# Deep Learning Approaches for Bengali Cyberbullying Detection on Social Media: A Comparative Study of BiLSTM, BiGRU and BERT Models

by

Kaji Mehedi Hasan Fahim
21341038
Nasita Nyla
23141053
Priti Saha
20101475
Mst. Shamima Akter
19101473
Musfiqur Rahman Shourav
19201116

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfilment of the requirements for the degree of
B.Sc. in Computer Science

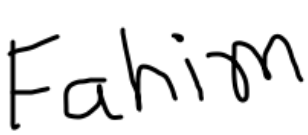
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 2023

# Declaration

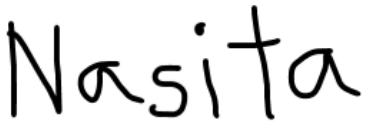
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**



Kaji Mehedi Hasan Fahim
21341038



Nasita Nyla
23141053

Priti Saha
20101475

Mst. Shamima Akter
19101473

Musfiqur Rahman Shourav
19201116

# Approval

The thesis/project titled "Deep Learning Approaches for Bengali Cyberbullying Detection on Social Media: A Comparative Study of BiLSTM, BiGRU and BERT Models."
submitted by:

1. Kaji Mehedi Hasan Fahim(21341038)
2. Nasita Nyla(23141053)
3. Priti Saha(20101475)
4. Mst. Shamima Akter(19101473)
5. Musfiqur Rahman Shourav(19201116)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 17, 2023.

**Examining Committee:**

Supervisor:
(Member)

Dr Farig Yousuf Sadeque
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Mr MD Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Chair)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Ethics Statement

In the goal of academic success, this thesis shows that researchers are always committed to doing research in a way that is fair and honest. With the utmost commitment to the values of honesty and respect, I promise to treat human subjects with care, handle data in an honest way, and stick to the rules of academic ethics. I have all the permissions and consents I need for people to take part, protecting their rights and privacy. Personal and private information has been carefully anonymized and kept safe in the field of data management. I have accepted my ethical responsibilities as a student, making sure that this study is done in a way that is open, honest, and meets the highest ethical standards.

# Abstract

As technology becomes more accessible, it is now much easier than ever to abuse someone by misusing it. Usually, people use slang or absurd language with the goal of bullying, harassing, and harming someone by using social media. Moreover, these types of cyberbullying activities are more widespread among teenagers and young people despite knowing the fact that these may break someone down emotionally and may lead them towards suicidal activities. Hence, our goal is to detect cyberbullying happening on social media in the Bengali language with the help of state-of-the-art deep learning and Natural Language Processing (NLP) techniques. We have examined with 3 different algorithms such as Bi-LSTM, Bi-GRU and BERT for both multiclass and binary classification. For both binary and multiclass classifications, BERT outperformed the other two models in terms of performance with the f1 score of 0.89 for binary and 0.85 for multiclass classification. Our proposed state-of-the-art transformer model BERT will detect whether a message or comment is sent to harass someone or not and could help to take immediate action against them. Therefore, our research might have a positive impact on changing the social media environment by detecting hate speeches and bullying messages.

**Keywords:** Natural Language Processing; Machine Learning; GloVe Embedding; FastText Embedding; Bi-LSTM; Bi-GRU; BERT;

# Dedication

This thesis is dedicated to those whose unwavering support and encouragement have guided our academic endeavours. I extend my deepest gratitude to our family for their boundless love, sacrifice, and belief in my aspirations. To our supervisor Dr. Farig Sadeque sir, whose wisdom and guidance have shaped our intellectual development, we owe a debt of gratitude that words cannot fully convey. And to all those who believe in the power of knowledge to transform lives, this work stands as a tribute to your enduring faith.

# Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Farig Yousuf Sadeque sir for his kind support and advice in our work. He helped us whenever we needed help.
Thirdly, to the the whole judging panel and oviously our co-supervisor Mr MD Tanzim Reza sir.
And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

# Table of Contents

# ListofFigures

# ListofTables

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$GRU$ Gated Recurrent Units

$LSTM$ Long Short-term Memory

$RNN$ Recurrent Neural Network

BERT Bidirectional Encoder Representations from Transformers

GloVE Global Vector

NLP Natural Language Processing

# Chapter 1

# Introduction

## 1.1 Context and Importance of Cyberbullying Detection on Social Media

In this new and advanced era, social media is an online platform that enables users to interact and communicate with their online friends. It also enables users to share their daily updates, content, photos, videos, documents, collaborations, ideas, information, and knowledge via the Internet. Users can interact and communicate with their online friends by using social media.

Recently, the most considerable attention has been paid to Bangla NLP, which makes it possible for machines to read Bangla language. Natural language processing is an area of linguistics and computer science associated with the interaction between computers and human language. Its primary objective is to develop computers capable of processing and analyzing huge volumes of natural language data. The objective is to create a machine that can "understand" the content of a document, including how language is used in various contexts. Natural language processing combines linguistics and computer science to determine how language functions and to create models that can comprehend, decode, and extract relevant details from text and speech. NLP combines research in linguistics, computer science, and data science to help computers interpret language more similar to humans.

The phases of preprocessing data in NLP include ambiguity, variability, sentence segmentation challenges, syntactic analysis, semantic analysis, lexicon, stemming, tokenization challenges, and lemmatization embedding. Ambiguity in sentiment analysis is when a word or a statement might have more than one meaning. That is, there is more than one way to understand a sentence. Ambiguity is a significant difficulty in natural language interpretation. Almost every step of natural language processing involves ambiguity. The stages of NLP are lexical analysis, syntactic analysis, semantic analysis, discourse analysis, and pragmatic analysis. A semantic network or frame network is a knowledge base that demonstrates the semantic relationships between ideas in a network. Semantic analysis is a way of getting at the exact meaning of a statement. A lexicon is usually quite systematic in how it is put together. It stores the definitions and uses of each word and the connections between words and their meanings. A lexeme is the smallest item in the lexicon and it is the stem of a word. The lexicon is a database that provides information about the

words in a language and the lexical categories they belong to. For instance, People normally use the term "pig" as a noun, but it can also be a verb or an adjective, as in the phrase "pig iron." In reality, a lexical entry will have more information about how a word operates, like what form a verb takes and whether or not it is transitive, intransitive, or ditransitive. The third step is syntactic analysis. The name of the method makes it clear that it is used to look at syntax, which is also termed syntax or parsing analysis. In this step, the exact meanings or "dictionary meanings" of the text are extracted. Syntax analysis compares a text to well-known grammar rules to figure out what it means. The fifth and final phase of NLP is pragmatic analysis. The pragmatic analysis looks at the communicative and social content as a whole and how it impacts how it is interpreted. It refers to the process of stripping away the meaning of the words used in a given scenario. It translates the text that is supplied to it based on what it has learned so far. Sentence segmentation is the first stage in dividing a text into its individual sentences. To achieve this, break the piece into sections and strip remove any punctuation, including commas and full stops.

Additionally, in order to tokenize a sentence, each word must be placed in its own sentence, and each sentence must be explained to the computer separately. Therefore, separating a sentence into its individual words and storing them is known as tokenization, and each word is known as a token. As tokens are the fundamental building blocks of Natural Language, processing raw text at the token level is the most common method. In addition, tokens can be subwords. For instance, the most sophisticated Deep Learning architectures in Natural Language Processing (NLP) use Transformer-based models to parse raw text at the token level. Popular deep learning architectures for NLP, such as RNN, GRU, LSTM and BERT, analyze unprocessed text at the token level. As a result, it is the initial step in text data modelling. Through tokenization, tokens are extracted from the corpus. The next step is to create a vocabulary using the terms listed here. The vocabulary is the collection of unique tokens within the corpus. Taking this into consideration is another way to expand one's vocabulary. Tokenization, however, is the process of separating a text into its individual words.

The next stage is to eliminate stop words or words that add nothing to the meaning of the sentence. These terms include was, in, and is, and they can be eliminated. Similarly, steaming is the process of locating a word's stem. By adding suffixes to word stems, new words are created. Lemmatization is the process of finding the root stem of a word. Root Stem gives the new base form of a term that already exists in the dictionary and is derived from. The root words of many words can also be determined by examining the tense, mood, gender, etc. In our dataset, we tokenized the phrases and used count-based model named GloVe (global vector) and FastText word embedding.

However, With the use of a deep learning algorithm, our primary objective in this scenario is to identify instances of cyberbullying or cyber harassment that take place on the social media sites that we operate. the data is collected from the popular social media platform Facebook. later, the comments are labelled and fed to the deep learning algorithms BiLSTM, BiGRU and BERT from which we can extract the predicted labelling of the comment. Therefore, if we are able to detect this sort

of harassment or crime in online platforms and identify the individuals while taking necessary steps, scrolling social media will be more mesmerizing and less toxic.

## 1.2 Problem Statement

Cyberbullying on social media has become an intractable problem in recent times. The main purpose of social media is to link people with each other. Besides, nowadays social media has become a platform where people use it to keep themselves updated, get services, for business purposes, learning, entertainment purposes, and so on a daily basis. Millions of users including students and children use Facebook, Twitter, and social media platforms on a daily basis. Social media have become a part and parcel of our daily life which means social media content has an effect on people's mental health, and behaviour. People are influenced by social media every day. So, it is really important to keep a healthy environment on social media.

At present, countless number post is noticeable that contains extremism, misinformation, harassment, violence, and content that are not age-appropriate. These types of statuses may create anxiety and depression among teenagers. And a lot of people are being bullied and harassed every day on online platforms every day. Besides, There are contents that may seem neutral but contain a different meaning. And in reality, many people share those posts without knowing the context behind them which influence other people in a negative way.

It has become very hard to for developers detect cyberbullying as a countless number of content is out there on social media. This is why developers are working on methods to detect posts that contain offensive and abusive stuff. Some of the effective approaches are BERT-based fine-tuning, robust modular neural architecture using RNN, evaluating baseline and dataset, combining machine learning and neural language processing, etc. By perfecting these methods we can easily detect cyberbullying and it will help us to ensure that nobody becomes a victim of cyberbullying.

## 1.3 Research Objective

The detrimental effects that some messages can have on an individual or on society have been one of the major issues with social media usage over the past few years. It's crucial to quickly identify cyberbullying when it occurs. Abuse can take many forms, including hate speech, offensive language, and cyberbullying. Several research have been done to identify and track down these kinds of language. For the purpose of identifying hate speech online, numerous techniques have been developed.

