Master of Science in Computer Science and Engineering

Identification of Cyberbullying Bangla Linguistic Texts Using Deep Learning and Transformer based Approaches

by

Md Khalid Saifullah

ID: 20MCSE036P

This thesis is submitted in partial fulfilment of the requirements for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science & Engineering

Chittagong University of Engineering and Technology (CUET)

Chittagong – 4349, Bangladesh

June 2024

CERTIFICATION

The thesis titled **Identification of Cyberbullying Bangla Linguistic Texts Using Deep Learning and Transformer based Approaches s**ubmitted by **Md Khalid Saifullah**, Roll No: **20MCSE036P**, Session: **2020-2021** has been accepted as satisfactory in partial fulfilment of the requirement for the degree of Master of Science in Computer Science & Engineering on 29 June 2024.

BOARD OF EXAMINERS

1. ____
Dr. Muhammad Ibrahim Khan Chairman (Supervisor)
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

2. ____
Dr. Mohammad Shamsul Arefin Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

3. ____
Dr. Kaushik Deb Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

4. ____
Dr. Abu Hasnat Mohammad Ashfak Habib Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

5. ____
Dr. Mohammad Shorif Uddin Member (External)
Professor
Vice Chancellor, Green University of Bangladesh, Dhaka, Bangladesh.

CANDIDATE’S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Md Khalid Saifullah
ID: 20MCSE036P

To my parents,

To my guiding stars, my parents: your love lights the path of my accomplishments.

This thesis is dedicated to you.

Acknowledgement

I extend my heartfelt gratitude to the numerous individuals whose time, guidance, and assistance have been instrumental in bringing this thesis to fruition. I owe a profound debt of gratitude to my supervisor, Professor Dr. Muhammad Ibrahim Khan, and co-supervisor Iqbal H Sarker, whose unwavering support and inspiration have steered me towards a meaningful outcome on this subject. Special appreciation goes to Md. Rajib Hossain, Ph.D., for generously sharing his invaluable expertise.

I am indebted to my defense committee for their insightful comments and invaluable suggestions. My sincere thanks to the Department of Computer Science and Engineering (CSE) at Chittagong University of Engineering and Technology (CUET) for accepting me as a graduate student and providing the platform to present my work. I am also grateful to the faculty members of CSE, CUET, whose guidance and assistance have been invaluable in the preparation of this thesis.

I am deeply grateful to all those with whom I have interacted, formally or informally, for their knowledge sharing and consultation throughout the research process.

I must express my profound appreciation to my parents and spouse for their unwavering support, understanding, and for managing family responsibilities, enabling me to pursue this research endeavour. This achievement would not have been possible without their sacrifice and encouragement. Finally, I am thankful to the Almighty Allah for granting me the strength and capability, both mentally and physically, to complete this thesis.

Abstract

In today's digital era, social media platforms such as Facebook, Twitter, and YouTube play crucial roles in facilitating idea expression and interpersonal connections. However, alongside increased connectivity, these platforms have inadvertently facilitated negative behaviors, notably cyberbullying. While extensive research has delved into cyberbullying in high-resource languages like English, there remains a significant dearth of resources for low-resource languages such as Bengali, Arabic, Tamil, and others, particularly concerning language modeling. This study aims to bridge this gap by developing a cyberbullying text identification system, named BullyFilterNeT, tailored specifically for social media texts, with Bengali serving as a test case. The intelligent BullyFilterNeT system effectively tackles challenges associated with Out-of-Vocabulary (OOV) words inherent in non-contextual embeddings and addresses the limitations of context-aware feature representations. To provide a comprehensive analysis, three non-contextual embedding models—GloVe, FastText, and Word2Vec—are developed for feature extraction in Bengali. These embedding models are integrated into classification models employing both statistical methods (SVM, SGD, Libsvm) and deep learning architectures (CNN, VDCNN, LSTM, GRU). Furthermore, the study utilizes six transformer-based language models; mBERT, bELECTRA, IndicBERT, XML-RoBERTa, DistilBERT, and BanglaBERT to overcome shortcomings observed in earlier models. Notably, the BanglaBERT-based BullyFilterNeT achieves the highest accuracy of 88.04% in our test set, demonstrating its efficacy in identifying cyberbullying text in the Bengali language.

Keywords: Cyberbullying; large language modelling; deep learning; transformers models; natural language processing (NLP); fine tuning; OOV; harmful messages

TABLE OF CONTENTS

Preliminaries **Pages**

Main Thesis Report

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

Abbreviation	**De-abbreviation**
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short Time Memory
CNN	Convolutional Neural Networks
ELECTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurately
GloVe	Global Vectors for Word Representation
GRU	Gated Recurrent Units
GPU	Graphics Processing Unit
IDF	Inverse Document Frequency
IndicBERT	Indic Languages BERT
LSTM	Long Short-Term Memory
MA	Macro Average
ML	Machine Learning
mBERT	Multilingual BERT
NLP	Natural Language Processing
OOV	Out of Vocabulary
RoBERTa	Robustly optimized BERT approach
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TF	Term Frequency
TPR	True Positive Rate
WA	Weighted Average
VDCNN	Very Deep Convolutional Neural Networks
XML	Cross-lingual Language Model

Chapter 1

Introduction

1.1 Background

Bullying has long been a part of human interaction, and with the rise of digital communication platforms, cyberbullying quickly emerged. Social media platforms like Facebook and Twitter facilitate constant communication with anyone at any time. Today, these platforms are central to social interactions and allow the formation of new relationships, often anonymously. This openness, however, increases the risk of exposure to harmful situations, including grooming, sexually inappropriate behavior, signs of depression, suicidal thoughts, and cyberbullying. The ability of social media to provide 24/7 access and anonymity makes it an attractive avenue for bullies to target victims beyond the school environment. Early detection of cyberbullying is crucial to minimize harm. However, due to linguistic differences and the influence of various social media factors (such as age, number of likes, and comments), current keyword-based or manual detection methods are inadequate for identifying cyberbullying in Bangla text. This underscores the need for developing an intelligent text identification system that uses machine learning to effectively detect and validate instances of social media bullying in Bangla.

Social media offers vast communication opportunities but also heightens online risks and threats, impacting people worldwide. Automatic detection of harmful messages is crucial for effective prevention. While significant research has focused on English content, Bangla content has often been neglected. However, as the 7th most spoken language, with the rise of the Unicode system and increased Internet use, Bangla usage on social media is growing. Despite this, there has been limited research on monitoring Bangla text on social media due to the lack of extensive annotated corpora, named dictionaries, and morphological analyzers, necessitating deeper analysis of the Bangla language. To address issues in online posts or conversations, machine learning algorithms and user-specific data analysis have shown better accuracy. Various machine learning techniques have been proposed for English, but applying these methods to other languages can lead to false detections, especially when content shifts from formal English to verbal abuse or sarcasm. Additionally, performance may vary due to linguistic differences and the socio-emotional behaviors of different populations. This thesis examines the performance and accuracy of widely used machine learning approaches on Bangla text.

In recent years, deep learning models have achieved remarkable success across various domains. Unlike traditional machine learning methods, which rely on human-generated features, deep learning models require vast amounts of data and can autonomously extract features through various neural network architectures, learning from their mistakes along the way [8]. These neural network models typically consist of multiple hierarchical layers, enabling layer-by-layer feature extraction and the application of non-linear activation functions. This allows for the modeling of complex features and the discovery of hidden deep features within texts [9]. The most commonly used deep learning models in sentiment analysis include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, Bidirectional LSTM (BiLSTM) networks, and Gated Recurrent Units (GRU) [10]. Specifically, CNNs are adept at capturing local features from aspect terms and sentiment terms.

Text identification using deep learning and transformer-based models represents a powerful and advanced approach for tackling classification tasks. Models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) utilize the capabilities of artificial neural networks to discern complex patterns and relationships from raw data. Transformers, a specific deep learning model architecture, excel at capturing long-range dependencies, making them highly effective for natural language processing tasks. A major benefit of deep learning and transformer-based models is their ability to automatically extract meaningful features from input data, which negates the need for manual feature engineering. By learning from large datasets, these models can identify intricate patterns and non-linear relationships, enabling them to manage complex classification tasks that traditional methods find challenging [42].

Moreover, deep learning models facilitate end-to-end learning, where the entire model is trained jointly, including the feature extraction, representation, and classification stages. This allows for the seamless integration of different components, optimizing the learning process and enhancing performance. Transformer-based models have revolutionized the field of Natural Language Processing (NLP), enabling breakthroughs in machine translation, text classification, sentiment analysis, and more. Their attention mechanisms, which capture dependencies between different positions within the input sequence, make them highly effective in understanding and processing textual data.

