CYBERBULLYING DETECTION USING SENTIMENT ANALYSIS IN SOCIAL MEDIA

A MINI PROJECT REPORT

Submitted by

AKASH NIXON	**(190501011)**
RAJSUDHAN M	**(190501094)**
RAKESH E	**(190501095)**
SANDHYA V	**(190501108)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)
ANNA UNIVERSITY :: CHENNAI 600 025

DECEMBER 2022

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)
ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"CYBERBULLYING DETECTION USING SENTIMENT ANALYSIS IN SOCIAL MEDIA"** is the bonafide work of **"AKASH NIXON (190501011), RAJSUDHAN M (190501094), RAKESH E (190501095)** and **SANDHYA V (190501108)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.R.ANITHA	**Dr. R. JAYABHADURI**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	PROFESSOR
COMPUTER SCIENCE & ENGG	**COMPUTER SCIENCE & ENGG**

Submitted for the project viva-voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Cyberbullying and aggressiveness have recently become significant issues that communities must address due to the extensive use of social media platforms worldwide, particularly among young people. People can use these platforms in a variety of ways to harass and threaten members of their communities. As a result, it is now more likely that cyber threats will materialize and proliferate. To fight prejudice, it is crucial for law enforcement organizations to identify bullying tweets. These organizations have lagged in terms of technology. Neural networks surpass several prominent machine learning methods for text classification with excellent accuracy because of various advantages in neural networks. To identify tweets that contains cyberbullying, this research work aims to compare nine text classification algorithms namely, three machine learning models and six shallow neural network models for Corona NLP Twitter dataset. According to the comparative study's findings, Bidirectional Encoder Representations from Transformers consistently outperforms other text classification models in terms of Accuracy, Precision, Recall, and F1 scores, with scores as high as 90%, 92%, 90%, and 91%, respectively.

ACKNOWLEDGEMENT

We thank our Principal **Dr. S. Ganesh Vaidyanathan,** Sri Venkateswara College of Engineering for being the source of inspiration throughout our study in this college.

We express our sincere thanks to **Dr. R. Anitha,** Head of the Department, Computer Science and Engineering for her encouragement accorded to carry this project.

With profound respect, we express our deep sense of gratitude and sincere thanks to our guide **Dr. R. Jayabhaduri**, **Professor**, for her valuable guidance and suggestions throughout this project.

We are also thankful to our Mini Project Coordinators **Dr. R. Jayabhaduri, Professor, Dr. N. Revathi, Associate Professor** and **Ms. V. Rajalakshmi, Associate Professor** for their continual support and assistance.

We thank our family and friends for their support and encouragement throughout the course of our graduate studies.

AKASH NIXON
RAJSUDHAN M
RAKESH E
SANDHYA V

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
Bi-GRU	Bidirectional Gated Recurrent Units
Bi-LSTM	Bidirectional Long Short Term Memory Networks
CNN	Convolution Neural Network
GRU	Gated Recurrent Units
IDF	Inverse Document Frequency
LR	Logistic Regression
LSTM	Long Short Term Memory Networks
NB	Naive Bayes
NLP	Natural Language Processing
NN	Neural Network
RNN	Recurrent Neural Networks
SA	Sentiment Analysis
SPC	Spanish Cyberbullying Prevention System
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency

CHAPTER 1

INTRODUCTION

1.1 CYBERBULLYING

Cyberbullying is the use of technology to harass, threaten, embarrass, or target another person. Online threats and mean, aggressive, or rude texts, tweets, posts, or messages all count. So does posting personal information, pictures, or videos designed to hurt or embarrass someone else.

Cyberbullying is bullying with the use of digital technologies. It can take place on social media, messaging platforms, gaming platforms and mobile phones. It is repeated behaviour, aimed at scaring, angering or shaming those who are targeted. Examples include: spreading lies about or posting embarrassing photos or videos of someone on social media, sending hurtful, abusive or threatening messages, images or videos via messaging platforms, impersonating someone and sending mean messages to others on their behalf or through fake accounts. Face-to-face bullying and cyberbullying can often happen alongside each other. But cyberbullying leaves a digital footprint - a record that can prove useful and provide evidence to help stop the abuse.

Cyberbullying takes place across virtually every social media platforms such as Twitter, Youtube, Whatsapp, Snapchat, Facebook, Instagram, etc. Figure 1.1 depicts people cyberbullied across various social media platforms.

This project focuses on Cyberbullying in Twitter since there is a lack of research into what types of advice and support are available in tweets for cyberbullying victims in Twitter.

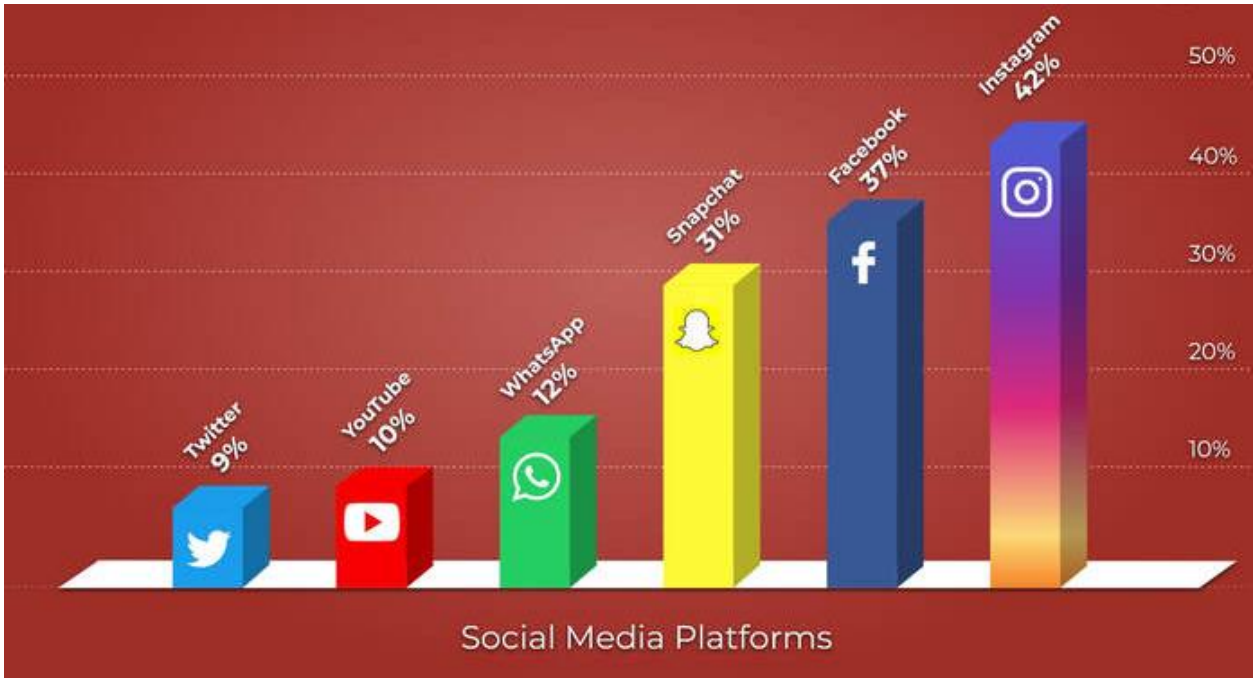


Figure 1.1 Percentage of People Cyberbullied Across Social Media Platforms

1.1.1 DIFFERENT TYPES OF CYBERBULLYING

Depending on the platform, bully, and the intention of harm, it can take on many different forms such as:

Exclusion refers to intentionally leaving the child out of online groups, conversations, forums, or communities. The intention is to make the child feel rejected or unwanted.

Doxing refers to revealing child's personal information like their phone number, address, or school without consent. Sometimes, the bully already knows the person, so they use that information to put them in danger. Other times, anonymous bullies can connect various details to glean information about their identity.

Cyberstalking includes pervasive, intense harassment that may consist of physical threats to the child's safety or well-being. Adults may cyberstalk children with the intent of meeting and sexually assaulting them. For this reason, cyberstalking can be one of the most dangerous forms of cyberbullying.

Trolling refers to direct insults or off-topic, controversial statements designed to stir a reaction from the child. Trolls aim to attack people and provoke them to react in the same way.

Gossiping includes bullies that may post, send, or share false gossip to harm child's reputation or impact their relationship with another person.

Masquerading - Bullies may construct a fake online identity to harm any child by creating fake social media profiles, emails, or usernames. Sometimes, they "catfish" to lure their victim into forming a fake relationship with them.

Dissing refers to sending or posting cruel information about the child online. Bullies may share items like videos, photos, messages, or comments with the intention of putting the child down.

Framing happens when a bully uses social media platforms to share inappropriate content under the child's name. The intention is to damage someone's reputation. For example, the bully might change their picture to an explicit one or post racist or crude content.

Impersonating involves bullies that may hack into someone's online accounts and pretend to be the victim by sharing or sending embarrassing or inappropriate material to other people.