The main goal of our research is to provide an environment where no one will attack anyone by using offensive language. To address this problem and create a better online environment, more research is urgently needed.

In order to address the issue of hate speech identification, the focus has recently switched toward machine learning and deep learning technologies. The lack of a large and diverse dataset is one of the main barriers to using cutting-edge deep-learning models to detect hate speech. For the purpose of identifying hate speech, we employ several deep learning models. By creating quick and effective algorithms as well as data-driven models for processing data in real-time, machine learning may produce reliable findings and analysis.

Due to the growing amount of hate speech that is being distributed on various social media platforms like Facebook, Twitter, YouTube, etc., our objective is to create a system where any toxic words and phrases will be automatically identified by our system and will be dealt with accordingly. To date, several research works have been done on this topic, and the majority of their proposed models have done a great job when detecting hate speech as mentioned by their accuracy. However, We will develop a system that takes into account the Bangla language as well because research on this topic in the Bangla language is still on the radar. Since cyberbullying on social media is becoming a threat nowadays, we have selected cyberbullying detection as our research topic.

# Chapter 2

# Related Work

In their paper, [12] Corazza et al. (2020) focused on using multiple languages such as English, German, and French for detecting hate speech. Firstly, this paper used a robust modular neural architecture using RNN where they fixed the number of neurons in the hidden layer to 100. Moreover, the authors used 3 different recurrent layers such as LSTM, GRU, and BiLSTM, and different NLP techniques such as word embedding, emoji embedding, emotion lexica, BERT( Bidirectional Encoder Representations from Transformer), etc to solve the problem. For neuron activation, the ReLu activation functions are used here for the hidden layer and the sigmoid activation function for the output neuron. Secondly, for the English, French and German datasets, the authors used 16000, 4000, and 5009 tweets respectively where tweets contain offensive languages and nonoffensive languages. The ratio of offensive and nonoffensive speeches in datasets of all three languages is around 32% go to the categories of offensive and 68% go to others(i.e., not offensive). Moreover, the authors splited the dataset into 60% for training, 20% for validation, and 20% for testing. Finally, for English datasets, LSTM and Fasttext embeddings performed relatively better results (0.823 F1). And for Italian datasets LSTM outperformed all the other algorithms in terms of performance results(0.805 F1). Lastly, GRU surpassed other algorithms for German datasets and the result was 0.758 F1.

Moreover, in another paper that we have reviewed, the researchers of this paper, [19] Plaza-del-Arco(2020) et al. applied the Spanish language to test their experiments on the identification of hate speech using three deep learning architectures such as LSTM, Bi-LSTM, and CNN also two machine learning techniques such as Support Vector Machine and Logistic Regression. The authors also applied Pre-trained BERT, XLM (Multilingual), and BETO(monolingual) models in order to do their experiment. The scientists gathered two datasets, the first of which was collected from HaterNet and the second from HatEval, both of the datasets contained hateful tweets. In the first HaterNet datasets in total, where the datasets contain hate speech towards negro and feminists, 2 million tweets were present from which the publishers picked 8710 tweets. Finally, only 6000 tweets were labeled where 1567 were hateful and 4433 were categorized as not hateful. Moreover, In the HatEval dataset, where the tweets were mainly targeted toward women and immigrants, in total 6600 tweets were present where 3019 were tagged as hateful and 3581 were not hateful. Finally, when it comes to performance, SVM(TF-IDF) surpassed all other ML and DL learning algorithms for HaterNet datasets, scoring 71.13 F1. Moreover, for the HatEval datasets, all the used deep learning models performed neck to neck

however Bi-LSTM performed slightly better scoring 75.49 F1 outperforming other traditional ML algorithms. But for pre-trained Language Models, BETO did relatively better in both datasets, for HaterNet dataset BETO scored 77.23 F1 followed by XLM and BERT, and for HatEval dataset, BETO scored 77.62 F1.

This publication [15] examined methods for identifying offensive language and hate speech using three-step classification. These three steps were: detection, classification, and prevention. Offensive and no offensive were the two choices. Any comments that contained offensive language were sorted according to whether or not a specific user was labeled. After that, the author classified the remark or speech as "others" if it wasn't directed at a specific person or group. Additionally, this paper presented some characteristics of the aggressive language detection literature, including linguistic characteristics that matched the word using a dictionary and attempted to classify them if there was any hate speech present. Out of the many models considered in this paper, the naive baseline model, which lumped all samples into a single category, was introduced first. Then, logistic regression was developed to foresee the probable outcome, and a classifier was developed to determine whether or not a given piece of discourse is offensive. A number of different methods for identifying offensive words were tested and compared, including the leaned BILSTM classifier, the Fast BiL-STM classifier, the AUX-FAST BiLSTM classifier, and hyper-parameter adjusting; however, for Danish data, logistic regression yielded the best results. Also, AUX-Fast-BiLSTM was superior to other methods when it came to classifying data. Lastly, for the purpose of locating a target Superior results could be achieved with Learned-BiLSTM. To understand where the classifier that was used for categorization, was lacking TF-IDF scores were calculated for the n-gram range. The researchers noticed some restrictions on the availability of data sources. The dataset of a widely-used medium must be chosen depending on the target language.

Furthermore, Romim et al.(2021) in this article [20] was focused on the Bengali dataset and tests various deep learning models on it. The dataset was made available primarily for future research accessibility by the author. For this study, 35,000 user comments from YouTube in seven different categories—crime, sports, politics, religion, celebrity & meme, TikTok, and entertainment—were gathered. The comments were extracted using the Face Pager program. The received comments were labeled and categorized for later use. The data sets were then annotated using a number of criteria for selecting hate speech, including dehumanizing and ideas that dehumanize an individual or group. Only ten thousand of the thirty thousand comments were hate speech, according to the data set's annotation. After analyzing the data, the author discovered that hate speech scored highest in the crime category and lowest in the celebrity category. The gathered data were prepared for testing. Word2Vec, FastText, and BengFast were utilized for word embedding, and Support Vector Machine (SVM), Long Short Term Memory, and Bi-directional Long Short Term Memory were utilized for deep learning in various combinations with the word embedding system. After analysis, we discovered that SVM has a higher accuracy rate (87.5%) and an F-1 score of 0.911, but errors were still found because the system occasionally had trouble differentiating between aggressive and dehumanizing words. The constraint of this paper was the lack of sufficient data. Even when the author managed to find data but those lacked labels even when the research was done on training 250 million Bengali texts but still lack of data set was visible, which

the author tried to overcome.

However, here the writer of this publication, [13] Mozafari et al.(2019) mainly focused on a transfer learning approach that uses a pre-trained language model BERT to detect hateful speech on social media. Due to a shortage of annotated data, it has become difficult for the researcher to identify situations that promote hatred on social media. BERT uses the labeled data set of English Wikipedia and BookCorpus containing 2500M and 800M embedded tokens to evaluate the background data. BERT contains an encoder with 12 layers, 12 self-attention heads, and 110 million parameters to detect hateful speech. After extracting the data from BERT some new fine-tuning strategies are applied to make the detection more accurate. The fine-tuning strategies use BERT base fine-tuning, CLS, Bi-LSTM, and CNN layer to detect the contextual meaning of a word. To identify hateful speech, the writer used 84.4 million tweets that contain hate language. Then different fine-tuning strategies were applied and we saw a significant amount of improvement after using BERT and fine-tuning strategies. For example, In precision,Recall,F1-Score BERT scores 91% , 91%, 91% whereas in precision,Recall,F1-Score BERT+CNN scores 92%,92%,92% respectively. This indicates BERT with fine-tuning strategies gives us more accuracy in detecting the context behind social media content.

Besides, in this study, [10] Sohn et al.(2019) worked on different BERT models to detect cyberbullying in different languages. Anonymity and mobility provided by social media services have increased the amount of hate and toxic speech on those platforms. However, an automatic recognition algorithm has been proposed by the writers in this study. A pre-trained BERT that learns deep bidirectional representations from a substantial token corpus. Basically what BERT does is it basically masks a word and then tries to predict the context behind a word. The writers proposed a multi-channel model with three versions of BERT for languages like English, Chinese, and multilingual to detect hate speech. Now, the mythology was they would translate a language into English with Google translation API and apply the BERT base fine tuning to detect hate speech. They used various versions of datasets like English, Spanish, Chinese, etc to detect cyberbullying. CNN, RNN, LSTM model 23, and GRU had been used to improve the effectiveness of this model. After running the model, they found out that In English and other languages, Multi-channel BERT fine tuning turned out to be the highest F1 score holder and the accuracy was higher than other models as well. Therefore, it can be said that Transfer learning is a very effective approach to detecting hate speech.

In order to detect hate speech in Arabic, [11] Albadi et al. (2022) described a series of experiments utilizing several neural networks including RNN and CNN. The issue of automatically detecting Arabic hate speech was only addressed by one study. CNN, GRU, CNN+GRU, and BERT were used for Arabic hate speech detection tasks. GHSD was used for training models and RHSD dataset was used for testing our models. For training, we used 75% of the data, and the remaining 25% of the data was used to test the improved models. The Keras library with TensorFlow as a backend was used to create all neural network models and the Hugging Face Pytorch library was used to implement BERT. Since n-gram-based models excelled at hate detection tests, we employed SVM and LR classifiers instead. The CNN model outperformed the other neural network models in terms of results. In comparison

to other models, it also obtained the highest hatred class recall. CNN's capacity to extract local and position-invariant characteristics such as nearby words and word orders was responsible for the performance gain. Moreover, Compared to the baselines and the others examined models, BERT did not show any improvement.

However, Jahan et al.(2021), in this report, [18] provided a critical assessment of the development of the automatic identification of hate speech in the text during the previous few years. Preferred Reporting Items for Systematic Reviews and Meta-Analyses(PRISMA) inclusion and exclusion criteria were found to be met by a total of 463 articles. The results showed that the SVM technique and several TF-IDF feature types were initially the most popular. We discovered 69 datasets in 21 distinct languages. English was one of the other languages, and it alone represented 26 datasets. 45% of all datasets were acquired through Twitter, making it the most common medium for gathering data on hate speech. The most widely used deep-learning model, covering 33% of all detected entries was BERT. Then, CNN covered 12% and LSTM covered 20%. The plot's examples of hybrid models included BERT+CNN (2%), LSTM+CNN (9%), LSTM+GURU (1%), and BERT+LSTM (2%). Additionally, CNN+LSTM and CNN+GRU both outperformed LSTM and CNN used alone. According to Badjatiya et al. [2021], LSTM architecture outperformed CNN in terms of performance. Different word-embedding models FastText, Word2Vec, and GloVe performed similarly when compared. But ELMO outperformed the competition just a little bit. BERT was a contextual relations-based model that was introduced in 2018, and multiple publications claimed it outperformed ELMO, CNN, and RNN models.