Bangla is the 7th most spoken language and its use over social media is increasing due to the popularity of the Unicode system and growing internet usage. Research on monitoring social media activity in Bangla text remains limited due to insufficient annotated corpora, dictionaries, and morphological analyzers. There is a need for further analysis from a Bangladeshi perspective to address this gap effectively. Cyberbullying has received significant research attention in recent years due to its rapid spread online, but Bangla cyberbullying research lags far behind to ensure cyberspace security. Therefore, data resources and models for Bengali need to be developed.

1.2 Problem Statement

Cyberbullying text identification refers to the process of detecting and recognizing instances of cyberbullying in written digital communications, such as text messages, emails, social media posts, or other online interactions. The rapid growth of media platforms like Facebook, Twitter, and YouTube has transformed communication, allowing individuals to express opinions on various topics. However, this has also led to the spread of offensive and hateful content. Cyberbullying, a significant issue, can cause psychological distress and undermine respectful conversations. According to research, this kind of conduct happened on Facebook and Twitter quite a bit. Among Bangladesh's 80.83 million Internet users [41], more than 90% regularly use Facebook, with the majority being young and vulnerable, necessitating urgent protection measures. Lately, numerous studies have focused on high-resource languages such as English.

This emphasis is attributed to the existence of adequately annotated cyberbullying corpora, pre-trained models for text-to-feature extraction, pre-trained cyber bullying identification models, and a suite of finely tuned hyperparameters [9]. Nevertheless, Bengali stands as the seventh most widely spoken language globally, with approximately 245 million people in Bangladesh and two states of India conversing in Bengali [16]. The growing popularity of Bangla on social media is attributed to the widespread acceptance of the Unicode system and the increasing use of the Internet [17]. Consequently, a substantial volume of Bengali bullying texts has proliferated across the unstructured web. Manually identifying these unstructured Bengali texts is impractical and financially burdensome. To address these challenges, the development of a cyberbullying identification system becomes imperative for government policymakers and security agencies. However, the scarcity of annotated corpora related to bullying, the absence of domain-specific pre-trained feature extraction and classification models, and the unavailability of well-tuned hyperparameters for domain-centric tasks pose significant obstacles. In recent years, considerable research has been dedicated to Bengali text classification [15], authorship attribution [14], sentiment analysis [2], and emotion classification [5]. However, the exploration of Bengali cyberbullying identification from textual data has been relatively limited [1, 3, 4]. Notably, the predominant approaches in existing research involve TF-IDF and non-contextual embedding-based (i.e., GloVe, Fast Text, Word2Vec) feature extraction, alongside statistical, CNN, and LSTM-based classification models. It is worth noting that the TF-IDF-based feature extractor falls short in capturing semantic meaning-based text features, while non-contextual embeddings like GloVe [26], FastText [6], and Word2Vec [22] struggle to extract context-aware features. To address these shortcomings, this research employs transformer-based language models. These models excel in extracting contextual text features, thus overcoming the limitations associated with traditional classification models. This shift is crucial for advancing the effectiveness of cyberbullying identification in the Bengali language.

1.3 Significance of the Research

The continually evolving landscape of online platforms, including Twitter, Facebook, Reddit, and others, has instigated extensive research into the identification and categorization of undesirable texts in recent years. This research spans diverse domains, addressing aggression classification [23], hate speech detection [11], abuse detection [24], toxicity classification [18], misogyny classification [21], trolling identification [8], cyberbullying detection [25], and offensive text classification [25]. While a substantial body of research has been dedicated to various languages, with a predominant focus on English, this work provides a concise summary of studies addressing cyberbullying detection/classification, and related topics in Bengali languages. The study includes an overview of studies conducted in English, Hindi, Arabic, and other languages. Additionally, Kumar et al. [19] presents an aggressive language identification dataset featuring three categories covert, and non-aggressive with annotations in both English and Hindi, encompassing 15,000 posts/comments on aggression.

Aroyehun et al. developed deep neural network-based English models employing data augmentation and a pseudo-labelling method. Employing LSTM and CNN-LSTM approaches, their system achieved macro F1-scores of 0.64 and 0.59, respectively. In another study, [28] utilized the TRAC-2 [20] dataset and a bootstrap aggregating-based ensemble with fine-tuned BERT models to detect violence and misogyny. They attained an

80.3% weighted F1 score on the test set of English social media posts. [30] compiled the Offensive Language Identification Dataset (OLID) consisting of 14,000 English tweets. They established a three-layer hierarchical annotation schema to detect, classify, and identify the targets of texts, utilizing SVM, BiLSTM, and CNN for baseline evaluation. CNN outperformed competitors in all three levels, achieving macro F1 values of 0.80, 0.69, and 0.47. Furthermore, Founta et al. [10] provided a dataset comprising 80,000 tweets categorized into hateful, abusive, spam, and normal classes. They employed a comprehensive methodology to address ambiguous categories. Additionally, Davidson et al. [7] created a dataset of 25,000 tweets categorized into hate, offense, and neither. The best macro F1-score of 0.90 was achieved using logistic regression with TF-IDF and n-gram features.

Previous research predominantly focused on high-resource languages, employing transformer-based language models with extensive standard bully identification corpora. However, limited attention has been given to low-resource languages like Bengali and Hindi. Research in low-resource bully identification often relies on non-contextual embedding models, such as GloVe, FastText, and Word2Vec. These embeddings face challenges in overcoming Out-of-Vocabulary (OOV) issues and extracting local and global contextual features specific to low-resource languages. In response to these challenges, this study introduces the **BullyFilterNeT** system. The system systematically develops a Bengali bully identification corpus and empirically evaluates various statistical, deep learning, and transformer-based language models. Ultimately, the top-performing model is selected to address the specific requirements of Bengali bully identification.

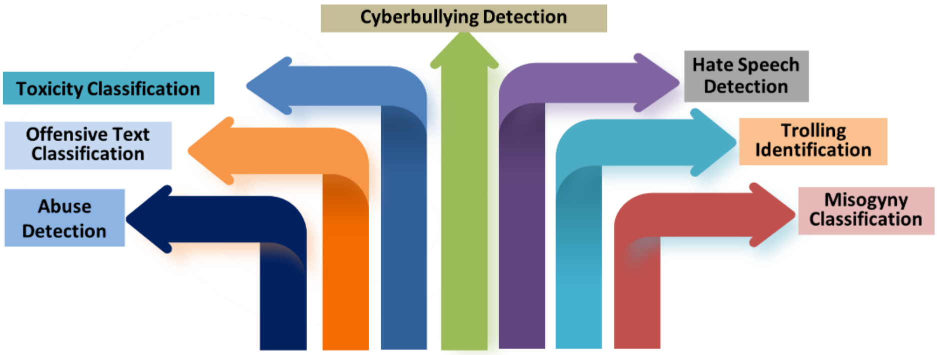


Fig. 1.1. Cyberbullying Filed Detection Diversity

1.4 Motivation

Cyberbullying that is motivated by various factors may demean the victim’s self-esteem, academic performance and emotional state [4-5]. According to [6], the psychological implications of cyberbullying have been associated with lower school performance, self-reported sadness, anger, fear and depression. This leads to results suggesting that suicidal thoughts and self-harm may be a consequence of having cyberspace as a bullying platform. Nevertheless, these facts only address one issue while the severity of their consequences is immense. Young people’s mental health necessitates early detection of cyberbullying attempts. To identify such threats, online content monitoring may be effective but again manual moderation would prove to be impractical due to the vast amount of information which social media carries thus giving rise to the need for an automated system that can process and help detect potential dangers [39].

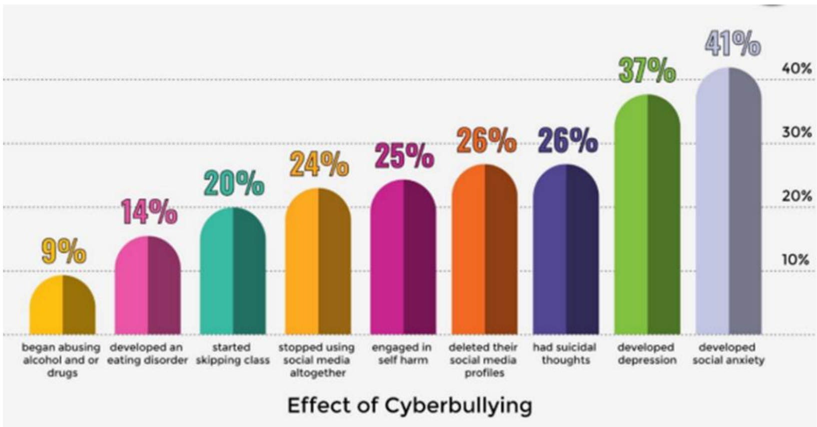


Fig. 1.2. Effect of Cyberbullying

During the recent years, many campaigns have been introduced against cyberbullying to enhance internet safety for children. Some examples include an anti-cyber bullying program known as KiVa (http://www.kivaprogram.net/) and the `Non-au harcelement' campaign in France; Belgian governmental initiatives and helplines which are in turn sources of information on online safety, e.g. clicksafe.be, veiligonline.be and mediawijs.be [38].

