of cyberbullying on tweets spending LSTM models:

Pre-processing: Pre-processing like eliminating special characters, hashtags, URLs, and mentions are characteristically accomplished for cleaning the text data beforehand engendering FastText embedding.

Tokenization: The tweet text has been tokenized into distinct words or subwords. FastText activates at the subword level which allows it for handling abbreviations, out-of-vocabulary words, and slang terms efficiently.

Training of FastText embedding: FastText embedding is produced by working out a FastText model on a huge amount of text data that could contain a group of tweets or a more wide-ranging dataset. The model acquires word demonstrations by allowing for both the separate words and their subword apparatuses.

Embedding generation: Every token in the tweet is planned to its consistent FastText embedding vector. In the area of subwords, the embedding of discrete subwords is characteristically averaged or united to epitomize the overall token.

Padding and truncation: Similar to other pre-processing stages, truncation or padding may be applied to guarantee steady input proportions for the LSTM model.

LSTM training and forecasting: While FastText embedding is organized, the LSTM model is trained using labeled instances to acquire outlines suggestive of cyberbullying. Throughout prediction, new tweets undergo the same pre-processing steps, comprising the generation of FastText embedding, and the LSTM model creates predictions based on the learning patterns.

FastText embedding offers numerous advantages in cyberbullying recognition on tweets. They are capturing both the morphological and semantic material of words, permitting the model to appreciate the significance of words also with variations, misspellings, or

abbreviations. This is mainly vital in the framework of tweets, wherever users frequently use creative language and unusual word constructions.

Furthermore, FastText embedding can switch out-of-vocabulary words using leveraging subword info. This is vital for cyberbullying recognition, as new slang expressions otherwise abusive language may appear over time, and the model requirements for handling such circumstances effectually.

It's worth observing to generate FastText embedding characteristically necessitates a huge amount of training data for capturing the language diversity for use efficiently. Yet, pre-formed FastText embedding trained on huge-scale amounts are also accessible, that can be applied directly if they cover the vocabulary and language characteristics of the tweet dataset.

Chapter Summary:

In conclusion, this chapter presented the introduction, dataset, mathematical notation for Detection of Cyberbullying in Twitter Space Using LSTM Neural Network, related work section, the flowchart diagram of the model and their working process, and the input gate, forget gate, and output gate of the LSTM model.

In summary, the LSTM model provides a decent method for analyzing cyberbullying on Twitter. Using the power of deep learning and analytics, models can identify
outlines of cyberbullying in tweets,
help lessen the dangers of online bullying, and upsurge security in the community.

The LSTM model was trained using a pre-processed dataset of tweets where all these tweets can be fed sequentially.
The model learns the hidden patterns and characteristics of cyberbullying and adjusts their weightiness and biases accordingly. The training consists of optimizing the model using techniques such as backpropagation and downscaling to reduce predictability.

CHAPTER 5

This second chapter of this report addresses the Algorithm that denotes the pre-processing of data, extraction of features, test and train validation splitting, defining the model, fitting the model, and lastly checking the accuracy of the model.

5.1 Datasets

We have taken the English dataset to train and test our deep-learning model. The full description of them is recorded below.

At first, we took two datasets for bullying detection. One is Train data and the other one is Test data . All datasets of Train and Test have been appended in an Excel data sheet.

We have taken all datasets from Kaggle and the link has been attached in the reference section. It contains **6595 tweets** after performing append and is classified into two different classes. The distribution of tweets among these dualistic classes is as charted.

Label	Count
Insulting	1743
Non-Insulting	4893

5.2 Data pre-processing steps:

Data pre-processing shows a vital role in arranging the tweet data for the detection of cyberbullying using LSTM models. Here are the main steps involved in data pre-processing:

I. Text cleaning:

Text cleaning is a vital step in pre-processing of tweet data for cyberbullying detection using LSTM models. Here are several common methods for cleaning text:

Eliminating Special Characters and Punctuation: Tweets often comprise special characters, like emojis, symbols, or emoticons that have not donated to the denotation of the text. These characters can be erased or swapped with proper tokens. Moreover, punctuation marks like commas, periods, and exclamation marks can be detached as they are not crucial for perceiving cyberbullying.

Managing Hashtags and Mentions: In tweets, indications (e.g., @username) and hashtags (i.e., #cyberbullying) are mutual. Depending on the precise necessities, one can select to eliminate them or swap them with any generic term. For instance, one could change references with "USER" and also hashtags with "HASHTAG."