According to the research papers [14] the scientist Nikiforos et al.( 2020) mentioned that cyberbullying which evolved through dialect conversation was hampering the ethnicity and behavior of the young generation. Firstly, the writer mentioned that many organizations were taking steps to detect bullying using natural language processing (NLP) as well as machine learning (ML) which may automatically determine the negative traits and overlapping nature of behavioral impulsive reactions, as recently discovered [8] . According to scientists [2] in both cases of virtual learning community AI and ML, techniques were used through rapidMinor studio by online learning platform and head of communications were the successful way. Both the communities (VLC-1 and VLC-2) created 500 and 83 segments respectively through anonymization and fully detected bullying. Hence, after presenting unigrams, tokenization and lowercase letters the author mentioned some algorithms which worked as magic like naive Bayes, naive Bayes kernel whose accuracy was 9400, ID3, decision tree, gradient boosted trees, deep learning 1 and 2, rule induction by numerical methods.. There were some challenges like a language with complex morphology and different wrong forms of words that were mentioned in the papers. Therefore, the result of the whole process to detect and prevent bullying was also satisfactory (min 86.20%) as there was no stability of bullying. Moreover, according to papers in the first virtual learning community (VLC-1) and second virtual learning community (VLC-2) teachers' active participation rate was around 14% and 43% respectively, which was moderate [7]. Furthermore, according to their experiment in papers applying tenfold cross-validation, natural language processing, and deep learning nowadays plays a crucial role in using artificial neural networking which helps to minimize harmful behavior in today's society [3]. So, in conclusion, the au-

thor of the papers also focuses on the linguistics analysis using the ( ML) and (DL) algorithms, using a Greek dataset, and applying a Computer Supported Collaborative Learning (CSCL) environment works better to detect bullying from society [21]. By elaborating the author's knowledge, integrating all the promising skeletons could solve the behaviour modification of virtual communities.

# Chapter 3

# Description of the Data

## 3.1 Primary Dataset

In this dataset [16], all the comments after scraping from Facebook comments, get labeled into 5 categories as Non-bully, Sexual, Threat, Troll, and Religious. First of all, a Threat is an aggressive activity by an individual or any organization trying to achieve access to a digital network for stealing information and corrupting data. It is a form of hate speech that could be harmful to an individual's mental health. Moreover, Trolling someone is also another kind of harassment that could be done using social media. Nowadays, trolling is a widespread phenomenon on the internet, and it can have a harmful impact on young people's physical, mental, and emotional well-being. A troll is a person who posts offensive or troublesome comments online with the intent of annoying individuals. However, Trolling is primarily done by people who want attention, feel insecure and want to achieve their own goals, or just like to hurt other people's feelings. When someone begins speaking harshly, it can sometimes be assumed that the unpleasant remarks and arguments are the work of trolls. Hate speech or trolling, for instance, is when someone replies to another person online by making fun of them in an offensive rather than sarcastic way.

| **Genre** | **Comments** |
|---|---|
| Non-Bully | সাহস সবার থাকেনা। নিজের মতামত সবাই তুলে ধরতে পারেনা। |
| Threat | ওরে ভাই কাবিল টারে লাথি মারা উচিৎ |
| Sexual | বাংলাদেশী মিয়া খলিফা |
| Troll | নোসের গ্রহে গেছিলো নাকি!! |
| Religious | সিকি ?? কেউই করে ?? তসলিমার মতো সবসময় চাপাতির তলায় থাকা কয়েকজন না-মুসলিম কে বাদ দিলে ?? |

Table 3.1: Examples of Data Labelled in Each Category

Furthermore, Cyberbullying also includes harassing people based on their religion.

Religious discrimination includes making fun of other people's religions or making fun of someone for having a strong religious belief.

Besides, Sexual bullying is also a harassing behaviour where sexuality is used to bully someone. Pressuring someone to do something proactive, relenting, making sexual jokes, body shaming, and characterizing someone publically in a bad way on social media is also considered sexual bullying. Sexual words that degrade someone's reputation also come under sexual harassment. The commentator actually characterizes someone as like a pornstar by The comment "বাংলাদেশী মিয়া খলিফা" which may make anyone feel uncomfortable and ashamed. This is why the comment has been detected as a sexual bullying comment.

| **Genre** | **No of Comments** | **Percentage** |
|---|---|---|
| Non-Bully | 15,340 | 34.9 % |
| Threat | 1,694 | 3.8 % |
| Sexual | 8,928 | 20.3 % |
| Troll | 10,462 | 23.8 % |
| Religious | 7,577 | 17.2 % |

Table 3.2: Amount of Data in five distinct classes

On the other hand, another type of comment is also present in our dataset categorized as non-bullying where comments do not indicate harassment, rather they are more like appreciating someone, loving or encouraging. The tone of speech can be determined using NLP techniques in order to identify specific attitudes like bullying, hate speech, etc.For example, "ইনশাআল্লাহ অনেক প্রত্যাশা নিয়ে অপেক্ষা করছে" this line is considered as non-bully. A word of hope is revealed by this line and no one is hurt by this sentence. Again, two types of expressions can come from a word, positive and negative. Therefore, one must understand what is meant by a sentence and distinguish between bullying and non-bullying.

there were 44,001 different genres of comments in the Bangla Language, including both bullying and non-bullying texts. The comments are targeted at various politicians, actors, athletes, singers, and social influencers, with 29,950 of them being targeted at women and 14,051 towards men. Hence, the percentage of data targeted toward women is 68.1% and 31.9 % is toward men.

Moreover, all the comments, after scraping from Facebook, get labeled into 5 categories as Non-bully, Sexual, Threat, Troll, and Religious. 34.9% i.e. 15,340 were labelled in the not bully category and 65.1% i.e. 28661 were labelled as bullying comments where 17.2% i.e. 7,577 were religious, 23.8% i.e 10,462 were labelled as a troll, 23.8% i.e 8,928 were labelled as sexual and 3.8% i.e 1694 were labelled as a threat (Table 3.2). The histogram in Figure 3.1 is the representation of the total five categories of data and the number of comments in each category.

Moreover, out of 44,001 comments, 61.25% i.e. 26,951 of comments indicate actors, 21.31% i.e. 9,375 of comments explicitly mention victims who are social influencers,

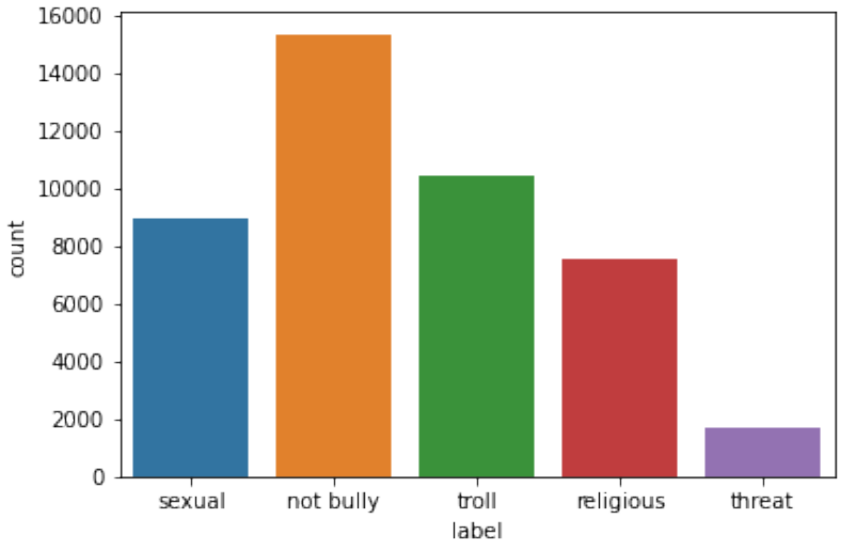



Figure 3.1: Representation of Data

4.68% i.e. 2,061 of comments mention athletes, 5.98% i.e 2,633 of comments indicate politicians and 6.78% i.e 2,981of comments indicate singers.

## 3.2 Secondary Testing Dataset

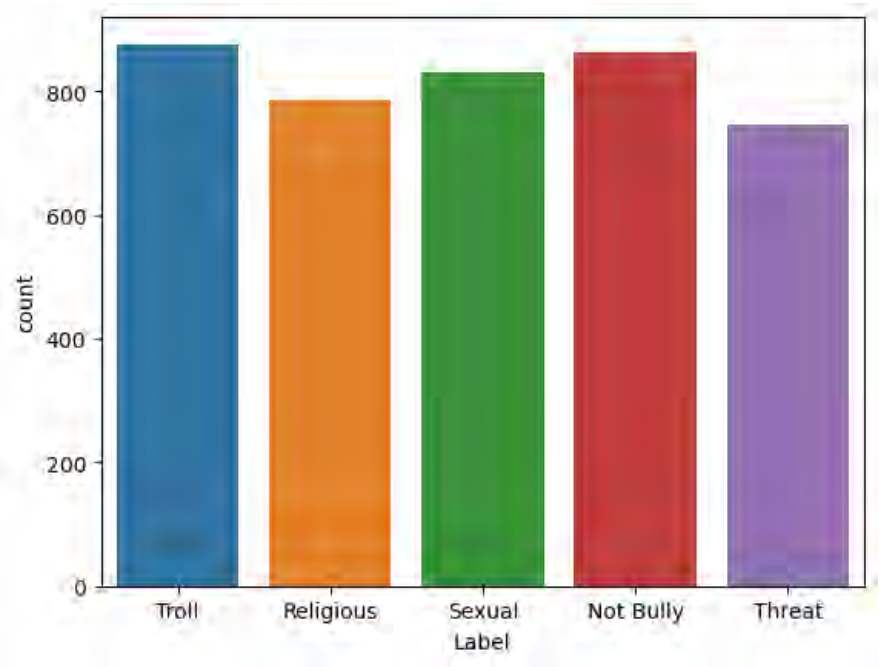



Figure 3.2: Representation of Secondary Testing Dataset

We collected a testing dataset of 4102 comments to use in the thorough evaluation of the model for this thesis. This dataset includes bullying and non-bullying comments in Bengali that are directed at a range of public figures, including politicians,

actors, sports figures, musicians, and social influencers. We have used social media platforms like Facebook, YouTube, Instagram, TikTok and Reddit to collect these comments. The ratio of not-bully and religious comments was high on the pages and channels of athletes, and the ratio of sexual remarks was especially high on the social media pages of actresses. After collecting the comments from the social media platforms, we annotated the collected comments into five categories: non-bully, sexual, threat, troll, and religious. Around 21.03% of the testing dataset, or 863 comments, were classified as not bullying; the remaining 3257 comments, or 79.98% of the dataset, were classified as bullying; these comments included 785 religious comments, 876 troll comments, 831 sexual comments, and 746 threats.