1.1.2 MAIN EFFECTS OF CYBERBULLYING

Cyberbullying can have a substantial negative impact on the victims either physically i.e., feeling exhausted from not getting enough sleep or experiencing symptoms like headaches and stomachaches or emotionally i.e., feeling guilty or losing interest in the things which they enjoy. If they feel harassed or mocked by others, they can be reluctant to speak out or try to resolve the issue. In extreme cases, cyberbullying might even lead to suicides. Being able to use the internet at any time gives the victim of cyberbullying the impression that he is under assault everywhere. The victim may have mental,

bodily, and emotional repercussions. Psychological issues including despair, loneliness, low self-esteem, phobias of schools, and social anxiety are common in cyberbullied victims. There are many negative repercussions of cyberbullying on humans. Problems might be solved, though, enabling people to regain their confidence and health.

1.2 TWITTER

Twitter is an open social network that people use to converse with each other in short messages, known as tweet. Launched in 2006, Twitter is most popular with millennials and young professionals. Whether it's sharing breaking news, posting updates about their company or following their favourite celebrities, people are using Twitter to connect with others and to discover new things every day. Twitter encourages users to follow and interact with different individuals, brands and media outlets, creating a real-time stream of messages tailored to their interests. People post updates, photos, videos and links to Twitter as they are happening, enabling insightful, real-time, search results. However, it has experienced significant growth of users in all age groups since its launch.

1.3 CYBERBULLYING IN TWITTER

Cyberbullying takes place in Twitter when a person posts tweets involving abusive and negative words such as 'hate', 'hell', 'sucks', 'die', etc or racist words involving colour of a person or impersonating words which is used

against another person. They are considered to be Bullying Tweets as it gives a high negative sentiment.

Figure 1.2 represents normal tweets where a person has tweeted about his favourite football club and next tweet is about his opinion towards President Biden and Donald Trump. These tweets are considered as Non Bullying since they do not contain any hate or abusive words.

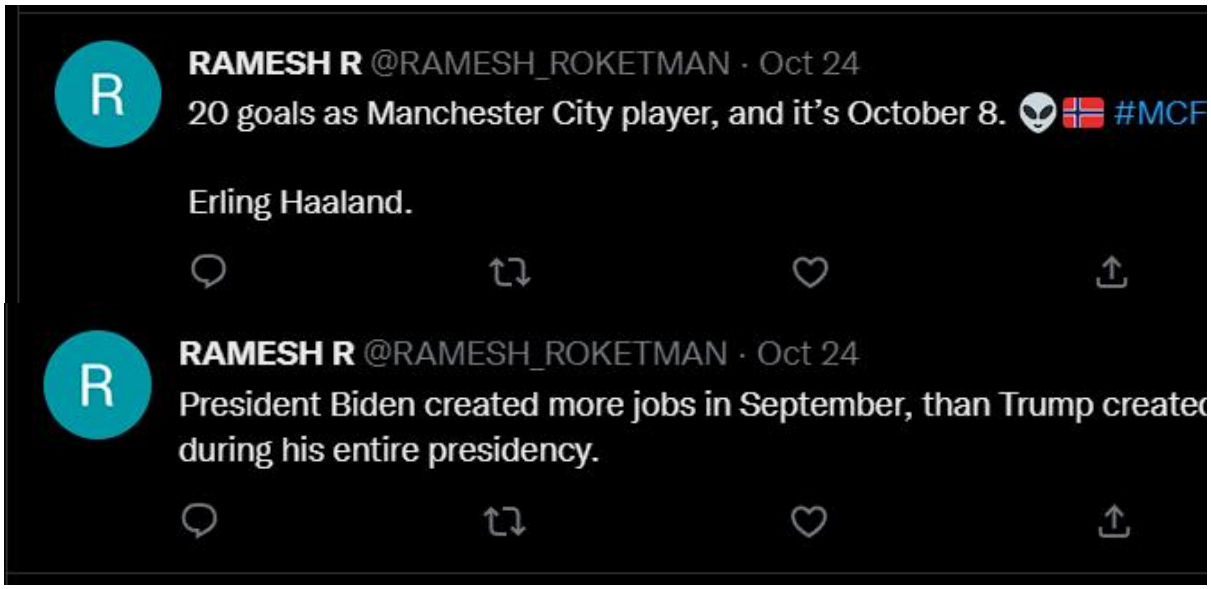


Figure 1.2 Examples of Non Bullying Tweet

Figure 1.3 represents Bullying Tweets where the person has shown hatred comments towards another person by using abusive words and discriminating the work done by that person.

Figure 1.3 Examples of Bullying Tweets

1.4 SENTIMENT ANALYSIS

Sentiment analysis (SA) is the automatic process of classifying text data according to their polarity, such as positive, negative and neutral. Sentiment analysis of social media content has become one of the growing areas of research in machine learning as it provides the ability to detect Cyberbullying in real-time. Since humans express their thoughts and feelings more openly than ever before, sentiment analysis is fast becoming an essential tool to monitor and understand sentiment in all types of data.

Sentiment analysis for tweets can be implemented in 3 different ways:

- Manual: Systems that analyze sentiment using a set of manual rules.
- Automatic: System that automatically detects opinions.
- Hybrid: Combination of both rule-based and automated methods.

Expanding the polarity categories to include different levels of positive and negative sentiments, tweets can be classified as:

- Very positive - Non Bullying
- Positive - Non Bullying
- Neutral - Non Bullying
- Negative - Bullying
- Very negative - Bullying

Text classification problems like sentimental analysis can be achieved in a number of ways using a number of algorithms. These are majorly divided into two main categories:

- A bag of word model: In this case, all the sentences in the dataset are tokenized to form a bag of words that denotes the vocabulary. Now each individual sentence or sample in the dataset is represented by that bag of words vector. This vector is called the feature vector. For example, ‘It is a sunny day’, and ‘The Sun rises in east’ are two sentences. The bag of words would be all the words in both the sentences uniquely.
- Time series approach: In this method, each word is represented by an Individual vector. So, a sentence is represented as a vector of vectors.

1.5 TEXT CLASSIFICATION ALGORITHMS FOR SENTIMENT ANALYSIS

Naive Bayes (NB) algorithm is used for probabilistic classification. It is widely used for various practical applications due to its efficiency in reducing computational costs. It is a scalable algorithm applicable to large sized datasets, also resulting in high classification accuracies. It principally assumes that a feature in a category is independent of its presence in another category.

Support Vector Machine (SVM) is a supervised algorithm that uses the separation margin between data points of classes as a classification criterion. The original m-dimensional feature space is reduced to a user-defined dimensional space. Support vectors are then determined to optimize the margin distance among data points of different categories. The algorithm automatically determines these support vectors found nearest to the separating margins (hyperplanes).

Logistic Regression (LR) is another statistical algorithm that works on predicting probabilities rather than classes. The logistic function is used to form a hyperplane in order to classify data points in the given classes. Textual features are input to the algorithm employed to generate forecasts about a data point belonging to a particular class.

Convolution Neural Networks (CNN) can extract an area of features from global information, with the convolution operation, a piece of data information can be extract together as the features, and it is able to consider the

relationship among these features. For computer vision, such as image analysis, it is able to extract a part of pixel data information, not only extract the pixels one by one, the features information can be extracted piece by piece, the piece contains multi pixels data information; when we transfer the text into matrix, it can also be considered as same as an image pixels' matrix, so we can do the same operation to the text data to make the input features to the model can be trained in another effective way.

Long Short-Term Memory Networks **(**LSTM) are a special type of Recurrent Neural Networks (RNNs) that are more advantageous compared to RNNs in terms of information retention. LSTMs overcome the problem of vanishing gradient descent encountered in traditional RNNs. LSTMs are highly preferred for tasks such as text classification and predictive modeling due to their extensive memory capacity. Such a network selectively decides what information is necessary to be transferred to further neurons and which data can be forgotten or omitted. These networks employ backpropagation and a gated mechanism.

Bidirectional LSTM (Bi-LSTM) is a robust mechanism used to enhance backpropagation in LSTM networks. While the information in an LSTM travels unidirectionally, Bi-LSTM allows data to move in both forward and backward directions. A Bi-LSTM processes inputs both reverse and serially. Architecturally, it is simply combining two LSTMs but in opposite directions. This allows the network to remember information from past to future by using the forward layer and future to past layer by using the backward LSTM layer.

Gated Recurrent Units (GRU) are also a type of RNN with a gated mechanism designed to deal with the vanishing and exploding gradient problem. These provide more testing accuracies than traditional RNNs because of the ability to remember long-term dependencies. GRUs are a more straightforward and dynamic version of LSTM networks specifically designed for updating or resetting information in their memory cells. The network constitutes an update gate that combines input and a forget gate present in LSTMs. Additionally, there is a reset gate for refreshing the memory contents.These are lightweight and have fewer parameters than LSTMs.

Bidirectional GRU (Bi-GRU) is a dual-layered structure similar to a Bi-LSTM with forward and backward neural networks. The idea of this structure is to transfer entire contextual information from the input to the output layer. Similarly to a bidirectional LSTM, in a Bi-GRU, the input information travels through a neural network in the forward direction and a neural network in the backward direction. The outputs from both these forward and backward layers are fused to provide the final output.