Eliminating URLs: Tweets frequently comprise URLs that can be substituted with a general term like "URL" or wholly detached, as they are contributing to the analysis content.

Managing Numeric Digits: Numeric digits may not be important for spotting cyberbullying in any text. One can pick to eradicate or exchange them with a generic term, like "NUM."

Eliminating Stop words: Stop words are usually the most used words that do not carry ample semantic meaning, like "a,", “and," "the," etc. Eliminating stop words can decrease focus and noise on more expressive words. Though, in specific cases, holding certain stop words may be essential that is liable in this context.

Modifying Spelling and Abbreviations: Text data in tweets frequently comprises misspelled words or acronyms. Applying this method including checking these spelling and intensifying abbreviations can help in standardizing the text and expand the model's understanding of this content.

It has to be noted that the extent of cleaning of all the text and the specific methodologies will vary dependent on the characteristics of the tweet dataset and the requirements of the cyberbullying research project. For analysis of
cyberbullying, it is suggested to try diverse text cleaning methods and assess their effects on the performance of LSTM models.

II. Tokenization

Tokenization is a central step in the procedure to analyze text data that include tweets, for detection of cyberbullying using LSTM (Long Short-Term Memory) models. Tokenization denotes the procedure of cutting down any text document into minor units called tokens. In the area of natural language processing (NLP), tokens are characteristically words or subwords.

While dealing with these tweets, tokenization develops mainly significantly due to the inadequate count of characters and the unique characteristics of this platform. Here are the steps of tokenization that can be applied in the framework of cyberbullying detection resolutions based on the LSTM models:

Pre-processing: Before tokenization, it is common to perform pre-processing steps such as removing special characters, hashtags, URLs, and also mentions. These steps are helping to clean the text and eliminate noise that might delay the detection method.

Word-level tokenization: In the tokenization of word-level, the tweet is divided into distinct words or tokens. Every word characterizes a discrete token, and the LSTM model procedures these tokens successively. For instance, the tweet "I dislike you, you are a failure" would be tokenized into ['I', 'dislike', 'you', ',', 'you', "are", 'a, 'failure'].

Tokenization of Sub-word-level: Sub-word-level tokenization is an extra method that splits the text into subwords units that can detect more comprehensive info. This method is mainly beneficial for managing out-of-vocabulary words, slang terms, or abbreviations. Common methods for subwords tokenization comprise Word Piece and Byte Pair Encoding (BPE). For example, the word "unavailable" might be tokenized into ['un', '##ava', '##ail', ##ble'].

Truncation and Padding: Later tokenization, the tweet's tokens are not of equivalent length. For ensuring reliable input dimensions for the LSTM model, truncation or padding is often applicable. Padding includes the addition of special tokens (e.g., <PAD>) to smaller tweets

for matching the length of lengthier ones, whereas truncation curtails stretched tweets to a predefined concentrated length.

Representation of Embedding: When tokenization is finished, every token wants to be transformed into a numerical illustration for the LSTM model to generate. This is frequently done by embedding words, that help in mapping words or subwords to impenetrable vectors in an incessant space. Prevalent word embedding methods comprise GloVe, Word2Vec, and Fasttext.

Training and prediction: With the embedded and tokenized data, the LSTM model can be skilled using labeled samples to acquire patterns indicative of cyberbullying. Throughout prediction, different tweets can be tokenized similarly, and the LSTM model can make predictions if they comprise examples of cyberbullying based on the learned outlines.

III. Stemming

Stemming is a text normalization method that is usually used in NLP tasks, containing cyberbullying detection resolutions based on LSTM models. Stemming targets to decrease words to their root or base form, recognized as a stem, by eliminating prefixes and suffixes. The main resolution of stemming is to lessen the dimensionality of the text records and assemble words with similar base meanings, irrespective of their modulations.

Though, in the circumstances of detection of cyberbullying on tweets, stemming might not be as operative or pertinent as it works for NLP tasks. The reasons are-

Unceremonious language and slang: Tweets frequently comprise slang, informal language, formations of creative words, and abbreviations that are the largest on social media stages. Stemming algorithms are intended for standard language and are not handling these variations efficiently. For instance, stemming the word "perfect" to "perfect" might disinvest the anticipated positive sentiment.

Short and context-dependent messages: Tweets are restricted to 280 characters which frequently indicates context-dependent and shortened text. Stemming might not deliver substantial remunerations in a few cases, using the meaning and context of a tweet are often reliant on the whole message, as well as phrases, specific words, and even emoticons.