# Chapter 4

# Tokenization and Word Embeddings

In this study, we have examined several models for detecting cyberbullying, including Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Gated Recurrent Unit (BiGRU), and Bidirectional Encoder Representations from Transformers (BERT) for both binary and multiclass classification. We have also used FasText and GloVe embedding methods with the BiLSTM and BiGRU models, and BERT uses contextual embedding.

## 4.1 Tokenization

First, we have tokenized the comments from our dataset. Tokenization is an important part of natural language processing (NLP). It involves breaking a section of text into smaller pieces called tokens. Tokenization is a key step in many NLP tasks, such as classifying text, named object identification, language modelling, and machine translation. In the first step of our work, we tokenized the raw text of our dataset. For example, we have applied tokenization for our Bangla dataset, which helped us break the raw data into small pieces called tokens. The process starts with an expression in its raw form. This text may be a single line, a phrase, a complete paper, or the entire collection. If the text received from our dataset is a multi-sentence document or paragraph, the initial phase is frequently to tokenize each sentence. The objective here is to locate the dividing lines between words and separate the text into separate sentences. As sentence separators, punctuation marks such as commas, exclamation points, and question marks are frequently used. Then comes the tokenization of words. After the text has been parsed into sentences, the next step is word tokenization. The objective of word tokenization is to separate the text into individual words, or tokens. Using spacing (space, tab, or newline) as a divider and splitting the text wherever there is a space is the simplest method to accomplish this. However, this approach might not work identically for all languages, as some possess complicated word arrangements or don't use spaces to split words. In this case, Subword tokenization may be used in specific situations, particularly for languages with complex word patterns or when addressing items that are not part of the vocabulary. Subword tokenization is a technique for dividing phrases into smaller components, such as character n-grams or subword units. The result of tokenization is a list of tokens, with each token representing a reduced portion of

the original text. In addition, we have used this tokenized sequence as input for our word embedding methods, such as FastText and GloVe.

### 4.1.1 FastText Embedding

FastText embeddings are vector representations of words that consider what words mean and how they are put together. This makes them good for NLP tasks like text classification, mood analysis, language modelling, and even more. We have applied fast text word embedding in our dataset. This word embedding model takes the comments from our dataset and turns them into the vector form for each sentence. The fasTtext word embedding model turns our whole dataset into numbers form.

FastText word embeddings perform by describing words as bags of character n-grams and dense vector representations for each of these subwords. The word embedding captures the meaning or semantic information of the word, and similarly, the subword embeddings capture the meaning of individual character n-grams. The model is taught to improve the embeddings in a methodology that takes into account how words are the same in meaning and how they are put together [6].

fastText requires a large corpus of text data, also known as a corpus, in order to extract information. This may include words, paragraphs, or even entire papers, depending on the size of the corpus and training sample. FastText is different in that it utilizes subword information. Instead of representing each word as a distinct vector, FastText divides each word into character n-grams. The text then connects the embeddings of each word's subword units to generate a vector version of the word with a fixed size. The most common method is to calculate the average of all subword embeddings for each character. Once the word representations for all the words in the text have been found, FastText trains a neural network model using the skip-gram or continuous bag-of-words (CBOW) method. The model determines how word embeddings should correspond to the adjacent words in context. During training, subword embeddings are modified based on the context in which they occur. The final result of the training process is an embedding space in which each word is represented by a dense vector with a set of possible dimensions (for example, 100 or 300). Our Bangla dataset was encoded using 300 dimensions ('cc.bn.300.bin') fasttext. This vector shows the semantic relationship between words so that similar words are closely together in the embedding space.

### 4.1.2 GloVE Embedding

GloVe, which initially stood for "Global Vectors for Word Representation," is a popular word embedding model in NLP. It utilizes both global data from the word co-occurrence matrix and local methodologies determined by the context window. By combining global and local information, GloVe aims to generate embeddings that more accurately depict the semantic relationships between words. During the training procedure, a large quantity of text is used to create a word co-occurrence matrix, where each cell indicates the frequency with which two words co-occur in a

particular window region [5]. The model then discovers how to break down this matrix into word embeddings by maximizing a specific goal function while maintaining track of the frequency with which words appear together.

We implemented the glove word embedding model to make the connection between words and checked whether the relation was correct or not. For our task, we have also used pre-trained GloVe word embedding to vectorize our Bangla dataset where we have used 300-dimensional vector space ('bn_glove.39M.300d.txt') glove. GloVe is used to learn word embeddings, which are representations of words in a continuous vector space based on how often they appear together in context.

GloVe begins by creating a word co-occurrence matrix, which measures the frequency with which each word appears next to other words in a fixed-size contextual area. The co-occurrence matrix indicates the frequency with which words appear together in sentences. Next, GloVe uses the word co-occurrence information to figure out a kind of random ratio. The goal is to find connections between words that make sense. This ratio contrasts the probability of two words co-appearing to a standard probability. GloVe introduces a goal function that leverages the chance ratio computed in the previous step to express the desired relationships among word embeddings. This function aims to reduce the discrepancy between the inner products of word embeddings and the logarithms of the chance ratios. The goal is to find word embeddings that reflect the likelihood of two words occurring in a connected manner. Then, GloVe applies repeated optimization techniques, such as gradient descent. During training, the model varies the word embeddings to reduce the variation in the dot product of the word embeddings and the logarithm of the chance ratios for the same word pairs in the co-occurrence matrix. This process is repeated until the model discovers embeddings that demonstrate how the words in the corpus are distributed. Once the word embeddings have been acquired, they should be capable of a variety of natural language processing tasks. These word embeddings capture the associations between words' meanings.

### 4.1.3 Contextual Word Embedding

Contextual word embeddings are a form of word representation model used in BERT that can capture the meaning of a word based on its context in a sentence or document. It includes three types of embeddings: word embeddings, position embeddings, and Segment embeddings. BERT has been pre-trained on a large corpus of text in order to acquire contextual word representations.

Tokenization plays a vital role in this approach. BERT uses WordPiece tokenization, which divides words into smaller subword units, rather than the conventional tokenization at the word level. This technique manages out-of-vocabulary words and reduces the size of the vocabulary, making BERT more efficient and adaptable to a variety of languages, including Bangla. Moreover, the attention masks allow BERT to disregard any padding tokens during training and evaluation in order to concentrate on the actual words. This assures that the contextual embeddings of the model are meaningful and contextually relevant since they represent the word's

meaning in the context of the entire sentence.

BERT reads the text in a bidirectional manner, considering the left and right context for each word. As the text is processed, contextual embeddings are generated for each token. These embeddings indicate the meaning of each word based on the context of the entire sentence.

# Chapter 5

# Experimental Design and Methodology

In this study, We have used 3 models i.e. BiLSTM, BiGRU, and BERT in order to perform the multiclass and binary classification task.

## 5.1 Bidirectional LSTM Model

BiLSTM is a recurrent neural network (RNN) architecture designed to process sequential data and capture long-term dependencies in both forward and backward directions. It is an extension of the standard LSTM model, which consists of three gates: a forget gate, an input gate, and an output gate. Here, the forget gate eliminates obsolete information from the cell state, while the input gate determines how much new information to store in the current cell state. In addition, the output gate determines how much of the cell state to output as the hidden state for the current time step [1].

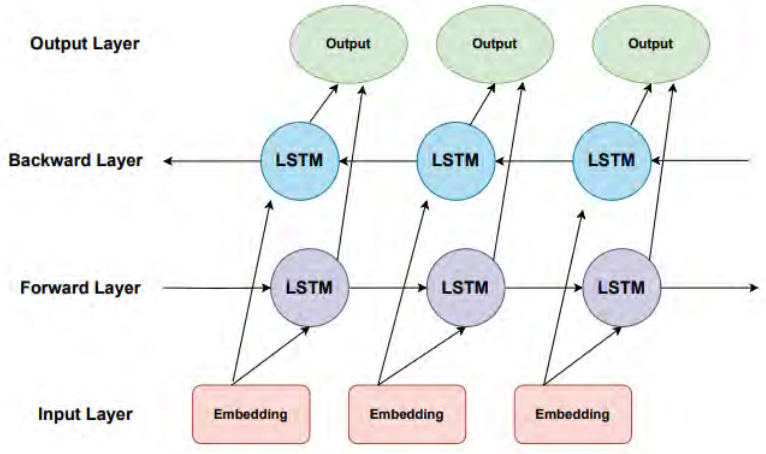



Figure 5.1: BiLSTM architecture

The architecture of a BiLSTM network consists of two LSTM layers, one processing the input sequence in the forward direction (from the beginning to the end of the sequence), and the other processing it in the backward direction (from the end to the beginning of the sequence). The forward LSTM receives an input vector at each time step and produces two outputs: the hidden state and the cell state. The hidden state represents the information learned up to that point in time, whereas the cell state assists the LSTM in retaining its long-term memory. In addition, similar to the forward LSTM, the backward LSTM generates a hidden state and a cell state at each time step. The outputs of both LSTM layers are then concatenated or otherwise combined to generate the model's final output.

Bidirectional LSTM can be represented as :

$$P(t) = combine(H(t)^f, H(t)^b)$$

Here, P(t) = The final probability vector

$$H(t)^f = LSTM\_forward(input\_sequence)$$

$$H(t)^b = LSTM\_backward(input\_sequence)$$

## 5.2 Bidirectional GRU Model

Moreover, BIGRU is another variant of the bidirectional recurrent neural network (RNN) architecture. It extends the conventional GRU model and has similarities to BiLSTM. In a GRU, there are two primary gates: the Reset Gate and the Update Gate. The reset gate is responsible for determining how much of the previous hidden state to neglect and how much of the new input to consider for the current time step, while the update gate is responsible for determining how much of the new input to consider. The update gate determines how much new information should be stored in its current hidden state [4].