Bidirectional Encoder Representations from Transformers (BERT) makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms - an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT’s goal is to generate a language model, only the encoder mechanism is necessary. As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of

words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word). The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.

1.6 SCOPE

The scope of this project is to identify the most suitable Text Classification Algorithm for Sentiment Analysis in Twitter out of many well known Text Classification Algorithms using the Corona NLP Twitter Dataset. By doing so, the proposed solution can detect online bullies in Twitter to stay alert of in real time thereby creating a more secure social media experience.

1.7 CHALLENGES

Dataset selection for this project is a very challenging task since it is chosen in such a way that the dataset is not explored and not used by other researchers across the world as per the literature survey. Web Scraping of the tweets is another challenging task because the Twitter web page renders the content dynamically therefore increasing the difficulty in scraping the tweets.

CHAPTER 2

LITERATURE REVIEW

2.1 REVIEW OF RESEARCH PAPERS

Chahat Raj, Ayush Agarwal, Gnana Bharathy, Bhuva Narayan and Mukesh Prasad's work, et al. [1] on Hybrid Models Based on Machine Learning and Natural Language Processing Techniques state that traditional machine learning algorithms pose a disadvantage due to the inability to yield high accuracies on vast volumes of data for supervised classification. They employed four popular machine learning approaches for cyberbullying detection: XGBoost, Naïve Bayes, SVM, and Logistic Regression. The need for neural networks arises due to large dataset sizes that most of the traditional machine learning algorithms fail to accommodate. Neural networks also offer robustness and higher classification results. By comparing the architectures of popular neural networks for cyberbullying classification like CNN, LSTM, Bi- LSTM, GRU, Bi-GRU, CNN-BiLSTM, and Attention-BiLSTM they concluded that all common metrics indicate good performance. Accuracy is over 90% for the proposed shallow neural networks and over 80% for all traditional machine learning models across all datasets. The neural network approaches demonstrate better performance than traditional machine learning algorithms.

John Hani, Mohamed Nashaat, Mostafa Ahmed, Zeyad Emad, Eslam Amer and Ammar Mohammed, et al. [2] propose a supervised machine learning approach for detecting and preventing cyberbullying. Several classifiers are

used to train and recognize bullying actions. The evaluation of the proposed approach on cyberbullying dataset shows that Neural Network (NN) performs better and achieves accuracy of 92.8% and SVM achieves 90.3. Also, NN outperforms other classifiers of similar work on the same dataset. The proposed approach is evaluated on a cyberbullying dataset from kaggle which is collected and labeled by the authors Kelly Reynolds in their paper. The performance of SVM and Neural Network classifiers are compared on both TF-IDF and sentiment analysis feature extraction methods. Experiments were made on different n-gram language models. The SVM classifier achieved the highest percentage using 4-Gram with accuracy 90.3% while the NN achieved highest accuracy using 3-Gram with accuracy 92.8%. It is found that the average accuracy of all n-gram models of NN achieves 91.76%, while the average accuracy of all n-gram models of SVM achieves 89.87%. They compared their work with another related work that used the same dataset, finding that their Neural Network outperformed their classifiers in terms of accuracy and f-score. However, detecting cyberbullying patterns are limited by the size of training data. Thus, a larger amount of cyberbullying data is needed to improve the performance. Hence, deep learning techniques will be suitable in the larger data as they are proven to outperform machine learning approaches over larger size data.

Andrea Perera and Pumudu Fernando, et al. [3] used a Twitter dataset, where globally people have seen the most amount of cyberbullying from Internet Archive. The author only stored the relevant data from the Twitter JSON object such as the text of the tweet, tweet ID, in reply to status ID and retweeted status ID, and used a SQL database to store them. The noise of the

text and unnecessary features can negatively affect the overall performance of the model. Therefore, For each sentence web links, emojis, extra punctuation and unwanted, repeated characters were removed. Next, they converted contractions (“I’m”, “there’re”) and abbreviations into converting them to formal language. The author adds the most common abbreviation to a list and tokenizes each sentence, to extract individual words in a sentence. Search abbreviation words in the list and replace them with the formal language word. Removed some of the pronouns from the English stop words list such as “you”,” she”,” they” etc. Converted all the text into lowercase in order to reduce the document space problem. 30% of the manually labeled Twitter dataset which is randomly selected is used for testing purposes, while the remaining 70% is applied to train the model. The evaluation parameters were accuracy, precision, recall and F1-Score. In this paper, the author has presented the proposed solution which uses natural language processing techniques and supervised machine learning to detect cyberbullying accurately along with the themes/categories associated with cyberbullying such as racist, sexual, physical mean, swear and other. In order to create an accurate cyberbullying detection, the author has proposed a system that will adapt to the language changes with a new hypothesis. The proposed solution resulted in 74.50% accuracy along with 74% precision, 74% recall and 74% F1 Score. As this research is still ongoing, the author is working on getting higher accuracy.

G. A. León-Paredes, et al. [4] have explained the development of a cyberbullying detection model using Natural Language Processing (NLP) and Machine Learning (ML). A Spanish cyberbullying Prevention System (SPC) is developed by applying machine learning techniques Naive Bayes, Support

Vector Machine, and Logistic Regression. The dataset used for this research is extracted from Twitter. The maximum accuracy of 93% is achieved with the help of three techniques used. The cases of cyberbullying detected with the help of this system presented an accuracy of 80% to 91% on average. Stemming and lemmatization techniques in NLP can be implemented to further increase the accuracy of the system. Such a model can also be implemented for detection in English and local languages if possible.

J. Yadav, et al. [5] proposes a new approach to cyberbullying detection in social media platforms by using the BERT model with a single linear neural network layer on top as a classifier. The model is trained and evaluated on the Formspring forum and Wikipedia dataset. The proposed model gave a performance accuracy of 98% for the Form spring dataset and of 96% for the Wikipedia dataset which is relatively high from the previously used models. The proposed model gave better results for the Wikipedia dataset due to its large size without the need for oversampling whereas the Form spring dataset needed oversampling. Google researchers has recently developed a language learning model called BERT, which can generate contextual embeddings and is also able to produce task specific embeddings for classification. A new approach is proposed to cyberbullying detection in social media platforms by using the novel pre-trained BERT model with a single linear neural network layer on top as a classifier, which improves over the existing results. The model is trained and evaluated on two social media datasets of which one dataset is small size and the second dataset is relatively larger size.

Amirita Dewani, Mohsin Ali Memon and Sania Bhatti, et al. [6] implemented cyber bullying detection using advanced preprocessing techniques & deep learning architecture for Roman Urdu data. This paper addresses toxicity/cyberbullying detection problem in Roman Urdu language using deep learning techniques and advanced preprocessing methods including usage of lexicons/resource that are typically developed to accomplish this work. They had performed extensive preprocessing on Roman Urdu microtext. This typically involves formation of Roman Urdu slang phrase dictionary and mapping slangs after tokenization. They also eliminated cyberbullying domain specific stop words for dimensionality reduction of corpus. The unstructured data is further processed to handle encoded text formats and metadata/non-linguistic features. Furthermore, they performed extensive experiments by implementing RNN-LSTM, RNN-BiLSTM and CNN models varying epochs executions, model layers and tuning hyperparameters to analyze and uncover cyberbullying textual patterns in Roman Urdu. The efficiency and performance of models were evaluated using different metrics to present the comparative analysis. Results highlight that RNN-LSTM and RNN-BiLSTM performed best and achieved validation accuracy of 85.5 and 85% whereas F1 score is 0.7 and 0.67 respectively over aggression class.

R. R. Dalvi, et al. [7] suggests a method to detect and prevent Internet exploitation on Twitter using Supervised classification Machine Learning algorithms. In this research, the live Twitter API is used to collect tweets and form datasets. The proposed model tests both Support Vector Machine and Naive Bayes on the collected datasets. To extract the feature, they have used the TFIDF vectorizer. The results show that the accuracy of the cyberbullying

model based on the Support Vector Machine is almost 71.25% that is better than the Naive Bayes which is almost 52.75%.

Fatma Elsafoury, Stamos Katsigiannis, Steven R. Wilson and Naeem Ramzan, et al. [8] implemented a deep neural network model called BERT. In this work, they conducted a series of experiments on five datasets to analyze the performance of BERT on the task of cyberbullying detection. Results showed that BERT outperforms other commonly used deep learning models on multiple cyberbullying-related datasets. In addition, even though the patterns of the attention weights of fine-tuned BERT are different from those of BERT without fine-tuning, results showed that attention weights are not meaningful when it comes to the model's prediction, and that BERT captures syntactical biases in the datasets. They used five cyberbullying-related datasets of varying sizes from several social media sources that contained different types of cyberbullying: Twitter-Racism, a collection of Twitter messages containing tweets that are labelled as racist or not, TwitterSexism, Twitter messages containing tweets labelled as sexist or not, Kaggle-Insult, a dataset that contains social media comments that are labelled as insulting or not, WTP-Toxicity, a collection of conversations from Wikipedia Talk Pages (WTP) annotated as friendly or toxic, and WTP-Aggression, conversations from WTP annotated as friendly or aggressive. Their findings indicate that the understanding of cyberbullying detection using pre-trained models like BERT can be improved by using gradient-based feature importance methods, which can assist in revealing some of the biases in the model or the dataset, thus helping towards fair detection of cyberbullying.