Loss of data: Stemming can outcome in the loss of info by dropping words to their base forms. In cyberbullying recognition, refined disparities in words or spellings may be

significant pointers of insulting or offensive language. By adding stemming, these variations could be misplaced, possibly decreasing the model's capability to recognize occurrences of cyberbullying precisely.

Although stemming may not be the most active practice for cyberbullying finding on tweets, it's worth noting that these pre-processing stages like eliminating URLs, special characters, and mentions are still appreciated to clean the text documents. Moreover, the use of word embedding can apprehend semantic and appropriate material that can contribute to capturing variations of words and accepting the significance of the text.

Generally, whereas stemming can be convenient in some NLP tasks, its presentation in cyberbullying recognition on tweets based on LSTM models are not delivered important assistance and might even outcome in damaging critical information. It's vital to sensibly deliberate the exclusive characteristics of tweets and the precise necessities of the cyberbullying discovery task at what time to determine pre-processing practices.

IV. Lemmatization

Lemmatization is an additional text normalization method that is used in cyberbullying detection resolutions on tweets based on LSTM models. Lemmatization objects to lessen words to the base or dictionary form, recognized as a lemma, by allowing for the word's context and part of speech. Unlike stemming, which simply eliminates prefixes and suffixes, lemmatization helps in the analysis of words morphologically to certify meaningful alterations.

Here are the stages of lemmatization that are applied in the context of cyberbullying recognition on tweets by using LSTM models:

Pre-processing: Earlier lemmatization, it is common to achieve pre-processing stages like eradicating special hashtags, URLs, characters, and mentions. These steps are helping clean the text and eradicate noise that is hindering the detection procedure.

Tagging of Part-of-speech: To achieve lemmatization precisely, the words in the tweet are allocated their consistent part-of-speech (POS) labels. POS tagging classifies every word as an adjective, verb, noun, etc., which is essential for defining the proper lemma.

Lemmatization: Constructed on the POS tags, every word in the tweet is converted into a dictionary or its base form using lemmatization methods. These procedures use language-

specific instructions, machine learning algorithms, or dictionaries to achieve the lemmatization procedure exactly.

Truncation and Padding: Afterward the lemmatization, related to the tokenization stage, padding, or truncation are applied to guarantee reliable input extents for the use of the LSTM model.

Embedding demonstration: Once lemmatization is finished, the lemmatized tokens are transformed into numerical representations using word embedding. These representations capture the semantic and contextual material for the words that are critical for the understanding of LSTM models of the tweet.

Training and prediction: With the help of embedded and lemmatized data, the LSTM model is trained using labeled instances to acquire patterns suggestive of cyberbullying. For the period of prediction, new tweets are lemmatized similarly. The LSTM model can create predictions around whether they comprise occurrences of cyberbullying based on the learned configurations.

V. Removal of Step words

The exclusion of stop words is a unique pre-processing stage in numerous tasks of text analysis, comprising cyberbullying finding on tweets based on LSTM models. Stop words are normally used words that are measured to have few semantic meanings and are regularly detached to lessen noise and progress the efficiency and usefulness of algorithms of text processing.

Here's how the deletion of stop words is applicable in the framework of cyberbullying detection on tweets consuming LSTM models:

Pre-processing: Earlier eradicating stop words, other pre-processing phases like eliminating hashtags, URLs, special characters, and mentions are normally completed for cleaning the text and eradicating inappropriate information.

Identification of Stop word: A set of stop words that are precise to the particular language is used. These stop words frequently contain prepositions, pronouns, articles, and other mutual words that do not transmit momentous meaning in the analysis.

Elimination of stop words: The stop words recognized in the earlier step are detached from the tweet's text. This can be attained by associating every word in the tweet with all sets of stop words and removing those that can match.

Stuffing and truncation: Afterward the elimination of stop words, like in the preceding phases, truncation or padding are applied to confirm dependable input measurements for the LSTM model.

Embedding demonstration: As soon as stop words are detached, the remaining meaningful tokens are distorted into numerical illustrations using word embedding. These illustrations apprehend the contextual and semantic evidence essential for the LSTM model to comprehend the tweet's content.

Prediction and Training: By the stop words detached and the embedded data are ready, the LSTM model can be qualified using labeled samples to acquire outlines suggestive of cyberbullying. Throughout prediction, new tweets are undertaking similar pre-processing stages, involving the elimination of stop words, and the LSTM model is making predictions grounded on the learned outlines.

The subtraction of stop words are helping to eradicate noise and diminishes the text dimensionality, possibly bettering the productivity and effectiveness of the LSTM model in cyberbullying recognition. By eradicating frequently happening but less instructive words, the model can emphasize more the content-higher terms that might be revealing the cyberbullying performance.