Bidirectional GRU can be represented as :

$$h(t) = combine(h(t)^f, h(t)^b)$$

Here, h(t) = The final probability vector

$$h(t)^f = GRU\_forward(input\_sequence)$$

$$h(t)^b = GRU\_backward(input\_sequence)$$

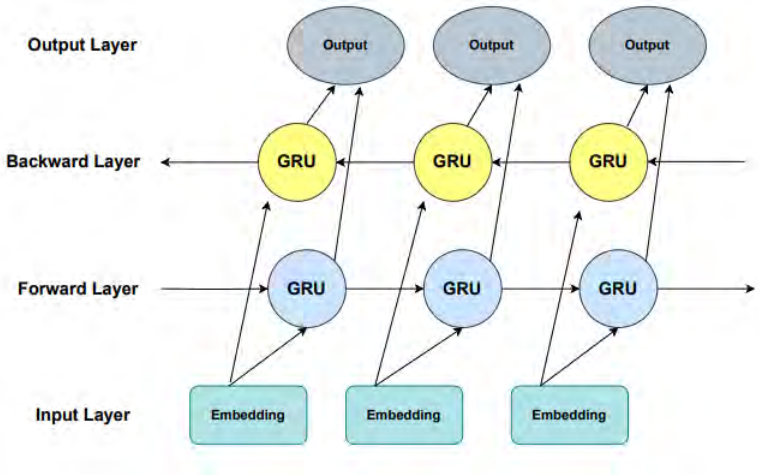



Figure 5.2: BiGRU architecture

BIGRU is designed to process sequential data and identify long-term dependencies in both forward and backwards directions. Forward BiGRU receives an input vector at each time step and generates two outputs: the forward hidden state and the combined update and reset gate vector. The Backward BiGRU, similar to the Forward BiGRU, generates two outputs at each time step: the backward hidden state and the combined update and reset gate vector.BIGRU networks have a similar architecture to BiLSTM networks, but instead of LSTM cells, they employ Gated Recurrent Units (GRUs). Forward BiGRU receives an input vector at each time step and generates two outputs: the forward hidden state and the combined update and reset gate vector.

## 5.3 BERT Model

We used another model for our cyberbullying detection task which is Bidirectional Encoder Representations from Transformers (BERT). BERT is a deep learning model that is used to handle natural language. It was developed by Google researchers in 2018, and it has since become one of the most significant models in NLP. BERT is a member of the Transformer Architecture family. The transformer is made up of an encoder and a decoder. For the BERT model, however, we only consider the encoder part of the transformer.

BERT is constructed using transformer architecture, which eliminates the need for sequential word processing. The transformer uses self-attention mechanisms to simultaneously attend to all words in a sentence. In addition, BERT is pre-trained using two unsupervised pre-training tasks: masked language modelling (MLM) and next sentence prediction (NSP) [9]. Random words in each sentence are masked in MLM, and the model is trained to predict the masked words based on their context. In NSP, BERT is given pairs of sentences and is trained to predict whether the second sentence is the actual next sentence following the first one in the original text.

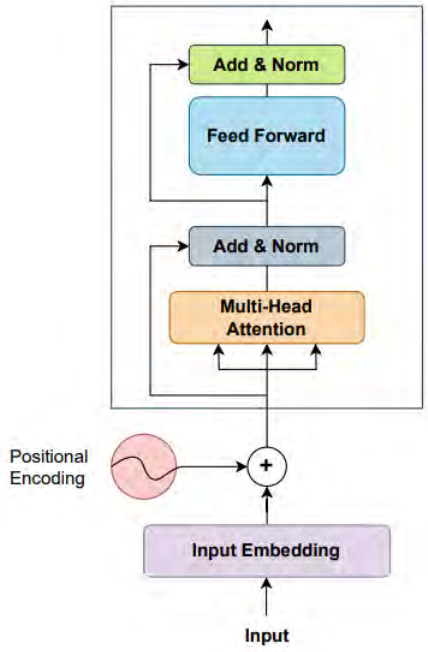



Figure 5.3: BERT architecture

In addition, unlike conventional models that read the text in a unidirectional manner, BERT is bidirectional, capturing contextual information from both the left and right sides of each word. Bidirectional context enables BERT to comprehend the meaning of a word based on the entire context in which it appears, resulting in a more precise comprehension of language.

## 5.4 Experimental Setup

This study includes the implementation of both binary and multiclass classification techniques for the purpose of detecting instances of cyberbullying.

### 5.4.1 Binary Classification

For the binary classification, we are required to detect if a comment is 'bully' or 'not bully'. Here, we have used 3 distinct deep learning architectures i.e. BiLSTM, BiGRU, and BERT. Firstly, we have used BiLSTM and BiGRU with FastText word embedding. Moreover, again with Glove embedding, we have again implemented both models to see which embedding techniques are well suited for our binary classification task. For all three models used here, we have split the dataset into 3 categories 80% for training, 10% for validation, and 10% for testing.

First, for the BiLSTM model, we used five layers to build it and trained it on our cyberbullying dataset. The first layer of the BiGRU model is an embedding layer derived from the two-word embedding techniques FastText and GloVe that we have

used. Each word in the vocabulary is mapped to a high-dimensional vector representation of size 300 here and the number of neurons is equal to the vocab_length * embedding dimension, which is the number of unique words in your vocabulary. In our dataset, the vocab_length is 67450. The second layer is a Bidirectional LSTM layer. This layer processes the input sequence in both forward and backwards directions, thereby capturing context from both sides of the sequence. There are 256 LSTM units in each direction (forward and backwards), resulting in a total of 512 LSTM units. The third layer is another Bidirectional LSTM layer with identical parameters. After the bidirectional LSTM layers, there is a Dense layer with 128 neurons and a Rectified Linear Unit (ReLU) activation function. This fully connected layer incorporates non-linearity and enables the model to learn complex feature relationships. Nonlinearity is introduced into the model using the ReLU activation function.

Given below is the equation for the ReLU function:

$$f(x) = max(0, x)$$

where, x = input of neuron

The final layer of the model is a dense layer with a sigmoid activation function and a single neuron. This layer is responsible for generating the final binary classification output, where the sigmoid function squashes the output to a range between 0 and 1. The sigmoid computation proceeds as follows:

$$f(x) = \frac{1}{1 + e^{-x}}$$

where, x = input of neuron

The model is compiled with binary cross-entropy as the loss function and the 'Adam' optimizer is used for model optimization; during training, accuracy is used as the evaluation metric. Additionally, 15 epochs are used to train the model. In addition, the binary cross-entropy loss quantifies the difference between the predicted probability and the actual label for each sample in the dataset. It encourages the model to generate probabilities that are high for the correct class and low for the incorrect class.

Binary cross-entropy is defined as:

$$-[y * log(p) + (1 - y) * log(1 - p)]$$

where y is the true label (0 or 1) and p is the predicted probability

In addition, we implemented this task using the BiGRU model and two word embedding techniques: GloVe and FastText. The BiGRU model is also trained with 15 epochs, the same number of layers, and with the same parameters as the BiLSTM model. However, instead of a BiLSTM layer, two BiGRU layers with 512 neurons (forward and reverse) are utilized in this instance. In this model, the bidirectional GRU layers process the input sequence in both forward and backwards directions,

capturing contextual information from both sides of every word.

We used BERT for our binary classification task lastly. We have imported a pre-trained BERT base model for the Bangla language named 'sagorsarker/bangla-bert-base' with the "num_classes" parameter set to 2 for binary classification. We have modified the imported model for our cyberbullying detection task in Bangla.

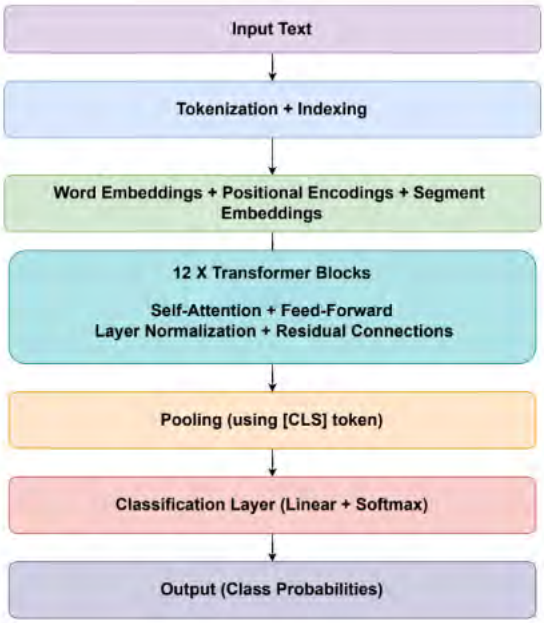



Figure 5.4: BERT base model with the internal architecture layers

In the BERT base model, The raw text is tokenized into individual words or subwords, and each token is assigned an index. Then, the index of each token is converted into a dense vector representation of size 768(hidden size of BERT). These word embeddings capture the contextual meaning of each token in the input text. The word embeddings have been added with positional encodings. These encodings indicate the position or sequence of tokens in the input sequence. The positional data is essential for the model to comprehend the sequence. In addition, BERT utilizes 12 transformer blocks and each transformer block is comprised of three principal sublayers:

**1. Self-Attention Mechanism:** This mechanism helps the model weigh the importance of different words in the context of each other. It consists of three learned linear transformations: query, key, and value. Self-attention allows the model to effectively capture long-range dependencies in the input text.

**2. Feed-Forward Neural Network:** The self-attention mechanism's outputs are processed by a feed-forward neural network. The feed-forward network incorporates non-linearity and enables the model to discover intricate patterns in the data.

**3. Layer Normalization and Residual Connections:** After each sub-layer (self-attention and feed-forward), layer normalization and residual connections are implemented. Layer normalization helps stabilize training, and residual connections allow the model to retain information from earlier layers.

After that, the pooled representation is obtained by applying pooling to the output of the [CLS] token, which represents the entire input sequence in a single vector. The pooled representation is fed through a linear layer followed by a softmax activation to make predictions for the classification task. In addition, the output comprises class probabilities, which indicate the likelihood that the input text belongs to various classes in the classification task. The class with the highest probability is the predicted class for the input text.

$$\sigma(\vec{z_i}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K$$

Where,

$\sigma$ = softmax

$\vec{z_i}$ = input vector

$e^{z_i}$ = input vector exponential function

K = number of classes

$e^{z_j}$ = output vector exponential function

In addition, three epochs are used to fine-tune the Bangla cyberbullying detection task. On top of that, AdamW has been used as an optimizer. AdamW is a variant of the Adam optimization algorithm, which was developed as a modification to the original Adam optimizer in order to resolve potential weight decay-related issues.