Parental control tools like NetNanny offer valuable assistance by blocking unsuitable or unwanted content. Similarly, certain social networks employ moderation tools reliant on keywords to flag potentially harmful material, like profanity or insults. However, these methods often fall short in identifying implicit or subtle forms of cyberbullying, where harmful behaviour may not be expressed explicitly. This highlights the necessity for more advanced systems capable of surpassing mere keyword recognition, thereby enhancing the effectiveness of cyberbullying detection efforts [38].

In the study undertaken [39], it was found that cyber victimization rates among youths were ranging between 20% and 40%. Research [39] focused on 12 to17 year olds living in United States and ascertained that at least 72% of them had experienced cyber bullying at least once over a period of 3. As per [39], it was reported that out of those interviewed, a total of 29% had suffered online harassment at some point; this survey covered individuals aged between nine and twenty-six years old from the USA, Canada, Australia and UK. According to a survey involving two thousand Flemish secondary school students (age range: 12-18), these findings showed that 11% of them had been bullied through the internet within half-year before this research [39]. Lastly, according to the large-scale EU Kids Online Report released in 2021 [39], for example, it is suggested that approximately) one fifth (20%) of all children aged between eleven and sixteen have already been confronted with hate messages on the net. Moreover, an increase in exposure of young children to cyber bullying is noted by a margin of twelve percent as compared to what was witnessed in the year 2010 thus highlighting its increasing influence globally. Due to widespread use of social networks without any privacy settings many Bangladeshi teenagers face threats and online harassment.

Bangladesh has a total of 80.83 million internet users [41]. In fact, over 90% of these social media users use Facebook alone, especially young and vulnerable people who need to be protected against any form of danger. The popularity of the Unicode system and the ever-increasing use of the Internet have contributed to this rise in the use of Bangla language on social media networks [10]. Nonetheless, there is minimal attention given to Bangla text for social media activity monitoring since it does not have numerous annotated corpora or name dictionaries and morphological analyzer systems [10] that require a detailed examination into Bangladesh’s point of view. Consequently, cyber bullying has gained much interest among researchers over the recent years due to its widespread nature. Despite being ranked as the third most spoken language in India after Hindi and Telugu, Bengali which is spoken by millions of people globally, still lags behind other Indian languages like Hindi when it comes to research on cyber bullying issues in Bengali language. Furthermore, with more advanced technology that comes at a cheaper price plus incentives from relevant Governments; Bangla document analysis is now more relevant than ever before.

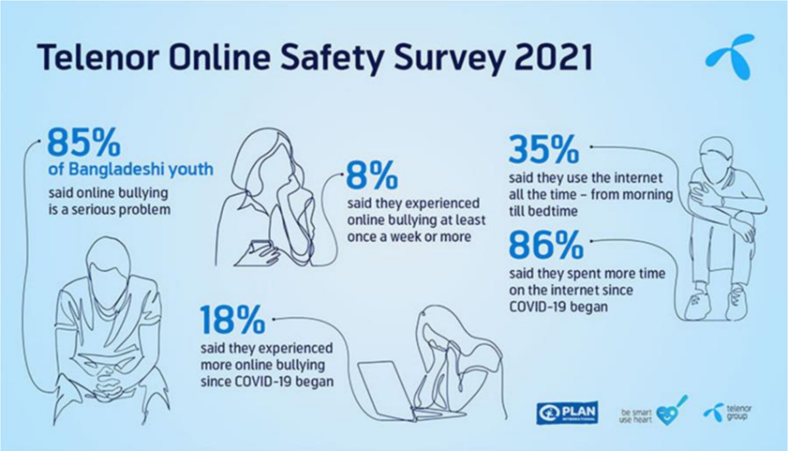


Fig. 1.3. Effect of Cyberbullying in Bangladesh Context

Generally, social media data are characterized by brevity, noise, lack of structure, and sometimes a mixture of multiple languages. Consequently, traditional methods of bullying detection such as guidelines, human moderation, and keyword searches prove inadequate for analyzing social media content [11]. Research suggests that employing machine learning algorithms and transformer-based analysis yields higher accuracy in detecting bullying instances in social media data compared to relying solely on keyword searches and textual analysis [12, 13, 14]. However, it's worth noting that machine learning techniques proposed in literature tend to be tailored to specific types of content. Linguistic disparities between English and non-English content can lead to variations in performance. Moreover, factors such as socio-emotional behaviours and user-specific information within the studied population significantly influence cyberbullying detection outcomes. For example, while Support Vector Machines (SVM), a popular learning method for English text, exhibited lower accuracy when applied to Arabic texts compared to Naive Bayes (NB) [40]. Hence, this study aims to delve deeper into cyberbullying detection, particularly focusing on its application to Bangla text.

Therefore, the summarized form of motivations behind this study are appended below:

Corpus Development:

- Unavailability of standard corpora.
- Lack of automatic corpus development system.

Development of an Intelligent Framework for Identifying Cyberbullying Text:

- Systematically gathers a cyberbullying text corpus.
- Extracts features from the text.
- Ultimately builds the model for textual cyberbullying.

Implementation of the Transformer-based Language Models:

- Capture context-aware textual features during the text-to-feature extraction phase.
- Fine-tune the transformer-based language model using the cyberbullying corpus.

To Ensure Cyberspace Security. Describing the importance of ensuring cyberspace security through the identification of cyberbullying in Bengali linguistic texts using deep learning and Transformer-based approaches helps stakeholders understand the significance of this endeavour. Here's why it's crucial to articulate this importance descriptively:

- **Contextual Relevance.** Describing the importance provides context for why this specific approach matters in the broader context of cyberspace security. It helps stakeholders, including policymakers, researchers, and the general public, understand the relevance of addressing cyberbullying in Bengali linguistic texts.

- **Awareness Building.** Descriptive forms help raise awareness about the prevalence and impact of cyberbullying in Bengali-speaking communities. By highlighting real-world examples and statistics, stakeholders gain a deeper understanding of the problem's scope and severity.

- **Technological Solutions.** Explaining how deep learning and Transformer-based approaches can be leveraged to address cyberbullying demonstrates the potential of technology to tackle complex societal challenges. Descriptive forms help demystify these advanced techniques, making them more accessible to a broader audience.

- **Long-term Impact.** Descriptive forms can elucidate the potential long-term impact of effectively addressing cyberbullying in Bengali linguistic texts. By fostering healthier online interactions and promoting digital citizenship, these efforts contribute to building a more resilient and secure cyberspace for future generations.

Contributes to Military Cyber Security Domain. Describing the importance of contributing to military cyber security through the identification of cyberbullying in Bengali linguistic texts using deep learning and Transformer-based approaches is crucial for several reasons:

- **National Security Concerns.** Military cyber security is paramount for safeguarding a nation's critical infrastructure, sensitive information, and strategic assets. Cyberbullying, especially if targeted at military personnel or their families, can pose security risks by compromising morale, operational readiness, and personnel well-being.

- **Data Security and Integrity.** Leveraging deep learning and Transformer-based approaches to analyze Bengali linguistic texts for cyberbullying requires robust data security measures to protect sensitive information and ensure data integrity. This contributes to enhancing overall cyber hygiene and resilience within military cyber security domains.

1.5 Objectives

The objective of this work is to create a cyberbullying detection and monitoring system specifically designed for Bangla text on social media platforms. To accomplish this objective, the study has outlined the following goals:

- To develop an intelligent cyberbullying text identification model for low-resource language.
- To extract the context-aware text features and overcome the limitations of statistical, convolutional, and sequential cyberbullying text classification models.

1.6 Research Questions

In conclusion, to encapsulate the findings of the research, this study canters around the following Research Questions (RQs):

- **RQ1**: How to develop an intelligent cyberbullying text identification model for low-resource language?

- **RQ2:** How can extract the context-aware text features and overcome the limitations of statistical, convolutional, and sequential cyberbullying text classification models?

1.7 Challenges

Text identification involves discriminating one category from others within a given set. While classifiers typically perform well when prior knowledge of all categories is available, encountering an unknown class during testing can lead to poor performance, even with state-of-the-art classifiers. While most work in this field has been in computer vision, the few efforts in Natural Language Processing (NLP) show instability in performance and lack an open-world recognition framework. Test categorization faces numerous challenges, with Bengali text categorization posing particular difficulties due to a shortage of Bengali resources.

Commercial applications of text classification abound, with email spam filtering being one of the most common. The ability to automatically classify documents by content holds significant commercial value for corporate internet, government departments, and internet publishers. While many machine learning methods are discussed, exploiting domain-specific text features often yields greater performance gains than changing algorithms. However, understanding the data remains key to successful categorization, an area in which many categorization tool vendors are weak.