Aditya Desai, Shashank Kalaskar, Omkar Kumbhar, and Rashmi Dhumal, et al. [9] implemented a deep learning model called BERT. The authors proposed a model based on various features that should be considered while detecting cyberbullying and implement a few features with the help of a bidirectional deep learning model called BERT. In this paper, a method to detect cyberbullying on social media is proposed that is not just based on the sentimental analysis but also considers the syntactic, semantic, and sarcastic nature of the sentence before classifying it as hate speech. A cyberbullying detection model is proposed based on transformers. Similar to RNN, transformers can also be used to solve a wide variety of NLP(Natural Language Processing) problems like translation and text summarization as they can take sequential data as input. The BERT is a recent paper published by researchers at Google AI Language. BERT, Bidirectional Encoder from Transformers is a bidirectional model that is pre-trained on unlabeled texts from both left and right directions to understand the meaning of both contexts. Since it's a bidirectional model it aims to understand the meaning of the word from both the left and the right context to derive a better meaning during the training phase. They used a tweet from Twitter with the trace of bullying and applied it to their model. The accuracy of the SVM and Naive Bayes model is 71.25% and 52.70% respectively, when applied on the same dataset. The result shows better accuracy when using the BERT model for sentiment analysis on the Twitter dataset. Our proposed model gave a better accuracy of 91.90% when applied to the Twitter dataset for the sentimental analysis which can be considered as a greater result when compared to the traditional machine learning models used on similar datasets. The BERT model can achieve more accurate results if provided with a large dataset.

Jalal Omer Atoum's work, et al. [10] on Cyberbullying Detection Through Sentiment Analysis proposes a SA model analyzes, mines, and classifies tweets. Several preprocessing stages must be done on the collected tweets for the SA process to be more effective. To evaluate the performance of the machine learning methods used in this research; namely the Naive Bayes (NB) and the Support Vector Machine (SVM), they had collected a total of 5628 tweets (Positive-cyberbullying, negative-no cyberbullying, and neutral). This set of tweets is manually classified into 1187 cyberbullying tweets, 2342 with no cyberbullying tweets and the remaining 2099 are neutral tweets. These tweets have gone through several phases of cleaning, annotations, normalization, tokenization, named entity recognition, removing stopped words, stemming and n-gram, and features selection. The results of the conducted experiments have indicated that SVM classifiers have outperformed NB classifiers in almost all performance measures over all language models. Specifically, SVM classifiers have achieved an average accuracy value of 92.02%, while, the NB classifiers have achieved an average accuracy of 81.1 on the 4-gram language model.

2.2 SUMMARY OF EXISTING PAPERS

Cyberbullying detection can be implemented using machine learning techniques, neural networks and deep learning techniques. Based on the literature survey done, it can be seen that the BERT model identifies cyberbullying tweets with the highest accuracy. The proposal of shallow neural networks moderates the need of complex deep neural networks, thus economizing resources. It is observed that neural networks highly outperform traditional machine learning algorithms. It is established that bidirectional

neural networks perform better in all scenarios. The attention mechanism is also observed to perform exceptionally well. Traditional machine learning algorithms such as SVM, Naive Bayes, XGBoost and Logistic Regression provide lower results compared to the shallow neural networks. Overall, using bidirectional RNNs and attention-based models for further advances in cyberbullying detection is the most effective approach.

While considering just one of the features which is sentimental features the BERT model achieved 91.90% accuracy when trained over dual cycles which outperformed the traditional machine learning models. The BERT model can achieve more accurate results if provided with a large dataset. They can try to achieve even better results in the cyberbullying detection process if they consider all the features that is proposed in their research paper. Based on all the features an application can be created to detect the bullying traces and thus help in detecting and reporting such posts. A combination of other models on top of the BERT model can also be used in the future to create a state-of-the-art model for the specific NLP tasks in detecting cyberbullying.

CHAPTER 3

PROPOSED WORK

3.1 TWITTER DATASET

The Corona NLP Twitter dataset for training and testing from Kaggle contains a set of different tweets throughout the world. The tweets have been pulled from Twitter during Covid-19 and manual tagging has been done then. There are 6 attributes namely UserName, ScreenName, Location, TweetAt, OriginalTweet and Sentiment. The training dataset contains 41157 rows and there are no null values present in it. The names and usernames have been given codes to avoid any privacy concerns. Figure 3.1 shows Corona NLP Twitter Dataset from Kaggle.

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative

Figure 3.1 Corona NLP Twitter Dataset

The other attributes provides a clear view on the location and the content of the tweet with appropriate sentiment as well, furthermore it also gives a clear view on the negative and positive tweets segregated by location which is

depicted in Figure 3.2. The attribute Sentiment has 5 polarity values such as ‘Extremely Negative’, ‘Negative’, ‘Neutral’, ‘Positive’ and ‘Extremely Positive’ which is then mapped into ‘Bullying’ and ‘Non Bullying’. Only the OriginalTweet and Sentiment attributes are taken into consideration for model building since other attributes are only for analysis purposes and hence they are dropped.

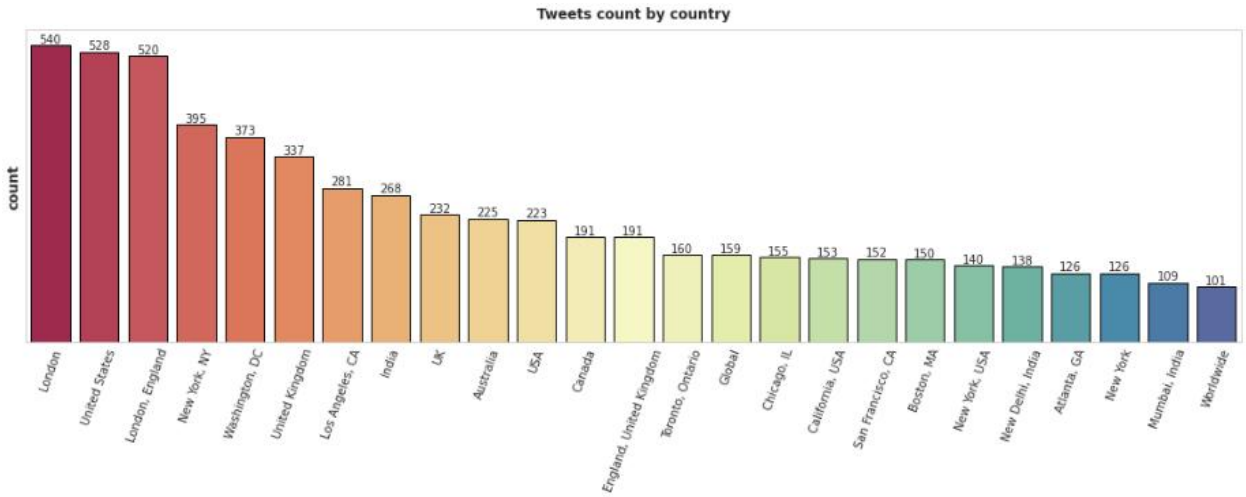


Figure 3.2 Country-wise Tweets Count

3.2 ARCHITECTURE DIAGRAM

Figure 3.3 illustrates the architecture diagram of Cyberbullying detection in Twitter where it contains modules such as Preprocessing, Feature Extraction, Supervised Text Classification Models, Model Evaluation and Classifying Tweets. Feature extraction algorithms such as TF-IDF and Tokenizers have been used for suitable models.

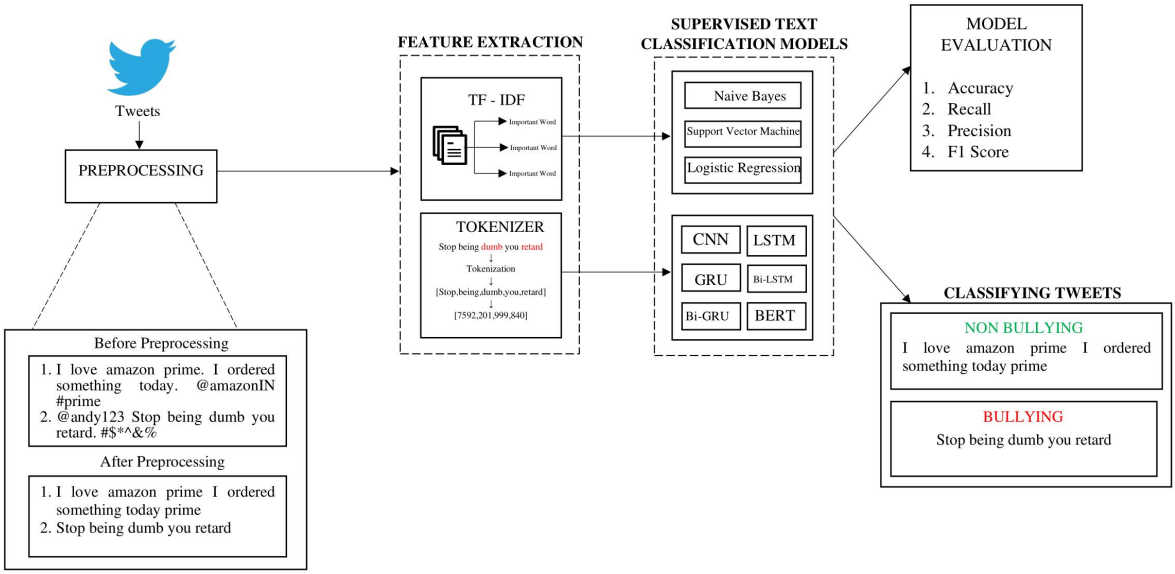


Figure 3.3 Architecture Diagram of Cyberbullying Detection in Tweets

3.2.1 PREPROCESSING TWEETS

Preprocessing the data becomes the first step in any Natural Language Processing task. When training a classifier with tweets data, there is a large amount of noise due to tweets' shortness, marks, irregular words etc. Preprocessing the tweets creates an impact on the classification model as it eliminates noisy data. The dataset also has linguistic diversity which has to be eliminated, because most people tweet in English, but some people tweet in their own language.