5.3 RESULT AND PERFORMANCE ANALYSIS

We tested the algorithm in different epochs. This dataset maintained Test: Train: Validation split of the dataset. Here are the results:

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Insulting**	333	1810	101	394	0.77	0.46	0.57	0.812
	Non-insulting	1810	333	394	101	0.82	0.95	0.88	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Insulting**	435	1701	210	292	0.67	0.60	0.63	0.809
	Non-insulting	1701	435	292	210	0.85	0.89	0.87	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
8	**Insulting**	473	1598	313	254	0.60	0.65	0.63	0.785
	Non-insulting	1598	473	254	313	0.86	0.84	0.85	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
10	**Insulting**	392	1752	159	335	0.71	0.54	0.61	0.813
	Non-insulting	1752	392	335	159	0.84	0.92	0.88	

Overall:

✓ We got the best result for epoch =10.

✓ We got the accuracy of 0.812736 And F1 score of 0.88, and 0.61.

NOTE:

True Positive (TP) = the number of data for which both the true value and the predicted value of the model are positive.

True Negative (TN) = the number of data for which both the true value and the predicted value of the model are negative.

False Positive (FP) = the number of data for which the truth value is negative but the predicted value of the model is positive.

False Negative (FN) = number of data for which the truth value is positive but the predicted value of the model is negative

Accuracy = how many predictions we got right = (True Positive + False Positive) / (Total number of data)

Precision = True Positive / (True Positive + False Positive)

Recall= True positive / (True positive + False negative)

F1 – score = 2* (precision * recall) / (precision + recall)

Chapter 6

CONCLUSION AND FUTURE WORK

In this article, our main goal was to inaugurate the convention of the LSTM neural network for the assignment of cyberbullying detection and at that time try to upsurge the accuracy of the LSTM neural network with the help of a freely available fasttext word embedding model. Afterward, robust testing using equally the methods and recording the results, I concluded that LSTM is an appropriate neural network for this kind of work. It was also demonstrated that providing the inputs in vector formula improved the accuracy of the model for certain cases. So fast text method was also helpful to surge the accuracy of this model.

In the result section it was noticeable that the Insulting class's F1 score is much lesser than the non-insulting class. This is the foremost shortcoming of this article.

Another inadequacy of the system was that, in some cases, our model failed to detect bullying.

LSTM-based cyberbullying exposure in tweets is a favorable method that controls the model's capability to detent sequential patterns and semantic demonstrations. Though it has its boundaries, with vigilant dataset and model improvements, LSTM can help in building active tools for opposing cyberbullying on social media stages.

The compensations for using LSTM for cyberbullying recognition comprise:

Catching temporal dependencies: LSTM can detect the successive nature of language by memorizing material from preceding time stages. This is vital for accepting the context of all tweets, as cyberbullying frequently includes a sequence of connected messages.

Managing variable-length input: Tweets can differ suggestively in length, but LSTM can progress input of diverse lengths successfully. It can spontaneously acclimatize to the changing lengths of tweets, assembling it appropriately for analyzing social media records.

Learning semantic illustrations: LSTM can acquire meaningful demonstrations of words and phrases, taking their semantic possessions. This allows the model to identify subtle linguistic signals and contextual gradations that are symptomatic of cyberbullying.

So some future research work based on this article necessarily looks into these inadequacies. There are numerous future scopes accessible for overcoming these shortcomings. In the purpose of future work, any researcher may use a few other popular RNN-based neural networks like GRU, BiLSTM, and BERT models and associate them with the LSTM model. A precise vital inventiveness can be done by nurturing the respective language's fast text model with the bullying words of that individual language. This will be leading to the reduction of null embedding of some bully words. Further available embedding can be also recycled to vectorize the words existing in a tweet.

We have faith that this article can be advantageous for future research in cyberbullying detection.

References

[1] Hani, J., Mohamed, N., Ahmed, M., Emad, Z., Amer, E. and Ammar, M., 2019. Social media cyberbullying detection using machine learning. *International Journal of Advanced Computer Science and Applications*, *10*(5).

[2] Kelly Reynolds, April Kontostathis, and Lynne Edwards. Using machine learning to detect cyberbullying. In 2011 10th International Conference on Machine Learning and Applications and Workshops, volume 2, pages 241–244. IEEE, 2011.

[3] Talpur, B.A. and O'Sullivan, D., 2020. Cyberbullying severity detection: A machine learning approach. *PloS one*, *15*(10), p.e0240924.