Hardware and software present dual challenges in text labelling. Bengali, being a low-resource language, lacks usable corpora for research. Overcoming this challenge involves collecting a large corpus from diverse Bengali language sources. Selecting appropriate categories and collecting data for each category is another challenge due to Bengali's highly inflected nature and rapidly evolving language usage. Semantic feature selection and extraction are crucial, with transformer-based models employed in the research framework for extracting semantic features.

Designing the text identifying layer is particularly challenging, as categorization accuracy heavily depends on layer design. Study addresses this challenge by employing deep learning and transformer-based model in the system, which helps overcome issues with low identification accuracy. Additionally, ensuring adequate hardware support is essential, as processing large volumes of Bengali text requires significant computational power. Thus, it has set up high-computation hardware to support Bengali cyberbullying text identification system.

Therefore, challenges related to RQ1 (How to develop an intelligent cyberbullying text identification model for low-resource language?) are as following:

- Unavailability of standard corpora (diverse linguistic contexts and cyberbullying instances).
- Lack of well annotated cyberbullying corpus.
- Usability issues.
- Shortage of contextual embedding model.

Consequently, challenges related to RQ2 (How can extract the context-aware text features and overcome the limitations of statistical, convolutional and sequential cyberbullying text classification models?) are as following:

- Non-contextual embedding is not able to semantic information.
- OOV issues are not managed by the non-contextual embedding i.e. word2vec, glove and FastText.

1.8 Contributions of the Work

The noteworthy contributions of this research and potential answers to the Research Questions (ARQs) are outlined as follows:

- This study developed an intelligent framework for identifying cyberbullying text. This framework systematically gathers a cyberbullying text corpus, extracts features from the text, and ultimately builds the model for textual cyberbullying identification.

- It implemented the transformer-based language models which capture context-aware textual features during the text-to-feature extraction phase and fine-tune the transformer-based language model using the cyberbullying corpus. The fine-tuned model overcomes the limitations of statistical, convolutional, and sequential models.

- Constructed a cyberbullying text identification corpus comprising 34,433 labeled texts. Within this corpus, 17,901 are categorized as “Bully”, and 16,521 are labeled as “Not-Bully”. The collection process involved manual gathering from social media, followed by annotation and verification tasks using a manual annotation approach.

- This study trained a total of 12 cyberbullying text identification models, employing a diverse range of methodologies. This includes three statistical models (SVM, Libsvm, SGD), four deep learning models (CNN, LSTM, VDCNN, GRU), and six transformer-based models (BanglaBERT, mBERT, DistilBERT, IndicBERT, XML-RoBERTa, bELEC TRA). Through empirical analysis, this study identified the top-performing model to detect cyberbullying texts.

1.9 Organization of the Thesis

This thesis report is organized into five chapters, which are outlined as follows:

Chapter 1: Introduction, problem statement, motivation, objectives, related work, and the organization of this project report.

Chapter 2: Background information on typical cyberbullying and machine learning algorithms.

Chapter 3: Development of the text identification system, including the data collection process from cyberspace and the proposed methodology for text identification model development. Several machine learning algorithms are identified and applied to the dataset, with the best performing model for Bangla text proposed for developing the cyberbullying detection system.

Chapter 4: Experimental results and discussions, showcasing the outcomes of the proposed system for cyberbullying detection and evaluating the performance of various parameters used in the detection process.

Chapter 5: Conclusion and future work of the thesis.

Chapter 2

Background and Related Work

2.1 An Overview

The continually evolving landscape of online platforms such as Twitter, Facebook, Reddit, and others has instigated extensive research into the identification and categorization of undesirable texts. This research spans diverse domains, including aggression classification, hate speech detection, abuse detection, toxicity classification, misogyny classification, trolling identification, cyberbullying detection, and offensive text classification. While a substantial body of research has focused predominantly on English, this paper provides a comprehensive overview of studies addressing violence, hate, and offensive text detection/classification in Bengali languages.

Online platforms have become pivotal arenas for communication, fostering both positive and negative interactions. The prevalence of undesirable texts such as hate speech, abuse, and cyberbullying necessitates robust mechanisms for their identification and classification. This thesis explores the current state of research in this area, with a particular focus on the advancements and challenges in detecting offensive texts in low-resource languages. Aggression classification has been extensively studied, with Kumar et al. presenting a dataset featuring covert and non-aggressive annotations in English and Hindi, encompassing 15,000 posts/comments [23]. Aroyehun et al. developed deep neural network-based English models using data augmentation and pseudo-labeling methods, achieving macro F1-scores of 0.64 and 0.59 with LSTM and CNN-LSTM approaches, respectively.

Another significant contribution comes from research utilizing the TRAC-2 dataset, where a bootstrap aggregating-based ensemble with fine-tuned BERT models achieved an 80.3% weighted F1 score in detecting violence and misogyny. Additionally, the Offensive Language Identification Dataset (OLID), compiled with 14,000 English tweets, implemented a three-layer hierarchical annotation schema to detect, classify, and identify text targets. CNN outperformed other models in baseline evaluations, achieving macro F1 values of 0.80, 0.69, and 0.47 at different classification levels [21].

Despite significant progress in high-resource languages, research on low-resource languages such as Bengali and Hindi remains limited. Existing studies often rely on non-contextual embedding models like GloVe, FastText, and Word2Vec, which struggle with Out-of-Vocabulary (OOV) issues and the extraction of local and global contextual features [26]. This gap highlights the need for more sophisticated approaches to address the unique challenges presented by low-resource languages. In response to these challenges, this study introduces the BullyFilterNeT system, which systematically develops a Bengali bully identification corpus. Various statistical, deep learning, and transformer-based language models are empirically evaluated to determine the most effective model for Bengali bully identification. This thesis contributes to the field by providing a detailed review of existing research on undesirable text detection across multiple languages and introduces a novel approach tailored to the specific requirements of Bengali bully identification. By addressing the challenges faced by low-resource languages, this study aims to enhance the efficacy of offensive text detection in diverse linguistic contexts.

2.2 Related Terminology

2.2.1 Language Model

Language models, such as those based on transformer architectures, are designed to process and generate human-like text by understanding the context and predicting subsequent tokens. These models leverage deep learning techniques and vast amounts of training data to achieve high levels of performance. Here, below are the technical description along with the key equations involved.

Transformer Architecture. The transformer architecture, introduced by Vaswani et al. (2017) [20], is foundational to modern large language models. It consists of an encoder-decoder structure, although many models, such as GPT (Generative Pretrained Transformer), use only the decoder part.

Encoder and Decoder Blocks. Each encoder and decoder block consist of:

- Multi-head self-attention mechanism.
- Feed-forward neural network.
- Layer normalization.

Self-Attention Mechanism. Self-attention allows the model to weigh the importance of different words in a sentence relative to a given word. The equations for the self-attention mechanism are as follows:

$$Q = XW_Q$$
$$K = XW_K$$
$$V = XW_V$$

Where Q, K, and V are the query, key, and value matrices, respectively. X is the input, and W_Q, W_K, and W_V are learned weight matrices. The scaled dot-product attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where d_k is the dimension of the key vectors.

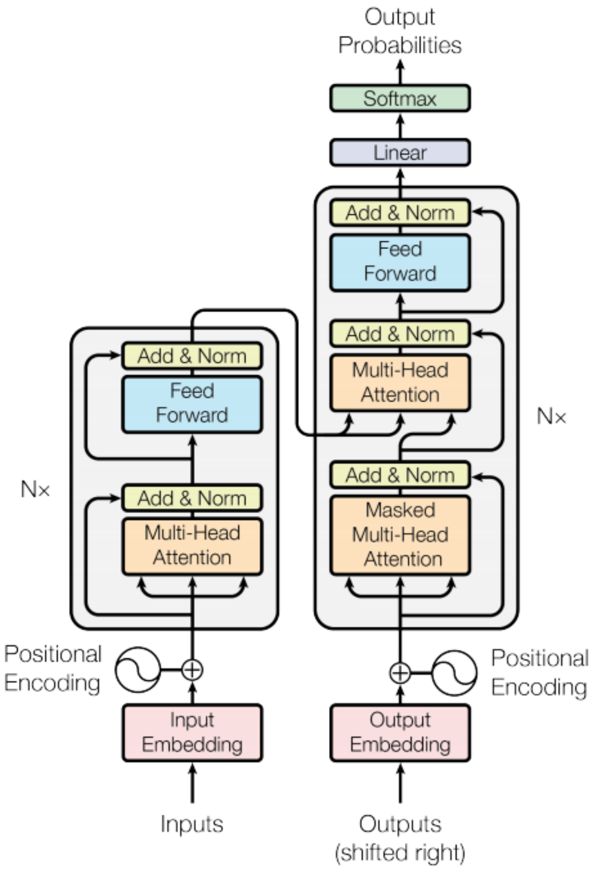


Fig: 2.1. Basic Structure of Large Language Model

Benefits of self-attention:

• Layer-wise minimizing computational complexity.