### 5.4.2 Multiclass Classification

For multiclass classification, our model must identify five distinct categories of bully and non-bully text: "not bully," "Sexual," "Troll," "Religious," and "Threat." BiGRU, BiLSTM, and BERT are also utilized here. BiGRU and BiLSTM models use GloVE and FastText word embedding, whereas BERT uses contextual word embedding. Similarly, for multiclass classification, our dataset is divided into 80% for training, 10% for validation, and 10% for testing.

Similar to the Binary classification, the multiclass classification BiLSTM model has 5 layers, with the first layer being an embedding layer with 300d representation and the number of neurons being equal to vocab_length * 300, where vocab_length

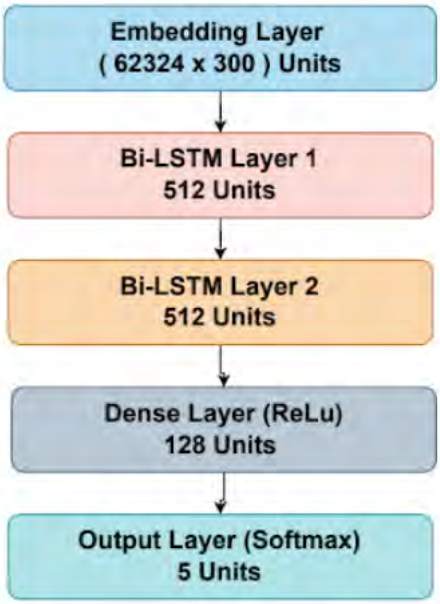



Figure 5.5: BiLSTM Model with Layers

equals 67450. In addition, the second layer is a BiLSTM layer with 256 units of forward-direction neurons and 256 units of backward-direction neurons for a total of 512 LSTM units. The third layer is an additional Bidirectional LSTM layer with the same parameters and 256 LSTM units in each direction, resulting in a total of 512 LSTM units. The fourth layer consists of 128 dense units with the ReLU activation function. This fully connected layer introduces non-linearity and enables the model to learn complex feature relationships. Nonlinearity is introduced into the model using the ReLU activation function. The final layer of the model is a Dense layer with five neurons, corresponding to the number of classes in the multi-class classification task, and a softmax activation function. This layer is responsible for generating the final multiclass classification output, where the softmax function converts logits to probabilities representing the likelihood of an input belonging to each of the 5 distinct classes, where each value represents the probability that the input belongs to the corresponding class.

Besides, BiGRU was used as a second model for multiclass classification. Similar to the BiLSTM, the same number of layers and parameters were applied here. In place of the two BiLSTM layers, however, we have utilized two BiGRU layers with 512 units each (forward and backwards). The initial Bidirectional GRU layer processes the input sequence in both forward and backwards directions, thereby capturing context from both sides of the sequence. In the forward pass, the input sequence is processed from the first to the last time step, and the GRU units' hidden states are updated accordingly. The final hidden state of the forward GRU captures contextual information from the entire forward sequence. In addition, in the backward pass, the input sequence is processed in reverse, beginning with the last time step to the first. The hidden states of the backwards GRU units are modified based on the input sequence in reverse order. The final hidden state of the backward GRU captures the contextual information from the entire backward sequence. For each

time step, the output of the Bidirectional GRU layer is the concatenation of the forward and backwards hidden states. The combination of forward and backward passes enables the model to incorporate bidirectional context, resulting in a more precise understanding of the Bangla language and enhanced performance on our cyberbullying detection task.

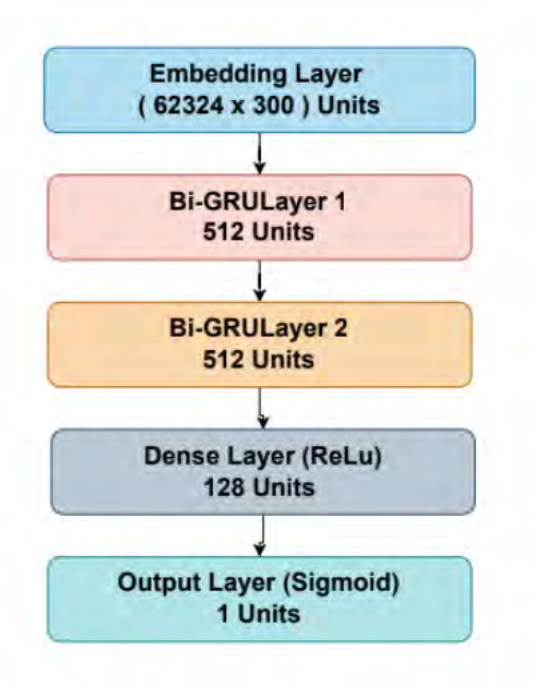



Figure 5.6: BiGRU Model with Layers

Here as for both BiGRU and BiLSTM, we have used sparse_categorical_crossentropy as a loss function. This loss function is frequently used for multiclass classification tasks involving class labels encoded as integers. It computes the loss of cross-entropy between the predicted probabilities and the actual integer class labels. Cross-entropy evaluates the difference between the predicted and actual probability distributions. Here, The term "sparse" refers to the fact that the target labels are not one-hot encoded. They are instead represented as integers that correspond to the class index. This indicates that the true class label has been provided as an integer rather than a one-hot encoded vector.

Also For both multiclass classification models, the Adam optimizer was utilized. Adam, which stands for "Adaptive Moment Estimation," is an extension of the stochastic gradient descent (SGD) optimization algorithm. Adam is renowned for its adaptive learning rate, which allows it to adjust the learning rate for each parameter based on the first and second moments of the gradients.

In addition, BERT has been used for the purpose of conducting multiclass classification in this study. Similar to binary classification, the 'sagorsarker/bangla-bert-base' model is used in this setting as well. The model is subjected to fine-tuning for our cyberbullying detection task, utilizing 3 epochs and applying the AdamW optimizer.

The BERT model architecture remains the same as the binary classification task, however, a few parameters are changed where the number of the classes is set from 2 to 5, in order to make the model fit for the multiclass classifier.

# Chapter 6

# Result and Error Analysis

The efficiency of our Deep learning-based system is evaluated in this section. We integrated GloVe and FastTest embedding with BiGRU and BiLSTM at the model implementation. Also, an integration of Contextual embedding with BRET is implemented as a classification model in our system dataset, consisting of 44,001 comments, where five types of categorical comments (Not Bully, Religious, Threat, Troll, and Sexual) were present in the dataset. For this, training models were created using 80% of the dataset's data and the remaining 20% were used as validation and testing for measuring the cyberbullying detection capability of the model. We used binary and multiclass classification in order to evaluate the bullying, non-bullying and 4 categories of bullying comments using the implemented models. The efficiency of our models is evaluated using several performance measures: Precision, Recall, F1 score and accuracy.

Precision is defined as the proportion of true positives to total predictions. Precision is how many true positive predictions a model makes out of all the positive predictions it makes. Simply put, precision is the percentage of accurately classified positives that the model correctly detected. Precision is especially helpful if the value of false positives remains high, which means that misclassifying a negative case as positive is undesirable. The precision formula is as follows:

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)}$$

Moreover, The value of recall represents the ratio between the number of right predictions and the total number of correct observations inside the sample space. Recall, commonly referred to as sensitivity or true positive rate, is a crucial machine learning evaluation parameter, especially for binary classification issues. The percentage of actual positive events that the model properly identified is known as recall. Recall provides the positive instances in the dataset the model accurately predicted as positive. The recall equation is:

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)}$$

Furthermore, The F1 score is a metric used in machine learning to assess how well a classification model performs. The model's precision and recall are combined into a single, well-balanced amount. When there is an unequal class distribution and we need to establish a midpoint between precision and recall, the F1 score is particularly helpful. Here's how we can calculate our F1 score:

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

Additionally, The performance of a classification model is typically evaluated in terms of accuracy in machine learning. It displays the percentage of accurately predicted cases among all the dataset's instances. Moreover, the percentage of correctly predicted events in all observations is called accuracy. A model's accuracy is deemed to be at its highest if and only if we have a symmetric dataset in which the values of FP and FN for the two classes are nearly equal. Other evaluation factors, such as the F1-score, may be taken into consideration since accuracy is not always the most appropriate option in many unbalanced data sets. The accuracy formula is straightforward:

$$Accuracy = \frac{(NumberofCorrectPredictions)}{(TotalNumberofPredictions)}$$

| Model | Precision | Recall | F1 Score | Accuracy |
|---|---|---|---|---|
| BiGRU+GloVE | 0.81 | 0.82 | 0.82 | 0.83 |
| BiLSTM+GloVE | 0.82 | 0.81 | 0.81 | 0.83 |
| BiGRU+FastText | 0.84 | 0.86 | 0.85 | 0.86 |
| BiLSTM+FastText | 0.84 | 0.86 | 0.85 | 0.86 |
| **Bangla BERT** | **0.89** | **0.89** | **0.89** | **0.90** |

Table 6.1: Model's Performance for Binary Classification

The results of our binary classification models are listed in Table 6.1. It shows that the BiGRU-GloVe model scored 0.81 in precession, 0.82 in the recall, 0.82 in F1 score, and 0.83 in accuracy whereas the BiLSTM-GloVe model scored 0.82 in

precession, 0.81 in the recall, 0.81 in F1 score and 0.83 in accuracy. Moreover, the BiGRU-FastText model scored 0.84 in precession, 0.86 in the recall, 0.85 in F1, and 0.86 in accuracy whereas the BiLSTM-FastText model performed 0.84 in precession, 0.86 in the recall, 0.85 in F1, and 0.86 in accuracy. However, The BERT model score is 0.89 in precession, 0.89 in recall, 0.89 in F1 score and 0.90 in accuracy. So, it is clearly visible that the highest precession, recall, F-1 score, and accuracy were obtained by the BERT for binary classification.