In tweets preprocessing, there are many steps like removing punctuations, urls, mentions, hashtags, emojis, converting all text to lowercase, removing unicode, newline characters and multiple spaces, etc and finally converting it to a uniform format.

3.2.2 FEATURE EXTRACTION

The following feature extraction algorithms have been employed for suitable supervised text classification models: Term Frequency - Inverse Document Frequency (TF-IDF) and Tokenizers.

3.2.2.1 TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY

The words which appear more often in a sentence such as “the”, “will”, and “you” are called stopwords has a lesser weight in textual data analysis and have very little significance. Instead, the words which are rare are the ones that actually help in distinguishing between the data, and carry more weight.

Taking two tweets as an example, Tweet 1: “The truck is driven on the highway”, Tweet 2: “The car is driven on the road”. Each tweet is in different documents. The TF-IDF for the above two documents is calculated in Table 3.1 using the below Equation 3.1, Equation 3.2 and Equation 3.3.

$$TF = \frac{\text{no. of times the term appears in the document}}{\text{total no. of terms in the document}} \quad \ldots \textbf{(3.1)}$$

$$IDF = log\left(\frac{\text{no. of the documents in the corpus}}{\text{no. of documents in the corpus contain the term}}\right) \quad \ldots \textbf{(3.2)}$$

$$\text{TF} = \text{TF} * \text{IDF} \quad \ldots \textbf{(3.3)}$$

Table 3.1 TF-IDF Calculation

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	0	0	0
Car	1/7	0	0.3	0.043	0
Truck	0	1/7	0.3	0	0.043
Is	1/7	1/7	0	0	0
Driven	1/7	1/7	0	0	0
On	1/7	1/7	0	0	0
The	1/7	1/7	0	0	0
Road	1/7	0	0.3	0.043	0
Highway	0	1/7	0.3	0	0.043

From Table 3.1, it is notable that TF-IDF of common words is zero, which shows they are not significant. On the other hand, the TF-IDF of “car” , “truck”, “road”, and “highway” are non-zero having more significance.

3.2.2.2 TOKENIZERS

Tokenizers are one of the core components of the NLP pipeline. Their purpose is to translate text into data that can be processed by the model. The tokenizer object allows the conversion from character strings to tokens understood by different models. Models can only process numbers, so tokenizers need to convert the text inputs to numerical data. The tokenizer takes each tweet as an input and returns two values - input IDs (numerical data) and

attention masks (binary data) which are the input tensors used by the neural network models. The required input tensors vary between different neural network models, and even different transformer use-cases.

3.2.3 SUPERVISED TEXT CLASSIFICATION MODELS

The following supervised text classification models have been trained for Cyberbullying detection in Twitter :

- Machine Learning Models:

  Naive Bayes, Support Vector Machine (SVM), Logistic Regression

- Neural Networks:

  Convolution Neural Network (CNN), Long Short Term Memory (LSTM), Gated Recurrent Networks (GRU), Bidirectional LSTM (Bi-LSTM), Bidirectional GRU (Bi-GRU), Bidirectional Encoder Representations from Transformers (BERT).

Pseudocode for machine learning models is given below where TF-IDF is used as the feature extraction algorithm:

Input : Training data T

Feature Extraction Algorithm : TF-IDF

For each : Model M in Naive Bayes, SVM, Logistic Regression

For each : Tweet T_w in Training data T

Get : Feature Extraction Algorithm F

Pass : Tweet T_w to F

For each : Extracted features E from F

Compare : Extracted features E to training data T
using Model M and store class in C

If : C is 0 , Tweet T_w is Bullying

Else if : C is 1, Tweet T_w is Non Bullying

The proposed shallow neural networks make use of Sigmoid activation function for binary classification of tweets with the classes 0 as Bullying and 1 as Non Bullying which is calculated using the Equation 3.4.

$$S(x) = \frac{1}{1+e^{-x}} \quad \ldots (3.4)$$

where,

$S(x)$ = Sigmoid function

e = Euler's number

Bidirectional Encoder Representations from Transformers (BERT) employs Softmax activation function for multi class classification of tweets to predict the scores for the classes Bullying and Non Bullying which is calculated using the Equation 3.5

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \ldots (3.5)$$

where,

σ	=	softmax
$\vec{z}$	=	input vector
e^{z_i}	=	standard exponential function for input vector
K	=	number of classes in the multi-class classifier
e^{z_j}	=	standard exponential function for output vector

Pseudocode for shallow neural networks is given below where Tokenizer is used for the feature extraction:

Input : Training data T

Feature Extraction Algorithm : Tokenizer

Activation Function : Sigmoid, Softmax

For each : Model M in CNN, LSTM, GRU, Bi-LSTM,

Bi-GRU, BERT

For each : Tweet T_w in Training data T

Get : Feature Extraction Algorithm F

Pass : Tweet T_w to F

For each : Extracted features E from F

Pass : Extracted features E to Model M

Output : Predict Score S from Activation Function

If : S > 0.5, Tweet T_w is Non Bullying

Else if : S < 0.5, Tweet T_w is Bullying

3.2.4 EVALUATION METRICS

Evaluation metrics are used to measure the quality of neural networks and machine learning models. It is very important to use multiple evaluation metrics to evaluate the model because a model may perform well using one measurement from one evaluation metric, but may perform poorly using another measurement from another evaluation metric. Using evaluation metrics are critical in ensuring that the model is operating correctly and optimally.

The confusion matrix is utilized for the performance evaluations of the methods used after the classification. For binary classification, the scheme of the confusion matrix is seen in Table 3.2. True Negative shows the number of negative examples classified accurately, True Positive indicates the number of positive examples classified accurately, False Positive indicates the number of actual negative examples classified as positive and False Negative is the number of actual positive examples classified as negative.

Table 3.2 Confusion Matrix

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

There are different types of Evaluation Metrics such as:

- Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Accuracy of the text classification for tweets is calculated using the Equation 3.6.

$$\text{Accuracy} = \frac{\text{True Positive}+\text{True Negative}}{\text{True Positive}+\text{False Positive}+\text{True Negative}+\text{False Negative}} \quad \textbf{. . . (3.6)}$$

- Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. Precision of the text classification for tweets is calculated using the Equation 3.7.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive}+\text{False Positive}} \quad \textbf{. . . (3.7)}$$

- Recall is the ratio of correctly predicted positive observations to the all observations in positive class. Recall of the text classification for tweets is calculated using the Equation 3.8.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive}+\text{False Negative}} \quad \textbf{. . . (3.8)}$$

- F1 score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. F1 score of the text classification for tweets is calculated using the Equation 3.9

$$\text{F1 Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad \ldots \textbf{(3.9)}$$

3.2.5 CLASSIFYING TWEETS

The binary classification of tweets is based on the classes and scores generated from the supervised text classification algorithms. The class labels are "Bullying" and "Non Bullying". After the tweet is passed into machine learning models, sentiment analysis is done and the model predicts either of the two classes: 0 - Bullying and 1 - Non Bullying.

The output generation is not the same when it comes to neural networks since the scores are generated from the output neurons based on the activation function used in the neural network. The values of these scores lies in a range of 0 to 1 instead of binary values 0 and 1 like in the case of machine learning models. If the score is greater than 0.5, the tweet is classified as Non Bullying, else if the score is less than 0.5, the tweet is classified as Bullying.

Considering the below tweets as the inputs,

Tweet 1 : "I love amazon prime I ordered something today prime".

Tweet 2 : "Stop being dumb you retard".

The classification models classifies Tweet 1 as Non Bullying and Tweet 2 as Bullying.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Processor : AMD Ryzen 5 3550H
- RAM : 8 GB
- Storage Space : 1 TB HDD

4.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 10 64-bit
- Web Browser : Google Chrome 107.0.5304.89
- Webdriver : ChromeDriver 107.0.5304.62
- IDE : Visual Studio Code (v 1.73)
- Programming Language : Python 3.10.x
- Web Application Framework : NodeJS, AngularJS, Flask
- Open Source Software Libraries : Tensorflow 2.10.0, Selenium 4.5.0, scikit-learn 1.1.3, transformers 4.24.0

4.2.1 VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a lightweight but powerful source code editor which runs on desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#,

Java, Python, PHP, Go, .NET). VS Code helps to be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Visual Studio Code includes an interactive debugger that can step through source code, inspect variables, view call stacks, and execute commands in the console.