• Maximizing parallelizable computations measured by minimum number of sequential operations required.

• Layers minimize maximum path length between different input and output positions in network with several layer types. The shorter the path between any combination of positions in the input and output sequences, the easier to learn long-range dependencies.

Multi-Head Attention. Multi-head attention allows the model to jointly attend information from different representation subspaces at different positions.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W_O$$

Where each head is computed using the self-attention mechanism described above, and W_O is the learned output weight matrix.

Position-wise Feed-Forward Networks. After the multi-head attention mechanism, a position-wise feed-forward network is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Where W_1, W_2 , b_1, and b_2 are learned parameters.

Layer Normalization. Layer normalization is applied to the outputs of the attention and feed-forward sub-layers:

LayerNorm (x + Sublayer (x). Where Sublayer refers to either the multi-head attention or the feed-forward network, and \ (x \) is the input to the sublayer.

Training and Optimization. Training large language models involves optimizing the loss functions described above using gradient-based optimization methods like Adam. The training process requires significant computational resources, often involving distributed computing and specialized hardware such as GPUs and TPUs.

Tokenization. Before training, text data is tokenized into sub word units using techniques like Byte-Pair Encoding (BPE) or WordPiece. This allows the model to handle a large vocabulary and deal with out-of-vocabulary words.

2.2.2 Semantic Feature Extraction

Feature extraction is a process of key information retrieved from raw text. In document categorization, semantic feature represents the actual meaning of a document. Each document assigned a predefined class or category with respect to their semantic meaning [10]. Semantic feature extraction is a crucial step in text classification tasks, including the identification of cyberbullying texts. It involves transforming raw text data into a set of meaningful features that capture the underlying semantics. For Bengali cyberbullying text identification, this process can be complex due to the unique linguistic characteristics of the Bengali language. Below, it outlines the key steps and techniques used in semantic feature extraction.

Text Preprocessing. Before extracting semantic features, the text must be preprocessed to ensure consistency and to remove noise. This step typically involves:

- Tokenization: Splitting the text into individual words or tokens.
- Normalization: Converting text to lowercase, eliminating punctuation, and managing special characters.
- Stopword Removal: Removing common words that do not carry significant meaning, such as "এই", "ও", "কি", etc.
- Stemming/Lemmatization: Reducing words to their root form. However, stemming in Bengali can be challenging due to its complex morphology.

Word Embeddings. Word embeddings are dense vector representations of words that capture semantic meaning. Popular techniques include:

- Word2Vec: Trains word vectors using the context of words in a large corpus. It captures semantic relationships based on the proximity of words in sentences.
- GloVe (Global Vectors for Word Representation): Generates word embeddings by aggregating global word-word co-occurrence statistics from a corpus.
- FastText: Similar to Word2Vec but also considers sub word information, which is beneficial for morphologically rich languages like Bengali.

- Example: Word2Vec in Bengali is given as an example as appended below:

~~~
```python
from gensim.models import Word2Vec
import nltk
# Example Bengali sentences
sentences = [
   ["আমি", "স্কুল", "যাচ্ছি"],
   ["তুমি", "কি", "করছো"],
   ["সে", "ভালো", "ছাত্র"],
]
# Train Word2Vec model
model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)
# Get vector for a word
vector = model.wv['স্কুল']
print(vector)
```
~~~

Sentence Embeddings. While word embeddings capture the meaning of individual words, sentence embeddings represent entire sentences. These embeddings can be obtained using models like:

- BERT (Bidirectional Encoder Representations from Transformers): Provides contextualized word embeddings which can be pooled to create sentence embeddings.

~~~
Example: BERT for Bengali Sentence Embeddings
```python
from transformers import AutoTokenizer, AutoModel
import torch
# Load pre-trained Bengali BERT model
model_name = "bangla-bert-base"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)
# Example Bengali sentence
sentence = "আমি স্কুল যাচ্ছি"
# Tokenize and obtain embeddings
inputs = tokenizer(sentence, return_tensors="pt")
outputs = model(**inputs)
sentence_embedding = outputs.last_hidden_state.mean(dim=1).squeeze()

print(sentence_embedding)
```
~~~

2.2.3 Convolutional

In deep learning convolutions layers apply a convolution operation to the input transfer the output to the next layer. The convolution matches the output of an individual neuron to visual stimuli. Each convolutions neuron processes data only for its receptive field [10].

2.2.4 Pooling

In convolutions networks may include local or global pooling layers combine the outputs of neuron clusters at one layer into a single neuron in the next layer. In generally the max-pooling and average-pooling are used in deep learning area. The max-pooling cluster the maximum value from a fixed region and the average pooling cluster the average value [10].

2.2.5 Fully Connected

In deep learning, fully connected layers connect every neuron in one layer to every neuron in another layer. Fully connected layer design for the high dimensional feature mapped into flatten array. It is in principle the same as the traditional multi-layer perceptron neural network [10].

2.2.6 Batch Size. Batch size is a critical hyperparameter in deep learning that refers to the number of training examples utilized in one iteration of the model update. Choosing an appropriate batch size can significantly affect the performance and efficiency of the model training process. Below, it provides a detailed explanation of the concept of batch size, its implications, and how it can be applied in the context of Bengali cyberbullying text identification using deep learning techniques.

Understanding Batch Size. Batch size determines the number of samples that are propagated through the network at one time. The primary batch sizes are:

- Small Batch Size: Typically, between 1 and 32.
- Medium Batch Size: Generally, ranges from 32 to 128.
- Large Batch Size: Ranges from 128 to 512 or even higher.

Implications of Batch Size.

- Training Speed: Larger batch sizes can leverage the parallel processing capabilities of modern GPUs, leading to faster training. However, they require more memory.

- Model Convergence: Smaller batch sizes offer a noisier gradient estimate, which can help escape local minima and potentially lead to better generalization but might slow down the convergence.

- Stability: Large batch sizes provide a more stable and accurate estimate of the gradient, potentially leading to faster convergence but might get stuck in local minima.

Choosing Batch Size for Bengali Cyberbullying Text Identification. The choice of batch size can be influenced by several factors, including the size of the dataset, available computational resources, and the specific characteristics of the task. For Bengali cyberbullying text identification, the following considerations were crucial for this study:

- Dataset Size: If the dataset is small, smaller batch sizes might be more appropriate to ensure that the model can learn from more varied examples in each epoch.

- Resource Constraints: If the computational resources (like GPU memory) are limited, smaller batch sizes are necessary to avoid memory overflow.

- Model Complexity: For complex models like transformers, larger batch sizes might be more beneficial to fully utilize GPU capabilities and achieve faster training.

Practical Implementation. In practical implementation, the batch size can be adjusted dynamically to find the optimal setting. Here is an example using PyTorch for a transformer-based model:

~~~
```python
import torch
from torch.utils.data import DataLoader, TensorDataset

# Example dataset
texts = ["আমি স্কুল যাচ্ছি", "তুমি কি করছো", "সে ভালো ছাত্র"]  # Bengali sentences
labels = [0, 0, 1]  # Labels indicating cyberbullying (1) or not (0)

# Tokenization and encoding would typically be here
# For simplicity, let's assume texts_encoded is the encoded representation of texts
texts_encoded = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  # Dummy encoded data

# Convert to PyTorch tensors
inputs = torch.tensor(texts_encoded)
targets = torch.tensor(labels)

# Create TensorDataset and DataLoader
dataset = TensorDataset(inputs, targets)
batch_size = 32  # This can be adjusted based on experimentation and resource availability
dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=True)

# Example training loop
model = ...  # Initialize your model
optimizer = ...  # Initialize your optimizer
criterion = ...  # Initialize your loss function

for epoch in range(num_epochs):
    for batch in dataloader:
        inputs_batch, targets_batch = batch

        # Forward pass
        outputs = model(inputs_batch)
        loss = criterion(outputs, targets_batch)

        # Backward pass and optimization
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    print(f'Epoch {epoch+1}/{num_epochs}, Loss: {loss.item()}')
```
~~~

Hyperparameter Tuning. It's often necessary to perform hyperparameter tuning to find the optimal batch size. This can be done using techniques like grid search or random search. In grid search, various batch sizes are evaluated to determine the best performance.

2.2.7 Epoch. An epoch in deep learning refers to one complete pass of the entire training dataset through the neural network. It is a critical hyperparameter in the training process of machine learning models, particularly in the context of deep learning. Here is a detailed explanation of the concept of an epoch, its significance, and its application in identifying Bengali cyberbullying text using deep learning techniques.