Table 6.2 contains the results of the models used for multiclass classification. It is listed in Table 6.2 that the GloVe+BiGRU model scored 0.72 in precession, 0.70 in the recall, 0.71 in F1, and 0.71 in accuracy whereas the GloVE +BiLSTM scored 0.72 in precession, 0.68 in the recall, 0.70 in F1, and 0.70 in accuracy. Moreover, the FastText +BiGRU scored 0.78 in precession, 0.77 in the recall, 0.77 in F1, and 0.78 in accuracy whereas the FastText +BiLSTM model scored 0.80 in precession, 0.74 in the recall, 0.76 in F1, and 0.77 in accuracy. However, The BERT model score is 0.86 in precession, 0.83 in recall, 0.85 in F1, and 0.85 in accuracy. Here we can see the BERT model has surpassed the other models in terms of precession(0.86), recall(0.83), F-1 score(0.85), and accuracy(0.85).

| Model | Precision | Recall | F1 Score | Accuracy |
|---|---|---|---|---|
| BiGRU+GloVE | 0.72 | 0.70 | 0.71 | 0.71 |
| BiLSTM+GloVE | 0.72 | 0.68 | 0.70 | 0.70 |
| BiGRU+FastText | 0.78 | 0.77 | 0.77 | 0.78 |
| BiLSTM+FastText | 0.80 | 0.74 | 0.76 | 0.77 |
| **Bangla BERT** | **0.86** | **0.83** | **0.85** | **0.85** |

Table 6.2: Model's Performance for Multiclass Classification

## 6.1 Comparative Result Analysis

Comparative result analysis for cyberbullying detection between two studies provides valuable insights into the effectiveness of different approaches and models. We have compared our work with another existing research in the literature that experimented with the same dataset for identifying and classifying cyberbullying content in digital communication. The analysis includes comparing the F1-score between two studies to assess the model's performance.

Table 6.3 shows the result comparison where for both binary and multiclass classification, our proposed model BERT performed better as compared with another research [17]. In that study, the authors introduced a hybrid neural network-based model for detecting expressions of bullying in the Bengali language, supporting both binary and multiclass classification. In the context of binary classification, the researchers leveraged a deep LSTM model consisting of 7 layers, resulting in an impressive F1 score of 0.82. Furthermore, the authors adopted a sophisticated hybrid

| Authors | Model | F1 Score (Binary) | F1 Score (Multiclass) |
|---|---|---|---|
| Ahmed et al. (2021) | NN+Ensemble | 0.82 | 0.84 |
| **Our Approach** | **BERT** | **0.89** | **0.85** |

Table 6.3: Comparative Result Analysis Between Two Works

approach for multiclass classification, which yielded a remarkable F1 score of 0.84. Their innovative strategies and model architectures led to significant performance improvements in both binary and multiclass scenarios. However, in our approach, We have achieved an F1-score of 0.89 for binary classification and 0.85 for multiclass classification using BERT which is better as compared with their work. Such comparative analyses help inform the field of cyberbullying detection, guiding researchers and practitioners toward more robust and efficient methods for addressing this critical issue in the digital realm.

## 6.2 Performances Evaluation of Best Performing Model i.e., BERT Using Secondary Testing Dataset

In order to ensure that our best-performing model, BERT, is as effective at identifying any Bengali online bullying comments, we have put it through further testing using a secondary dataset collected by ourselves.

For Binary classification, BERT achieved an f1-score of 0.87 using the testing dataset while it achieved 0.89 using the original dataset 10% testing split. In terms of class-wise performances, BERT also detected 89% bullying text and 86% non-bullying text correctly (Fig 6.1).

For Multiclass classification, BERT achieved a 0.82 f1 score using the secondary testing dataset. Moreover, for detecting class-wise bullying and non-bullying text, BERT successfully detected 91% not bully comments, 79% sexual comments, 71% troll comments, 85% religious comments and 70% threat comments (Fig 6.2).

| Classification Type | Precision | Recall | F1 Score | Accuracy |
|---|---|---|---|---|
| Binary | 0.87 | 0.88 | 0.87 | 0.88 |
| Multiclass | 0.85 | 0.79 | 0.82 | 0.82 |

Table 6.4: BERT's Performances Using Testing Dataset

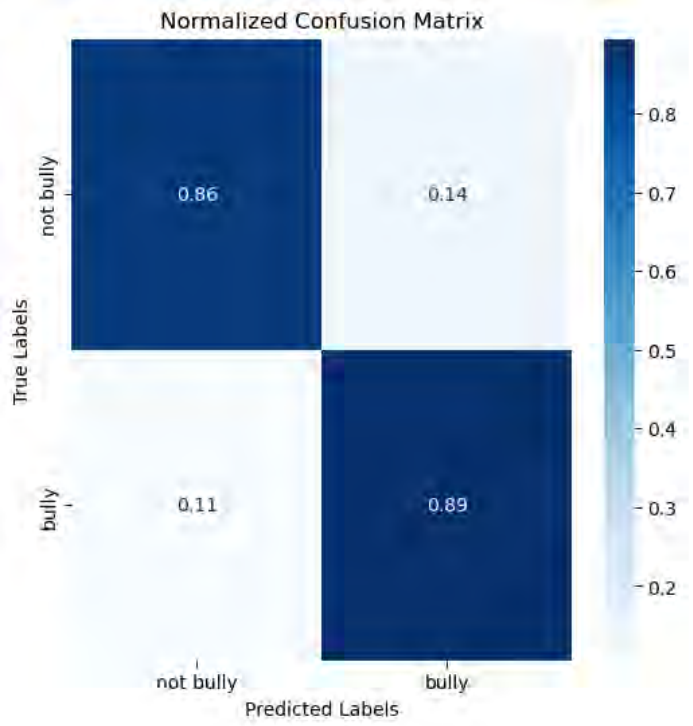



Figure 6.1: BERT Confusion Matrix for Binary Classification using Secondary Testing Dataset

## 6.3 Error Analysis

Besides, Figure 6.3 and Figure 6.4 show the Confusion Matrix of our proposed best-classifying model in both binary and multiclass classification i.e. BERT.

Figure 6.1 shows the confusion matrix Binary classification using BERT. As for binary classification, the data is categorized into two levels, bully and Not bully. As we can observe from the confusion matrix out of 100% of Not Bully comments the BERT successfully detected 85% of the Not Bully comments. The remaining 15% of the Not Bully comments were falsely classified. The reasons might be because of the data imbalance between the two classes. The allocated data for non-bully comments is 34.9% and the remaining data for the bully comments. On the other hand, the BERT detected 93% of the Bully comments successfully whereas only 7% of the Bully comments were falsely detected by the comments. The percentage of misclassification for bully text is less where the main reason could be the data allocation is more for bully comments. However, in terms of class-wise performance for binary classification, BERT surpassed both BiGRU and BiLSTM with the highest number of times accurately detecting bully comments. But BiGRU with FastText has shown slightly better performances capturing non-bully comments with 86% times accurately compared with BERT's 85%.

Figure 6.2 shows the confusion matrix Multiclass classification using the BERT. The model successfully detected 88% of the Not Bully data successfully, The remaining 12% falsely classified Not Bully comments were detected by the model as Sexual(4%), Troll(7%), and Religious (1%). Again, The model correctly identified

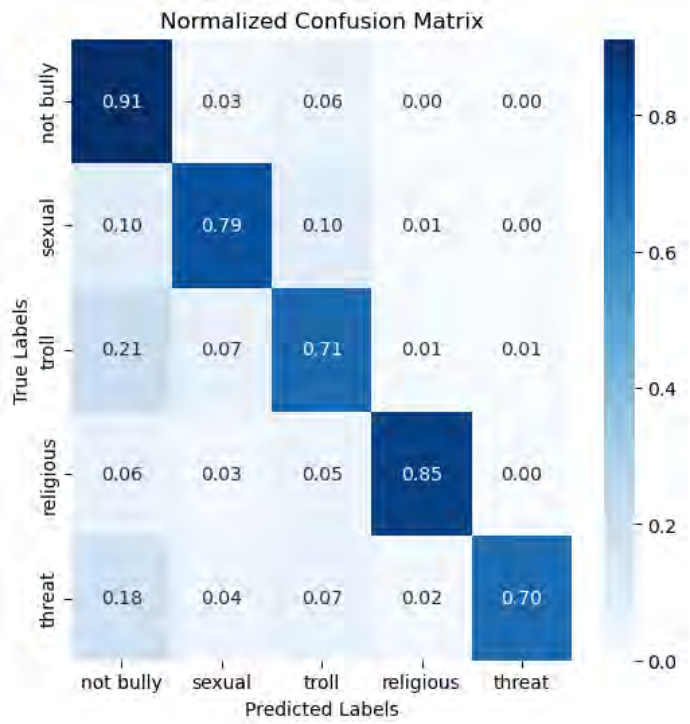



Figure 6.2: BERT Confusion Matrix for Multiclass Classification using Secondary Testing Dataset

87% of the sexual comments; the remaining 13% were incorrectly identified by the model as Not Bully (5%), Troll (7%), and Religious (1%). Moreover, The BERT classification model successfully detected 75% of the Troll comments. The rest were falsely classified comments that were detected as Not Bully (15%), Sexual (8%), and Religious (1%). Furthermore, 90% of the religious comments were detected successfully by the classification model. The rest religious 10% of the comments were classified as Not Bully (4%), Sexual(2%), and Troll(4%). Lastly, our best multiclass classification model BERT successfully detected 77% of the Threat comments, the remaining 23% of comments were falsely detected as Not Bully (15%), Sexual(2%), Troll (7%), and Religious (1%). BERT struggled a detect troll and threat comments as we can observe in Figure 6.2 where 25% misclassification for troll and 23% comments were incorrectly classified by BERT. For both of the classes, BERT predicted them as not-bully 15% times which indicates the complexity and ambiguity inherent in online text data. These categories may share linguistic similarities with non-bullying remarks, making it difficult for the model to distinguish between them consistently. In addition, the model's training data do not provide enough distinctions for these subtle classes, resulting in occasional misclassification. However, Compared with the other two models in multiclass classification, in terms of class-wise performance, BERT surpassed both BiLSTM and BiGRU models while detecting each distinct class accurately with better accuracy.