[4] Rezvan M, Shekarpour S, Balasuriya L, Thirunarayan K, Shalin VL, Sheth A. A Quality Type-aware Annotated Corpus and Lexicon for Harassment Research. Proceedings of the 10th ACM Conference on Web Science. New York, NY, USA: ACM; 2018. pp. 33–36.

[5] Iwendi, C., Srivastava, G., Khan, S. and Maddikunta, P.K.R., 2020. Cyberbullying detection solutions based on deep learning architectures. *Multimedia Systems*, pp.1-14.

[6] Al-Ajlan, M.A. and Ykhlef, M., 2018. Deep learning algorithm for cyberbullying detection. *International Journal of Advanced Computer Science and Applications*, *9*(9).

[7] Murnion, S., Buchanan, W.J., Smales, A. and Russell, G., 2018. Machine learning and semantic analysis of in-game chat for cyberbullying. *Computers & Security*, *76*, pp.197-213.

[8] Ahmed, M.F., Mahmud, Z., Biash, Z.T., Ryen, A.A.N., Hossain, A. and Ashraf, F.B., 2021. Cyberbullying detection using deep neural network from social media comments in bangla language. *arXiv preprint arXiv:2106.04506*.

[9] Ahmed, Md Faisal, et al. (2021), "Bangla Online Comments Dataset", Mendeley Data, V1, doi: 10.17632/9xjx8twk8p.

[10] Muneer, A. and Fati, S.M., 2020. A comparative analysis of machine learning techniques for cyberbullying detection on twitter. *Future Internet*, *12*(11), p.187.

[11] Murshed, B.A.H., Abawajy, J., Mallappa, S., Saif, M.A.N. and Al-Ariki, H.D.E., 2022. DEA-RNN: A hybrid deep learning approach for cyberbullying detection in Twitter social media platform. *IEEE Access*, *10*, pp.25857-25871.

[12] Ghosh, S., Chaki, A. and Kudeshia, A., 2021, April. Cyberbully detection using 1D-CNN and LSTM. In *Proceedings of International Conference on Communication, Circuits, and Systems: IC3S 2020* (pp. 295-301). Singapore: Springer Singapore.

[13] Dewani, A., Memon, M.A. and Bhatti, S., 2021. Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data. *Journal of big data*, *8*(1), p.160.

[14] Gada, M., Damania, K. and Sankhe, S., 2021, January. Cyberbullying detection using lstm-cnn architecture and its applications. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-6). IEEE.

[15] Fang, Y., Yang, S., Zhao, B. and Huang, C., 2021. Cyberbullying detection in social networks using Bi-gru with self-attention mechanism. *Information*, *12*(4), p.171.

[16] Kumar, A. and Sachdeva, N., 2022. Multi-input integrative learning using deep neural networks and transfer learning for cyberbullying detection in real-time code-mix data. *Multimedia systems*, *28*(6), pp.2027-2041.

[17] Hasan, M.T., Hossain, M.A.E., Mukta, M.S.H., Akter, A., Ahmed, M. and Islam, S., 2023. A Review on Deep-Learning-Based Cyberbullying Detection. *Future Internet*, *15*(5), p.179.

[18] Maity, K., Saha, S. and Bhattacharyya, P., 2022, August. Cyberbullying Detection in Code-Mixed Languages: Dataset and Techniques. In *2022 26th International Conference on Pattern Recognition (ICPR)* (pp. 1692-1698). IEEE.

[19] Yadav, J., Kumar, D. and Chauhan, D., 2020, July. Cyberbullying detection using pre-trained bert model. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1096-1100). IEEE.

[20] Al-Hashedi, M., Soon, L.K. and Goh, H.N., 2019, November. Cyberbullying detection using deep learning and word embeddings: An empirical study. In *Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems* (pp. 17-21).

[21] Singh, N.K., Singh, P. and Chand, S., 2022, November. Deep Learning based Methods for Cyberbullying Detection on Social Media. In *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 521-525). IEEE.

[22] Shanto, S.B., Islam, M.J. and Samad, M.A., 2023, February. Cyberbullying Detection using Deep Learning Techniques on Bangla Facebook Comments. In *2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)* (pp. 1-7). IEEE.

[23]Chia, Z.L., Ptaszynski, M., Masui, F., Leliwa, G. and Wroczynski, M., 2021. Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection. *Information Processing & Management*, *58*(4), p.102600.

[24] https://www.kaggle.com/competitions/detecting-insults-in-social-commentary/data