Understanding Epoch. An epoch is a unit of time in deep learning training. During one epoch, every training sample in the dataset is seen by the model once. Training a model for multiple epochs means that the entire dataset is used to update the model’s weights multiple times.

Significance of Epoch.

- Model Training: More epochs allow the model to learn from the data iteratively, gradually improving its performance.

- Convergence: Properly setting the number of epochs ensures that the model converges to a point where it adequately learns the patterns in the data.

- Overfitting and Underfitting: Too few epochs can lead to underfitting, where the model fails to learn the underlying data patterns. Too many epochs can lead to overfitting, where the model learns the training data too well, including the noise, and performs poorly on new, unseen data.

Choosing the Number of Epochs for Bengali Cyberbullying Text Identification. Choosing the optimal number of epochs is crucial. This choice depends on the complexity of the model, the size of the dataset, and the specific task.

- Early Stopping: One common technique to find the optimal number of epochs is to use early stopping. This involves monitoring the model's performance on a validation set and stopping training when performance stops improving.

- Cross-Validation: This technique involves dividing the dataset into multiple folds and using different folds for training and validation to estimate the optimal number of epochs.

Practical Implementation. Here is an example of how the epochs were implemented in a training loop using PyTorch for a transformer-based model for Bengali cyberbullying text identification:

~~~
```python
import torch
from torch.utils.data import DataLoader, TensorDataset

# Example dataset
texts = ["আমি স্কুল যাচ্ছি", "তুমি কি করছো", "সে ভালো ছাত্র"]  # Bengali sentences
labels = [0, 0, 1]  # Labels indicating cyberbullying (1) or not (0)

# Tokenization and encoding would typically be here
# For simplicity, let's assume texts_encoded is the encoded representation of texts
texts_encoded = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  # Dummy encoded data

# Convert to PyTorch tensors
inputs = torch.tensor(texts_encoded)
targets = torch.tensor(labels)

# Create TensorDataset and DataLoader
dataset = TensorDataset(inputs, targets)
batch_size = 32
dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=True)

# Example training loop
model = ...  # Initialize your model
optimizer = ...  # Initialize your optimizer
criterion = ...  # Initialize your loss function
num_epochs = 20  # Number of epochs

for epoch in range(num_epochs):
    running_loss = 0.0
    for batch in dataloader:
        inputs_batch, targets_batch = batch

        # Forward pass
        outputs = model(inputs_batch)
        loss = criterion(outputs, targets_batch)

        # Backward pass and optimization
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        running_loss += loss.item()

    # Print epoch statistics
    print(f'Epoch {epoch+1}/{num_epochs}, Loss: {running_loss/len(dataloader)}')
```
~~~

Hyperparameter Tuning for Epochs. To determine the optimal number of epochs, used techniques like cross-validation or early stopping. It affects the model’s ability to learn patterns in the data and generalize to new, unseen data. By leveraging techniques such as early stopping and cross-validation, one can find the optimal number of epochs to balance between underfitting and overfitting, ensuring the best possible performance of the model.

2.2.8 Maximum Sequence Length.

Maximum sequence length is a hyperparameter used in natural language processing (NLP) models, especially in models that handle sequential data such as text. It defines the maximum number of tokens (words, sub words, characters) that a model can process in a single input sequence. Any input longer than this length is truncated, and shorter inputs are padded to this length.

Significance of Maximum Sequence Length.

- Memory Efficiency: Setting a maximum sequence length helps manage memory and computational efficiency since processing very long sequences can be resource-intensive.

- Model Performance: The chosen length can impact model performance. If it's too short, important contextual information might be lost due to truncation. If it's too long, it may introduce unnecessary noise and increase computational load.

Choosing the Maximum Sequence Length for Bengali Cyberbullying Text Identification.Choosing an appropriate maximum sequence length involves considering the nature of the dataset and the model's requirements:

- Dataset Analysis: Analyze the distribution of text lengths in your dataset to determine a length that captures most texts without excessive truncation.

- Model Constraints: Different models (e.g., RNNs, Transformers) have different capabilities and limitations regarding sequence length.

Practical Implementation. In transformer-based models like BERT or its variants, the maximum sequence length can be set during tokenization. Below is an example using the Hugging Face Transformers library to illustrate how to set and use maximum sequence length in a Bengali cyberbullying text identification task.

~~~
```python
from transformers import BertTokenizer, BertForSequenceClassification, Trainer, TrainingArguments
import torch

# Example Bengali texts
texts = ["আমি স্কুল যাচ্ছি", "তুমি কি করছো", "সে ভালো ছাত্র"]  # Bengali sentences
labels = [0, 0, 1]  # Labels indicating cyberbullying (1) or not (0)

# Initialize the tokenizer and model
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
model = BertForSequenceClassification.from_pretrained('bert-base-multilingual-cased')

# Define the maximum sequence length
max_length = 128

# Tokenize the texts
inputs = tokenizer(texts, padding=True, truncation=True, max_length=max_length, return_tensors='pt')

# Convert labels to tensor
labels = torch.tensor(labels)

# Example training loop
train_dataset = torch.utils.data.TensorDataset(inputs['input_ids'], inputs['attention_mask'], labels)
train_dataloader = torch.utils.data.DataLoader(train_dataset, batch_size=8, shuffle=True)

optimizer = torch.optim.AdamW(model.parameters(), lr=5e-5)

num_epochs = 3
for epoch in range(num_epochs):
    model.train()
    for batch in train_dataloader:
        input_ids, attention_mask, labels = batch
        optimizer.zero_grad()
        outputs = model(input_ids=input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss
        loss.backward()
        optimizer.step()
    print(f'Epoch {epoch+1}/{num_epochs}, Loss: {loss.item()}')
~~~

2.2.9 Weights

The neuron in a deep neural network computes an output value by applying some mathematical method to the input values which got from previous layer. The mathematical function that is applied to the input values is specified by a vector of weights and a bias. Learning in a neural network progress by making incremental adjustments to the biases and weights. The vector of weights and the bias are called a filter and represents some feature of the input [10].

2.2.10 Training Phase

In the field of machine learning and deep learning the training phase is an information retrieval session where the model is generating from raw data source. In deep learning the training phase consume huge time for model generation purpose.

2.2.11 Development Phase

The model beauty or accuracy of training phase measurement can be doing in the development phase. Training data set and development data sets are totally different. Development data set only use for model performance evaluation. Model overfitting and underfitting also calculate from the training and development phase error. The development phase also suggests that, which distribution should be added to the training data set.

2.2.12 Testing Phase

In deep learning, the testing phase is performance measurement phase of proposed algorithm. The testing phase data is totally different from the development and training data set. The overall accuracy of an algorithm calculates from the testing phase.

2.2.13 Feature Extraction and Classification Algorithm

Bengali document categorization system depends on two steps, the first step is the text to feature extraction and second step is the classifier model generation for classification purpose.

2.2.14 Term Frequency-Inverse Document Frequency (TFIDF) based Feature Extraction Algorithm

In document categorization system the first step is the feature extraction. there are many ways to extract the feature from raw text, TF-IDF is one of them. The TF-IDF is a statistical algorithm [3]. The Algorithm working technique describe bellow:

- Count the word frequency $f_{t,d}$.
- Calculate the term-frequency using eq.2.1

$$tf\ (d,t) = 0.5 + \frac{f_{t,d}}{\max(f_{t,d} \in d)} \qquad (2.1)$$

  Here t represents the term and d represent the document.

- Calculate the inverse document frequency using eq.**??**.'

$$idf\ (d,t) = log + \frac{N}{\mid D \in d, t \in d \mid} \qquad (2.2)$$

  Here t represent the total number of document and D represent the individual document length.

- Calculate the *tfidf*(*d,t*) = *tf*(*d,t*) × *idf*(*d,t*).

  Now for each word calculate the tfidf value. The tfidf is the statistical feature extraction method.

2.2.15 Word2Vec based Feature Extraction Algorithm

Word2Vec [5] is an unsupervised feature extraction algorithm which extract the semantic feature for each word. In this algorithm input take as raw text data and output is embedding model. The embedding model row represent the number of words and column represent the feature dimension. The detail of Word2Vec algorithm describe here:

- Calculate a unique word for all given documents.
- Calculate the word frequency for each word.
- Parameters initialize by random value.
- Apply Bayesian rule to calculate the semantic expectation value.
- For each word initialize the feature value using random value.
- Feed to the network and prepare the embedding model.

2.2.16 Support Vector Machine (SVM) based document classification Algorithm

Support Vector Machine is a traditional machine learning algorithm [2] which used for the different type of classification purpose. Algorithm working technique given bellow:

- Each word represents by numerical feature value.
- Weight vector initialize by the random value.
- Different projection plane applied to the data distribution.
- Calculate the loss value and try to minimize the loss value.
- Calculate the margin value.
- Select the minimum loss vale plane. Now the final weight value is the SVM model which is used during the text categorization.

2.2.17 Deep Learning-based Text Classification Algorithm

In the recent year the deep learning-based algorithm is most usable and update algorithm for document categorization. Document categorization purpose deep learning algorithm vary from layer design to layer design. Here we describe a key steps of deep learning algorithm for document categorization purpose.

- Text to feature extract from feature extraction module.
- Input feed by a 2D feature matrix.
- Input propagates through the layer architecture.
- Soft-max classifier applied to the output layer.
- Backpropagation applied for error readjustment.
- Minimize the error value.
- Generate model for classification purpose.

2.2.18 Fine-tuning

Fine-tuning a pre-trained language model for the specific task of identifying Bengali cyberbullying text involves several key steps and technical considerations. The following description outlines the process, including dataset preparation, model adaptation, and training.

Pre-trained Language Model Selection. Choose a pre-trained language model that supports Bengali, such as multilingual models like mBERT (Multilingual BERT), XLM-R (Cross-lingual RoBERTa), or a specifically pre-trained Bengali model like BanglaBERT. These models have already learned general language representations from large-scale multilingual corpora.

Model Adaptation.

- Input formatting.
- Prepare the inputs to the model.
- Input IDs: Tokenized text converted into input IDs.
- Attention Masks: Binary masks indicating which tokens should be attended to.
- Labels: Binary or categorical labels indicating the presence or absence of cyberbullying.

Fine-Tuning Procedure.

- Hyperparameters: Set appropriate hyperparameters such as learning rate, batch size, number of epochs, and optimizer. Commonly used optimizers include AdamW.

- Training Loop: Implement the training loop where the model is fine-tuned on the specific task.

- Load Pre-trained Model: Load the pre-trained language model with its corresponding tokenizer.

- Add Classification Layer: Add a task-specific classification layer on top of the pre-trained model. This is typically a linear layer mapping the model's hidden states to the desired number of output classes.

- Model Evaluation. Evaluate the fine-tuned model on a held-out test set using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to ensure robust performance.

- Error Analysis. Analyze misclassified instances to understand common failure modes and improve data pre-processing or model architecture if needed.

Example Code Outline

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer, TrainingArguments
import torch
from datasets import load_dataset

# Load pre-trained model and tokenizer
model_name = "bert-base-multilingual-cased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=2)

# Prepare dataset
dataset = load_dataset('path_to_bengali_cyberbullying_dataset')
def preprocess_data(examples):
    return tokenizer(examples['text'], truncation=True, padding='max_length', max_length=128)

encoded_dataset = dataset.map(preprocess_data, batched=True)

# Training arguments
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,
    weight_decay=0.01,
)

# Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=encoded_dataset['train'],
    eval_dataset=encoded_dataset['validation'],
    tokenizer=tokenizer,
)

# Fine-tune the model
trainer.train()
```