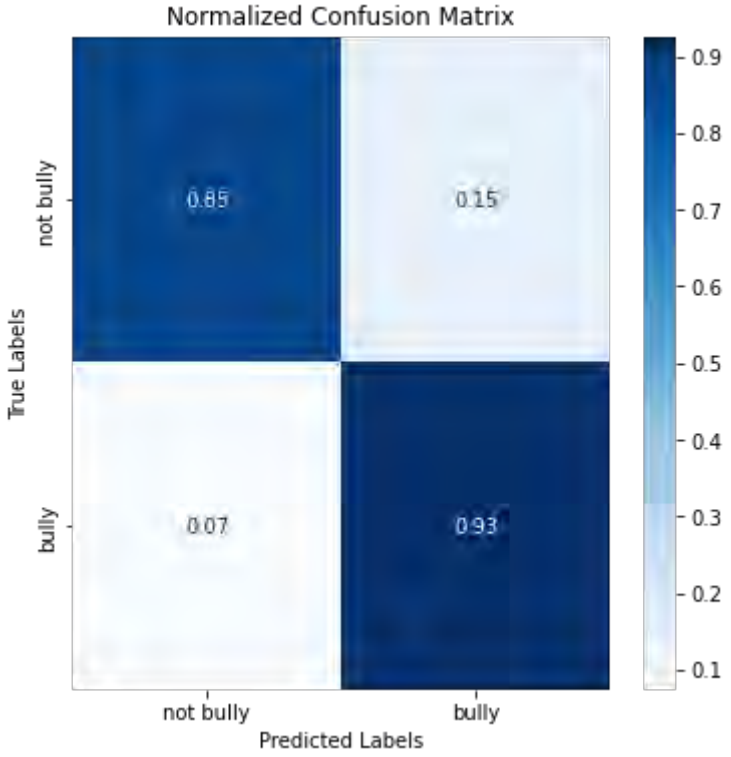



Figure 6.3: BERT Confusion Matrix for Binary Classification

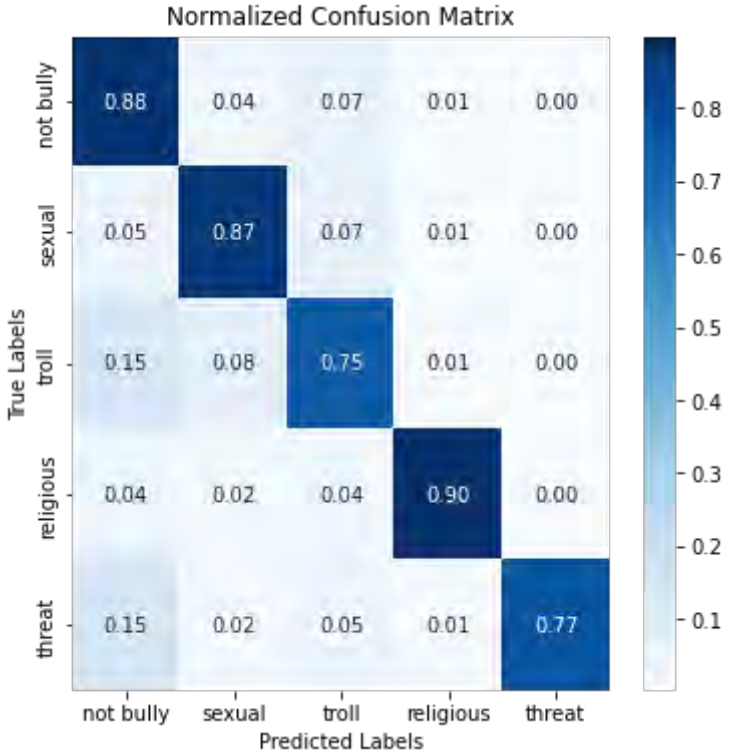



Figure 6.4: BERT Confusion Matrix for Multiclass Classification

# Chapter 7

# Conclusion

we have developed our research with a comprehensive discussion on the detection of cyberbullying in social media platforms for the Bangla language, there are numerous methods to detect harmful behaviour using various supervised machine learning algorithms. However, We have examined 3 different deep learning models i.e. BiGRU, BiLSTM and BERT and word embeddings i.e. GloVE, FastText and contextual embeddings with BERT for this study. Through extensive comparison, it has been determined that each model has its own strengths and weaknesses. BiLSTM and BiGRU, with their recurrent neural network architectures, achieved competitive results in cyberbullying detection tasks, demonstrating the significance of sequential context understanding. In contrast, the BERT model, with its contextualized word embeddings and transformer architecture, outperformed traditional recurrent models in a number of aspects. It demonstrates extraordinary abilities in capturing subtle linguistic nuances and context, which are essential for the accurate identification of cyberbullying examples. Additionally, If we can make our proposed model i.e. BERT useful and usable then it can contribute a lot in terms of preventing bullying from society as well as online social media. By keeping track of the proportion of damnation and slander terms within a post, our overall effort to prevent abusive behaviour can be effective in preventing aggressive attitudes. As a result of rapidly growing technology people easily get involved in harassing or pestering activities. Our proposed model can contribute positively while detecting abusive languages for Bangla. Though the usage of social media has both positive and negative sides, a well-trained BERT model can contribute positively to minimizing the toxic characteristics by taking responsible actions. this could make a great positive impact if our proposed model and algorithms perform accurately, as detecting cyberbullying and taking action against can enhance the physical, mental and emotional conditions of individuals. Moreover, It can bring a positive outcome to technology and social media and the proposed model will help the generation through various information and techniques from the invasions of social media reprimands.

# Bibliography


[1] S. Hochreiter **and** J. Schmidhuber, **?**Long short-term memory,**?** *Neural computation*, **jourvol** 9, **number** 8, **pages** 1735–1780, 1997.

[2] R. C. Vreeman **and** A. E. Carroll, **?**A systematic review of school-based interventions to prevent bullying,**?** *Archives of pediatrics & adolescent medicine*, **jourvol** 161, **number** 1, **pages** 78–88, 2007.

[3] Y. Yu, T. R. Bhangale, J. Fagerness **andothers**, **?**Common variants near FRK/COL10A1 and VEGFA are associated with advanced age-related macular degeneration,**?** *Human molecular genetics*, **jourvol** 20, **number** 18, **pages** 3699–3709, 2011.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre **andothers**, **?**Learning phrase representations using RNN encoder-decoder for statistical machine translation,**?** *arXiv preprint arXiv:1406.1078*, 2014.

[5] J. Pennington, R. Socher **and** C. D. Manning, **?**Glove: Global vectors for word representation,**?** **in***Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* 2014, **pages** 1532–1543.

[6] P. Bojanowski, E. Grave, A. Joulin **and** T. Mikolov, **?**Enriching word vectors with subword information,**?** *Transactions of the association for computational linguistics*, **jourvol** 5, **pages** 135–146, 2017.

[7] S. Nikiforos, S. Tzanavaris **and** K. L. Kermanidis, **?**Bullying in virtual learning communities,**?** **in***GeNeDis 2016* Springer, 2017, **pages** 211–216.

[8] S. A. Özel, E. Saraç, S. Akdemir **and** H. Aksu, **?**Detection of cyberbullying on social media messages in Turkish,**?** **in***2017 International Conference on Computer Science and Engineering (UBMK)* 2017, **pages** 366–370. DOI: 10.1109/UBMK.2017.8093411.

[9] J. Devlin, M.-W. Chang, K. Lee **and** K. Toutanova, **?**Bert: Pre-training of deep bidirectional transformers for language understanding,**?** *arXiv preprint arXiv:1810.04805*, 2018.

[10] H. Sohn **and** H. Lee, **?**MC-BERT4HATE: Hate Speech Detection using Multichannel BERT for Different Languages and Translations,**?** **in***2019 International Conference on Data Mining Workshops (ICDMW)* 2019, **pages** 551–559. DOI: 10.1109/ICDMW.2019.00084.

[11] R. Alshalan **and** H. Al-Khalifa, **?**A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere,**?** *Applied Sciences*, **jourvol** 10, **number** 23, 2020, ISSN: 2076-3417. DOI: 10.3390/app10238614. **url**: https://www.mdpi.com/2076-3417/10/23/8614.

[12] M. Corazza, S. Menini, E. Cabrio, S. Tonelli **and** S. Villata, **?**A Multilingual Evaluation for Online Hate Speech Detection,**?** *ACM Trans. Internet Technol.*, **journvol** 20, **number** 2, **march** 2020, ISSN: 1533-5399. DOI: 10.1145/3377323. **url**: https://doi.org/10.1145/3377323.

[13] M. Mozafari, R. Farahbakhsh **and** N. Crespi, **?**A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media,**?** **in***Complex Networks and Their Applications VIII* H. Cherifi, S. Gaito, J. F. Mendes, E. Moro **and** L. M. Rocha, **editors**, Cham: Springer International Publishing, 2020, **pages** 928–940, ISBN: 978-3-030-36687-2.

[14] S. Nikiforos, S. Tzanavaris **and** K. L. Kermanidis, **?**Virtual learning communities (VLCs) rethinking: influence on behavior modification—bullying detection through machine learning and natural language processing,**?** *Journal of Computers in Education*, **pages** 1–21, 2020.

[15] G. I. Sigurbergsson **and** L. Derczynski, **?**Offensive Language and Hate Speech Detection for Danish,**?** English, **in***Proceedings of the 12th Language Resources and Evaluation Conference* Marseille, France: European Language Resources Association, **may** 2020, **pages** 3498–3508, ISBN: 979-10-95546-34-4. **url**: https://aclanthology.org/2020.lrec-1.430.

[16] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain **and** F. B. Ashraf, **?**Bangla Text Dataset and Exploratory Analysis for Online Harassment Detection,**?** *ArXiv*, **journvol** abs/2102.02478, 2021.

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

[18] M. S. Jahan **and** M. Oussalah, **?**A systematic review of Hate Speech automatic detection using Natural Language Processing,**?** *ArXiv*, **journvol** abs/2106.00742, 2021.

[19] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López **and** M. T. Martín-Valdivia, **?**Comparing pre-trained language models for Spanish hate speech detection,**?** *Expert Systems with Applications*, **journvol** 166, **page** 114 120, 2021, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.114120. **url**: https://www.sciencedirect.com/science/article/pii/S095741742030868X.

[20] N. Romim, M. Ahmed, H. Talukder **and** M. Saiful Islam, **?**Hate Speech Detection in the Bengali Language: A Dataset and Its Baseline Evaluation,**?** **in***Proceedings of International Joint Conference on Advances in Computational Intelligence* M. S. Uddin **and** J. C. Bansal, **editors**, Singapore: Springer Singapore, 2021, **pages** 457–468, ISBN: 978-981-16-0586-4.

[21] M. Scardamalia **and** C. Bereiter, **?**Knowledge building: Advancing the state of community knowledge,**?** **in***International handbook of computer-supported collaborative learning* Springer, 2021, **pages** 261–279.