4.2.2 TENSORFLOW

TensorFlow is an open-source software library developed by the Google Brain team for internal use. It enables differentiable and dataflow programming across various tasks. Being a symbolic math library, it is used for various machine learning applications such as neural networks etc. It is utilized at both research level, as well as production purposes. Prior experience with TensorFlow has become a prerequisite to work in the machine learning industry. Command for installing Tensorflow is ‘pip install tensorflow’.

4.2.3 TRANSFORMERS

The transformer package is provided by huggingface.io. It tries to solve the various challenges in the NLP field; it provides pre-trained models, tokenizers, configs, various APIs, ready-made pipelines for the inference etc. The transformers package gives the advantage of using pre-trained language models along with their data-processing tools. Most of the models are provided directly and made available in the library in PyTorch and TensorFlow. Transformers package requires TensorFlow or PyTorch to work, and it can train models in just some lines of code and pre-process the text data easily. Command for installing Transformers is ‘pip install transformers’.

4.2.4 ANGULAR JS

AngularJS is a JavaScript framework designed to extend the syntax of HTML. AngularJS empowers developers to build rich internet applications (RIAs) more easily. Where many JavaScript frameworks focus on expanding the capabilities of JavaScript itself, AngularJS instead provides methods for enhancing HTML. AngularJS expands the capabilities of HTML beyond being a simple markup language. By adding features such as data-binding, AngularJS allows developers to avoid the complex workarounds that are traditionally necessary when attempting to build responsive web applications with HTML front ends.

4.2.5 SELENIUM

Selenium is an open source tool which is used for automating the test cases carried out on web browsers or the web applications that are being tested using any web browser. It provides a single interface that lets programmers write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#. A Selenium WebDriver directly opens up a browser based on the specification declared through Selenium scripts or the Client APIs. Another key distinguishing feature is the fact that it leverages the functionality of the native operating system to control the browser instead of relying over the JavaScript commands to initiate and operate the browser. Command for installing Selenium is ‘pip install selenium’.

CHAPTER 5

IMPLEMENTATION MODULES

The overall system consists of the following modules:

1. Web Scraping
2. Tweets Collection and Preprocessing
3. Feature Extraction
4. Real-time Tweet Classification
5. Report Generation

5.1 WEB SCRAPING

The simple, structured format of Twitter and its various posting functions makes it relatively easy to navigate and scrape. Scraping tweets can yield many insights into sentiments, opinions and social media trends. Tweets and profile picture of an account is scraped dynamically after getting Twitter profile URL as an input using Selenium package. With the help of Xpaths in HTML DOM, patterns can be matched to a document structure for scraping data. It specifies the parts of a document in a tree structure manner where the parent node is written before the child node inside a pattern. Figure 5.1 shows a tweet posted by Elon Musk and Figure 5.2 shows the Xpath extraction of that particular tweet.

Figure 5.1 A Tweet Posted by Elon Musk

```
<div dir="auto" lang="en" class="css-901oao r-1nao33i r-
37j5jr r-1blvdjr r-16dba41 r-vrz42v r-bcqeeo r-bnwqim r-
qvutc0" id="id__xbgc6etzmob" data-testid="tweetText">…
</div> == $0
```

Figure 5.2 XPath Extraction From Tweets For Scraping

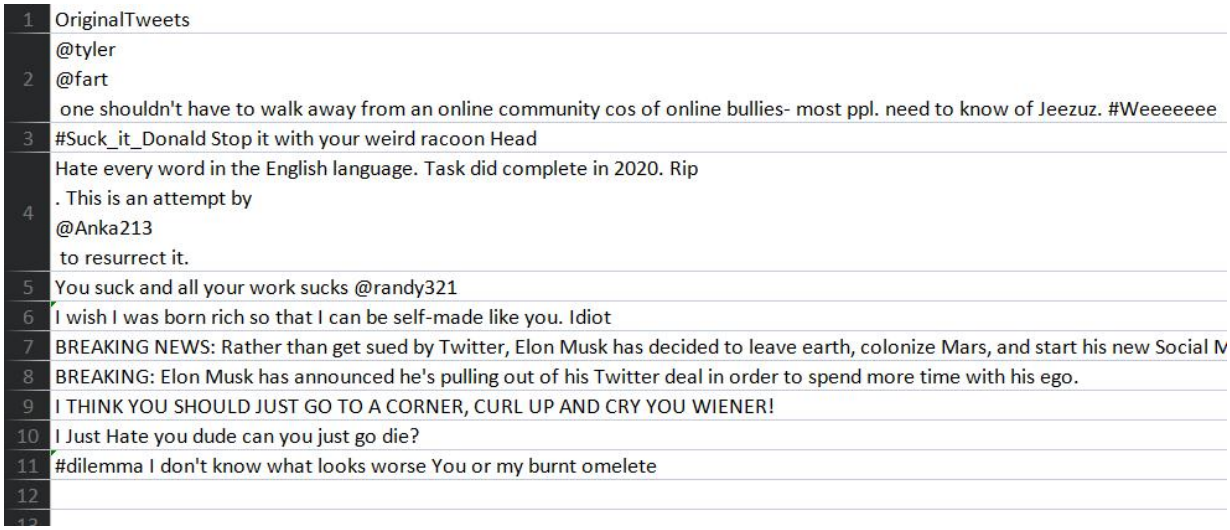
Xpath extracted for the tweet posted by Elon Musk is "//div[@data-testid ='tweetText']" and used for scraping using **visibility_of_all_elements_located** function from Selenium package which is depicted in Figure 5.3

```
tweets = WebDriverWait(driver, 5).until(EC.visibility_of_all_elements_located((
    By.XPATH,"//div[@data-testid='tweetText']")))
```

Figure 5.3 Code Snippet For Scraping Tweets

5.2 TWEETS COLLECTION AND PREPROCESSING

This module depicts the way of collecting real-time tweets after webscraping from the specified Twitter page and saving them as a dataframe. The dataframe is then converted to Comma Separated Values (CSV) file so that it is easier to import tweets into a spreadsheet regardless of the specific software being used. Figure 5.4 depicts the collection of real-time tweets stored in CSV file after webscraping.



	OriginalTweets
2	@tyler @fart one shouldn't have to walk away from an online community cos of online bullies- most ppl. need to know of Jeezuz. #Weeeeeee
3	#Suck_it_Donald Stop it with your weird racoon Head
4	Hate every word in the English language. Task did complete in 2020. Rip . This is an attempt by @Anka213 to resurrect it.
5	You suck and all your work sucks @randy321
6	I wish I was born rich so that I can be self-made like you. Idiot
7	BREAKING NEWS: Rather than get sued by Twitter, Elon Musk has decided to leave earth, colonize Mars, and start his new Social M
8	BREAKING: Elon Musk has announced he's pulling out of his Twitter deal in order to spend more time with his ego.
9	I THINK YOU SHOULD JUST GO TO A CORNER, CURL UP AND CRY YOU WIENER!
10	I Just Hate you dude can you just go die?
11	#dilemma I don't know what looks worse You or my burnt omelete
12	

Figure 5.4 Collection of Real-Time Tweets after Webscraping

Preprocessing tweets is the next step after tweets collection. The following steps are performed to clean the raw tweets in the saved CSV file:

- Removing twitter handles as they are already masked as @user due to privacy concerns which hardly give any information about the nature of the tweet.

- Getting rid of emojis, punctuations, numbers, and even special characters such as ‘$’, ‘&’, ‘%’ since they are not helpful in differentiating different types of tweets.
- Removing hyperlinks, non UTF-8 or ASCII characters such as '\x9a\x91\x97\x9a\x97’.
- Hashtags, newline characters(‘\n’) and multiple spaces are also removed to reduce the length of the tweet.
- Finally, the tweets are passed to a tokenizer to remove long tokenized sentences since most of the long tokenized sentences would be in languages other than English.

5.3 FEATURE EXTRACTION

The TF-IDF algorithm used for machine learning models is imported from scikit-learn package. The function **TfidfTransformer** is used with the parameter **use_idf = True** passed to enable IDF operation for the tweets which is then transformed into TF-IDF representation using the function **transform()**.

Tokenizer used for shallow neural networks is imported from transfomers package which is an open source library provided by HuggingFace. **BertTokenizerFast** is the name of the Tokenizer class being imported which has the function **encode_plus** that returns a dictionary containing the input IDs and attention masks for the tokenized tweet. The maximum length of the tokens is 128 which is passed as the second parameter for the function.

5.4 REAL-TIME TWEETS CLASSIFICATION

The preprocessed tweets are then passed as an input to various text classification models in this module which classifies the tweets as either Bullying or Non Bullying. All the real-time tweets scraped from the given Twitter URL are classified and the count of those classifications are stored in a dataframe.