2.3 Literature Reviewed

While numerous studies have explored the efficacy of machine learning-based models in early stages, deep learning and transformer based has often been overlooked for detecting cyberbullying of low resource language. However, a few studies have begun to utilize supervised and semi-supervised learning approaches. Semi-supervised methods leverage classifiers constructed from a training corpus comprising a small set of labelled instances and a larger set of unlabelled ones [13]. These methods are particularly useful for addressing data sparsity, a common challenge in cyberbullying research where data may be scarce or dispersed. The automatic detection of cyberbullying is typically framed as a binary classification task, involving the differentiation between bullying and non-bullying posts. In this context, binary classifiers produce outputs categorized into a positive class (representing instances containing textual cyberbullying) and a negative class (comprising instances lacking bullying signals) [19].

The availability of suitable data is a fundamental challenge in developing effective models for detecting cyberbullying. Over the past few years, only a handful of datasets have been made publicly accessible for this specific purpose. These include training sets provided within the framework of the CAW 2.0 workshop, a cyberbullying corpus from MySpace and formspring annotated with Mechanical Turk, and more recently, the Twitter Bullying Traces dataset [43]. Given the scarcity of available data, many studies have had to construct their own corpora from social media platforms known for hosting cyberbullying content, such as YouTube, Twitter, Instagram, and Facebook. Notably, due to the widespread use of Facebook and Twitter globally, several studies have developed their own corpora based on these platforms.

Despite the challenge posed by limited data availability, in recent years, numerous successful research efforts have been conducted in automatic text analysis and cyberbullying detection. These studies have explored various features for predictive power, including n-grams (with and without tf-idf weighting), part-of-speech information (e.g., first and second pronouns), and sentiment information based on lexicons (polarity and profanity) [45]. Despite their apparent simplicity, content-based features, encompassing lexical, syntactic, and sentiment information, are frequently leveraged in recent cyberbullying detection approaches. More than 41 papers have tackled cyberbullying detection using content-based features, underscoring the critical role of such information in this task. Thus, within this research, content-based features are duly considered [45].

Despite the significance of user-related information and their activities, such as the number of posts on social networks, age, gender, location, and the number of friends and followers, few studies have thoroughly examined these features. Additionally, network-based attributes are crucial for identifying cyberbullying. Some studies have assessed the power imbalance between the bully and the victim, as well as the bully's popularity by analyzing interaction graphs and the bully's network position. Recently, research has focused on detecting cyberbullying using multi-modal data from specific platforms [33]. For instance, studies on Instagram have combined textual features from posts with user metadata, demonstrating that this combination improves classification performance.

A significant amount of research has been conducted in the areas of text categorization and cyberbullying detection within the English language. Some researchers have utilized text mining to classify posts and conversations. For instance, Yin, Xue, and Hong employed supervised learning for text classification, using N-grams for labelling and TF-IDF for weighting [31]. Dinakar, Reichart, and Lieberman conducted comparative research using various supervised approaches [34]. They collected and manually labelled YouTube comments, then implemented various binary and multiclass classifications. Kelly Reynolds applied decision trees [42] and k-nearest neighbors (KNN) in her studies. Support Vector Machines (SVM) have garnered considerable attention due to their superior performance in text classification tasks. One study examined the theoretical application of SVM in text classification, while another by Zhijie et al. compared SVM with Naive Bayes (NB) and KNN, finding that SVM outperformed both classifiers. However, all these studies were conducted exclusively on English text [36].

Due to linguistic differences between English and non-English content, the performance and accuracy of algorithms vary when applied to non-English text. Research indicates that the Naive Bayes Classifier can be effectively used for classifying Indian text. A study combining Naive Bayes with Ontology-Based Classification demonstrated better performance for Punjabi text [38]. Previous research has shown that Support Vector Machines (SVM) yield better classification results than Naive Bayes for Urdu. Additionally, Artificial Neural Network models perform better than the Vector Space Model for Tamil content. Techniques such as Decision Trees, Neural Networks, and N-grams have also proven effective in text classification for non-English content [46]. However, very few studies have focused on Bangla text. This research aims to evaluate the performance of various machine learning algorithms on Bangla text, identifying those with the highest accuracy for non-English languages. Furthermore, it will analyze the impact of user-specific data on the detection of cyberbullying in Bangla social media content.

In cyberbullying detection research, datasets are often created using keyword searches, leading to a biased collection of positive (i.e., bullying) instances. To balance these datasets, data resampling techniques are used, incorporating negative data from a background corpus. To avoid such biases in data collection, this research employed a different approach by randomly crawling Twitter and Facebook data, entirely bypassing the keyword search technique. Instead, all instances were manually annotated for the presence of bullying, resulting in a corpus that realistically represents the distribution of bullying instances.

It is important to note that the performance of automatic cyberbullying detection algorithms varies considerably across different studies. These performance scores are influenced not only by the chosen algorithm and its parameter settings but also by several other factors. These factors include the evaluation metrics used, such as F1 score, precision, recall, and AUC. Additionally, the type of corpus (e.g., Facebook, Twitter, ASKfm, Instagram), class distribution (whether balanced or unbalanced), and the annotation method (whether annotations are automatic, crowdsourced, or done by experts) also play significant roles in determining the effectiveness of these detection systems [35].

Detecting bullying-related text from the web is an evolving and trending research topic in NLP. While extensive research has been conducted on the textual analysis of bullying in English, relatively few studies have focused on the Bangali language (Müller et al., 2023) [34]. Previous research in this area can be broadly categorized into two main fields: (i) bully-related sentiment and emotion analysis, and (ii) fake news detection. Recently, numerous studies have focused on sentiment and emotion analysis using bullying-related texts. Liu and Liu (2021) developed a public sentiment analysis system to assess the effects of cyber bullying using English tweets. They collected 2,678,372 cyber bullying-related tweets in English, with 1,971,342 (73.6%) containing user locations. Of these tweets, 1,146,866 (42.8%) expressed positive sentiment, 720,737 (26.9%) were neutral, and 810,769 (30.3%) were negative. Another study by Malla and P.J.A. (2021) introduced the MVEDL method to identify English COVID-19-related tweets, achieving a maximum accuracy of 91.75% on the test dataset. Kabir and Madria (2021) developed an emotion classification system for English tweets using machine learning, achieving 89.51% accuracy and a 64.75% Jaccard score. However, this study was limited to only two machine learning models (BiLSTM and XML-RoBERTa) for classifying emotions and did not account for data-level uncertainty and the impact of late fusion.