The BERT model used has a 4 layer architecture with the first layer for getting the input IDs of the tweets, second layer for getting the attention masks of the tweets, third layer contains the BERT Main Layer and the final layer contains the output neurons for the two classes Bullying and Non Bullying. The output is generated from the BERT model after classifying real-time tweets along with their Softmax scores.

5.5 REPORT GENERATION

The classified real-time tweets are counted and a report is generated and saved as a CSV file which is then displayed in the web application created using Angular JS. The report generated gives information about the Profile Picture of the Twitter Handle, Name of the Twitter Handle, Number of Tweets posted, Number of Tweets Classified as Bullying and Percentage of Bullying Tweets.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 PERFORMANCE METRICS

Comparative study between various text classification models is performed for Corona NLP Twitter dataset based on the performance metrics such as Accuracy, Precision, Recall and F1 score and the results have been displayed in Table 6.1 for Bullying class and in Table 6.2 for Non Bullying class respectively.

Table 6.1 Performance Metrics Result for Bullying Class

Bullying Class				
Model Name	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naive Bayes	0.74	0.66	0.83	0.73
SVM	0.84	0.81	0.82	0.81
LSTM	0.77	0.73	0.75	0.74
CNN	0.7	0.64	0.68	0.66
GRU	0.7	0.64	0.68	0.66
Logistic Regression	0.83	0.8	0.81	0.8
Bi-LSTM	0.77	0.73	0.73	0.73
Bi-GRU	0.78	0.75	0.73	0.74
BERT	**0.9**	**0.87**	**0.9**	**0.88**

Table 6.2 Performance Metrics Result for Non Bullying Class

Non Bullying Class				
Model Name	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naive Bayes	0.74	0.84	0.67	0.75
SVM	0.84	0.86	0.85	0.85
LSTM	0.77	0.81	0.8	0.8
CNN	0.7	0.74	0.71	0.73
GRU	0.7	0.74	0.71	0.73
Logistic Regression	0.83	0.85	0.85	0.85
Bi-LSTM	0.77	0.8	0.8	0.8
Bi-GRU	0.78	0.8	0.82	0.81
BERT	**0.9**	**0.92**	**0.9**	**0.91**

From the Table 6.1 and 6.2, it is inferred that BERT model has high performance metrics than other text classification models.

6.2 VISUALIZATION

The performance metrics results from the comparative study are plotted in double bar graphs. Figure 6.1 illustrates Accuracy for classes Bullying and Non Bullying across different machine learning and shallow neural network models for Corona NLP Twitter dataset.

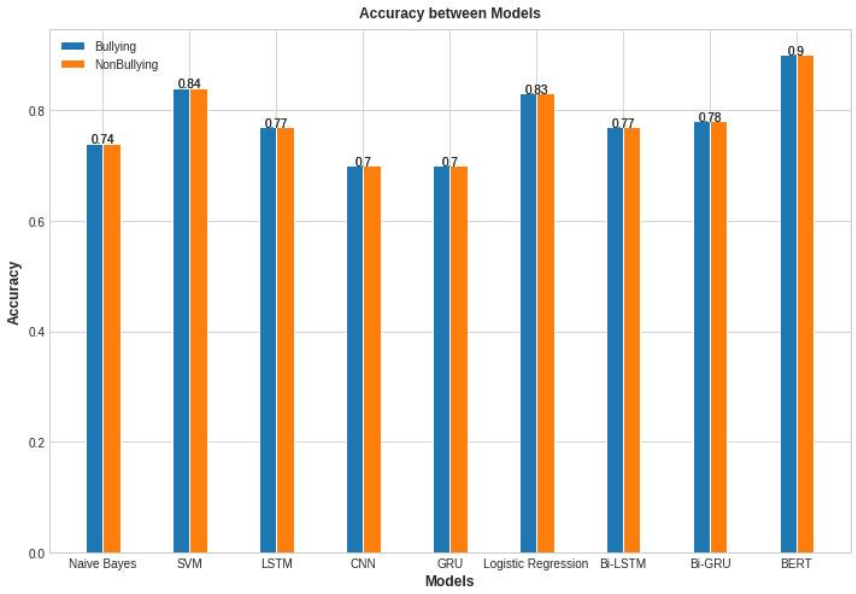


Figure 6.1 Accuracy Metric Between Text Classification Models

Figure 6.2 illustrates Precision across different text classification models for Corona NLP Twitter dataset.

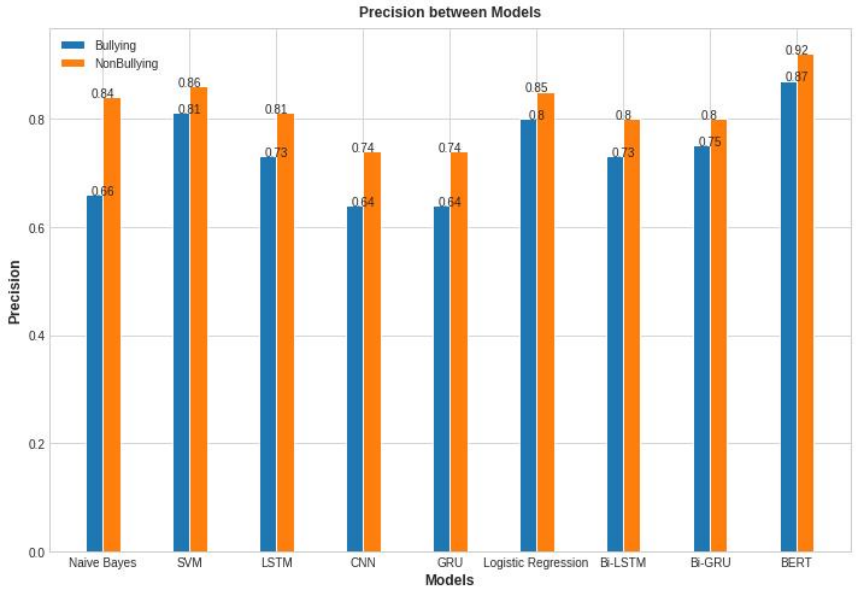


Figure 6.2 Precision Metric Between Text Classification Models

Figure 6.3 illustrates Recall across different text classification models for Corona NLP Twitter dataset.

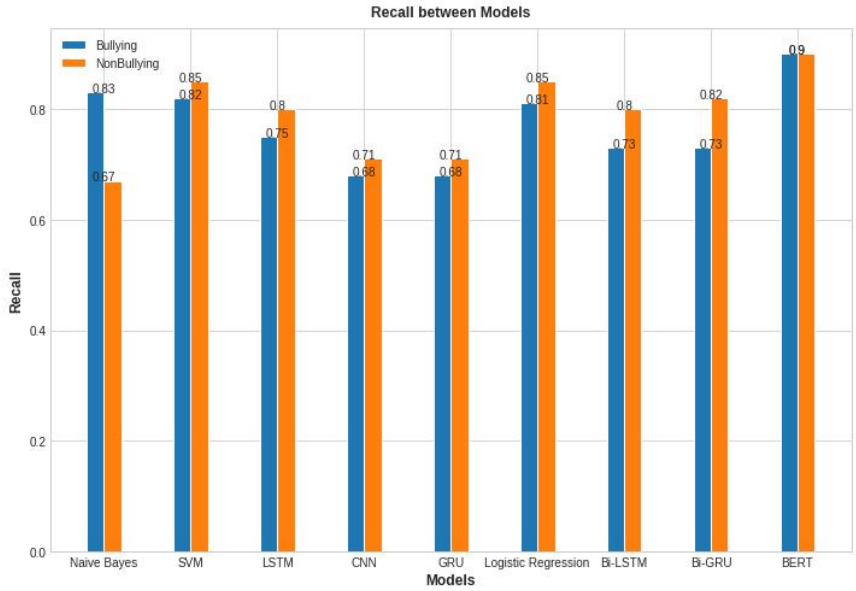


Figure 6.3 Recall Metric Between Text Classification Models

Figure 6.4 illustrates F1 Score across different text classification models for Corona NLP Twitter dataset.

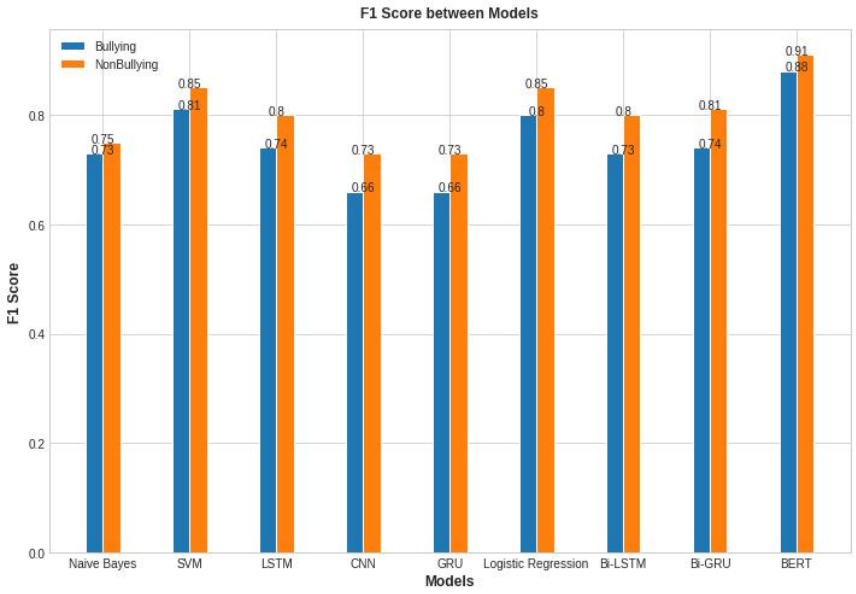


Figure 6.4 F1 Score Metric Between Text Classification Models

6.3 ANALYZING BEST CLASSIFICATION MODEL

After comparing performance metrics across different models, it is notable that BERT model performs text classification with the highest accuracy. Based on the performance comparison tables, it is observed that shallow neural networks highly outperform machine learning models. Bidirectional neural networks models have been observed to perform better for real-time tweets.

Figure 6.5 depicts the BERT model summary for real-time tweets classification and Figure 6.6 shows the output generated from the BERT model after real-time classification of tweets for the classes Bullying and Non Bullying along with their Softmax scores.

```
Model: "model"
_________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
=================================================================================================
 input_1 (InputLayer)           [(None, 128)]        0           []

 input_2 (InputLayer)           [(None, 128)]        0           []

 bert (TFBertMainLayer)         TFBaseModelOutputWi  109482240   ['input_1[0][0]',
                                thPoolingAndCrossAt               'input_2[0][0]']
                                tentions(last_hidde
                                n_state=(None, 128,
                                 768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 dense (Dense)                  (None, 2)            1538        ['bert[0][1]']

=================================================================================================
Total params: 109,483,778
Trainable params: 109,483,778
Non-trainable params: 0
_________________________________________________________________________________________________
```

Figure 6.5 BERT Model Summary

```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 [==============================] - 16s 16s/step
SCORE :

99.99547004699707
99.99347925186157
99.99927282333374
99.99775886535645
99.97593760490417
99.97453093528748
51.33717656135559
80.56979775428772
99.99738931655884


Not Bullying
Bullying
Not Bullying
Bullying
Not Bullying
Not Bullying
Not Bullying
Bullying
Not Bullying
PS C:\Users\Asus\Documents\miniproject>
```

Figure 6.6 Classification of Real-Time Tweets by BERT Model

Figure 6.7 depicts the CSV file of the report generated after real-time classification of tweets by BERT model.

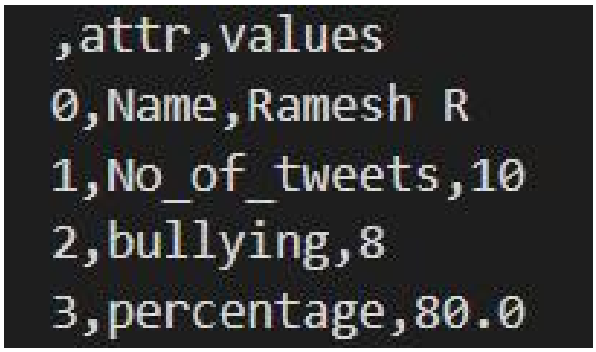


Figure 6.7 CSV File of the Generated Report

Figure 6.8 shows the generated report in the web application.

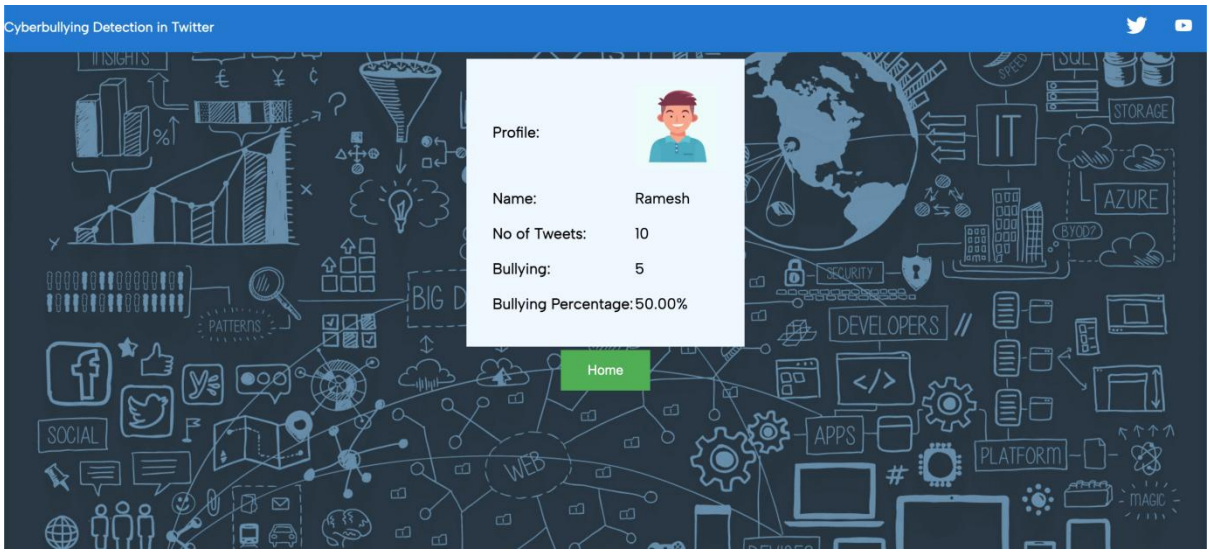


Figure 6.8 Report Generated in the Web Application

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

This research work identifies the best text classification algorithm from three machine learning models and six shallow neural network models in detecting Cyberbullying tweets using TF-IDF and Tokenizer feature extraction algorithms for Corona NLP Twitter dataset. According to the comparative study's findings, it is found evident that Bidirectional Encoder Representations from Transformers (BERT) consistently outperforms other text classification models in terms of Accuracy, Precision, Recall, and F1 scores, with scores as high as 90%, 92%, 90%, and 91%, respectively. Therefore, BERT is used for real-time classification of tweets.

7.2 FUTURE WORK

Consideration of all sentiment analysis features, including sarcastic, syntactic, semantic, and social aspects, can improve findings in the process of detecting cyberbullying. Future research work may combine additional models with the BERT model to develop a cutting-edge model for the specific NLP tasks involved in identifying cyberbullying. Web Scraping can also be extended to other social media platforms like Facebook and Instagram to scale the proposed Web Application across all social media platforms.

REFERENCES

1. Chahat Raj, Ayush Agarwal, Gnana Bharathy, Bhuva Narayan and Mukesh Prasad (2021) 'Hybrid Models Based on Machine Learning and Natural Language Processing Techniques, Electronics, 10(22), p. 2810.

2. John Hani, Mohamed Nashaat, Mostafa Ahmed, Zeyad Emad, Eslam Amer and Ammar Mohammed (2019) 'Supervised Machine Learning Approach for Detecting and Preventing Cyberbullying', International Journal of Advanced Computer Science and Applications Vol. 10 Issue 5.

3. Andrea Perera and Pumudu Fernando (2021) 'Accurate Cyberbullying Detection and Prevention on Social Media', Procedia Computer Science, Vol. 181, pp. 605-611.

4. G. A. León-Paredes (2019) 'Presumptive Detection of Cyberbullying on Twitter through Natural Language Processing and Machine Learning in the Spanish Language', CHILECON pp. 1-7.

5. J. Yadav, D. Kumar and D. Chauhan (2020) 'Cyberbullying Detection using Pre-Trained BERT Model', ICESC, pp. 1096-1100.

6. Amirita Dewani., Memon, M.A. & Bhatti S. (2021) 'Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data', Journal of Big Data Vol. 8, pp. 1-20.

7. R. R. Dalvi, S. Baliram Chavan and A. Halbe (2020) 'Detecting A Twitter Cyberbullying Using Machine Learning', ICICCS, pp. 297-301.

8. Elsafoury, Fatma and Katsigiannis, Stamos and Wilson, Steven and Ramzan, Naeem (2021) 'Does BERT pay attention to cyberbullying?', 44th International ACM SIGIR Conference on Research and Development in Information Retrieval Online, pp. 1900-1904.

9. Aditya Desai, Shashank Kalaskar, Omkar Kumbhar and Rashmi Dhumal. (2021) 'Cyber Bullying Detection on Social Media using Machine Learning', ITM Web of Conferences, Vol. 40, p. 03038.

10. Jalal Omer Atoum, (2020) ‘Cyberbullying Detection Through Sentiment Analysis’, International Conference on Computational Science and Computational Intelligence (CSCI), pp. 292-297.

11. https://huggingface.co/docs/transformers/main/en/index

12. Dataset from Kaggle : https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification

13. https://huggingface.co/docs/transformers/model_doc/bert

14. https://medium.com/mlearning-ai/twitter-sentiment-analysis-with-deep-learning-using-bert-and-hugging-face-830005bcdbbf

15. https://www.lambdatest.com/blog/getting-started-with-selenium-python/

16. https://avishkabalasuriya980330.medium.com/serve-angular-application-in-python-flask-server-bd37c8a0b431