Theocharopoulos et al. (2022) developed a COVID-19-related content analysis system using the BERT model to analyze Twitter tweets, achieving a maximum accuracy of 99.00% in distinguishing between positive and negative sentiments. Another system, COVID-Twitter-BERT (CT-BERT), created by Müller et al. (2023), focused on vaccine sentiment and stance analysis related to COVID-19, achieving maximum accuracies of 86.90% for vaccine sentiment and 74.80% for vaccine stance detection. However, this system primarily targets Twitter data and lacks clear research directions for low-resource languages.

Seilsepour et al. (2023) introduced a hybrid CNN-GRU model for fake news related tweet sentiment analysis, which incorporates a topic model for sentiment identification and achieved a maximum accuracy of 86.70% across 70 topics. Hall et al. (2022) also developed a Twitter-based sentiment analysis system for fake news related texts, employing transformer-based language models.

Given the widespread issue of abusive language online, detecting such language and finding effective solutions has become a significant research focus. While much work has been done in English, there is growing interest in detecting abusive language in Bangla. Researchers are developing new methods to tackle this problem. Feature extraction is crucial for processing data and significantly impacts classifier performance. For example, N-gram features extracted using a TF-IDF vectorizer were employed in a study where a GRU-based deep learning model categorized 5,126 Bengali comments into six categories, achieving a 70% accuracy rate. Another study created a dataset of 30,000 comments, annotated by 50 annotators three times. They used Word2Vec, FastText, and BengFastText as word embedding models and tested SVM, LSTM, and BiLSTM classifiers. The SVM outperformed the others with an accuracy of 87.5%.

Limitations of Existing Works

Building on existing research statements, this study provides a summarized overview of the most significant works, emphasizing their limitations and highlights. The details are presented in Table 1. Drawing insights from the highlights and limitations of the most relevant existing research, this study investigates the distinctions between the proposed BullyFilterNeT system and previous works. The following section will expound on these differences in detail.

Table 2.1: Deductions from Reviewed Literature

Model	Highlights	Limitations
VADER (Valence Aware Dictionary and Sentiment Reasoner) (Liu and Liu, 2021)	Task: Text analysis, A total of 2,678,372 English Tweets, Positive 42.8%, Neural 26.9%, Negative 30.3%	Non-context aware, only for English text, failed to process high dimensional features
XML-RoBERTa (Kabir and Madria, 2021)	Task: Speech classification, Maximum accuracy of 89.51%, overcome OOV issues	Overlooks the hyperparameters tuning and limited for low-resource languages
BERT (Theocharopoulos et al., 2022)	Task: Sentiment analysis, Maximum accuracy 99.00% for Twitter Tweets, Extract context-aware features	Only consider the positive and negative sentiment
CT-BERT (Müller et al., 2023)	Task: Sentiment & Stance detection, Maximum accuracy 86.90% for vaccine sentiment & 74.80% for vaccine stance	Only consider BERT-based models, Not used fusion-based models
CNN-GRU (Seilsepour et al., 2023)	Task: Sentiment analysis, Topics-based model, Maximum accuracy of 86.70%	Non-contextual features, OOV word issues, Not tuned hyperparameters
BERT+CNN (Alghamdi et al., 2023)	Task: Fake news detection, Maximum F1 score of 98.00%, Model hybridization	Domain dependent dataset (only consider Twitter Tweets)
AraBERT (Ameur and Aliane, 2021)	Tasks: Fake news & hate speech detection for Arabic, Domain-specific Transformer model, Overcome OOV word issues, Maximum accuracy of 98.58%	Not tuned the hyperparameters & Not fused with multilingual transformer score
Ensemble (Mohammed and Kora, 2021)	Task: Fake news detection, Maximum accuracy of 96.31%	Non-contextual features, Statistical Classifiers, unable to handle high dimensional features

In the realm of text analysis, the current body of research primarily concentrates on high-resource English texts, exploring facets such as misinformation, disinformation, fake news, sentiment analysis, and emotion detection. However, research efforts in the specific domain of Bangla bullying-related texts have been relatively scarce, particularly in cyber bullying detection. While some studies have examined hate speech or bullying texts in both high-resource and low-resource languages like English and Arabic, they have not tackled challenges such as out-of-vocabulary (OOV) words, adapting hyperparameters for new domains, and assessing the impact of monolingual and multilingual transformer-based language models on Bengali bullying text identification.

Bangla, being a morphologically diverse language, still confronts significant challenges, including a shortage of corpora, tuned hyperparameters, and domain-specific language models. Recognizing these limitations, this research diverges from existing studies in three key aspects:

(i) This study proposes a systematic framework for corpus development tailored to Bengali text identification. It has created corpora relating to Bangla Embedding Corpus, and Bullying Text Identification Corpus, marking the endeavour to develop such resources in the Bangla text domain.

(ii) It has devised a hyperparameter tuning algorithm explicitly designed for fine-tuning transformer-based language models in a new domain, specifically the Bengali cyber bullying text identification system.

(iii) It introduces a multilingual transformer-based models to enhance the detection accuracy of the BullyFilterNeT system.

Research Gap and Area Chosen for Research

These distinctions facilitate a focused analysis of Bengali bullying text, harnessing a dedicated language model and innovative methodologies to bolster performance and address limitations present in existing research. Bearing the foregone problem statement and considering the significance of the problem and scope, the following research gaps have been set for the research:

- Previous research predominantly focused on high resource languages.
- Limited attention has been given to low-resource languages like Bengali.
- Research in low-resource bully identification often relies on non-contextual features and Out-of-Vocabulary (OOV) issues.
- In response to these challenges, this study introduces the Cyberbullying Text Identification System for Bengali which incorporate lack of cyberbullying textual resource issues, OOV issues and adopted hyperparameter for domain specific tasks.

2.4 Related Work

The document categorization system depends on two steps. The first step is the feature extraction from raw text which is known as word embedding and second step is to design a document classifier model preparation.

2.4.1 Feature Extraction using Word Embedding

There are many valuable researches conducted on word embedding. In linguistics word embeddings are discussed in the research area of distributional semantics. It aims to quantify and categorize semantic similarities between linguistic items based on their distributional properties in large samples of language data. Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension. Word embeddings come in two different styles, one in which words are expressed as vectors of co-occurring words, and another in which words are expressed as vectors of linguistic contexts in which the words occur. Word2vec is a group of related models that are used to produce word embedding. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the word's probability of co-occurrence. Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associate (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information gets encoded as vector differences as well. For this reason, the resulting word vectors perform very well on word analogy tasks, such as those examined in the word2vec package. To deal with co-occurrences that happen rarely or never which are noisy and carry less information than the more frequent ones the authors use a weighted least squares regression model. Training is performed on aggregated global word-word co-occurrence statistics from a corpus and the resulting representations showcase interesting linear substructures of the word vector space. In the recent year the Word2Vec [3, 11, 12] and Glove [13] algorithms achieved the state-of-the-art word embedding result for English, French, Arabic and Turkish languages. Word2Vec is imposing in Bengali word embedding purpose [3] and archived a good result. One of the main limitations of word embeddings is that possible meanings of a word are conflated into a single representation. Sense embeddings are a solution to this problem: individual meanings of words are represented as distinct vectors in the space. Glove also has a limitation like as Poor performance on word analogy task and frequent words contribute disproportionately high to the similarity measure.

2.4.2 Text Identification based on Traditional Machine Learning Techniques

The traditional machining learning is machine learning technique where the data volume size is smaller. In information retrieval, TF-IDF or short for term frequency Inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequent in general. TF-IDF is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use TF-IDF. Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. TF-IDF can be successfully used for stop-words filtering in various subject fields, including text summarization and classification. One of the simplest ranking functions is computed by summing the TF-IDF for each query term; many more sophisticated ranking functions are variants of this simple model. There are very few researches have been conducted on Bengali text document classification. In information retrieval theory term frequency (TF) and inverse document frequency (IDF), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The TF-IDF based features are the tradition technique which only depends on documents term frequency or BoW model [48, 49, 50]. Lexical feature only carries the limited amount of information such as avg. sentence length, avg. number of words length, number of different word and so on [4]. TF-IDF feature performs lower accuracy due to absence of semantic information. The TF-IDF feature not contained the word position and correlation of the word. The lexical and TF-IDF feature are not working properly for Bengali language due to its large inflectional diversity in verbs, tense, noun, etc. In our work, we use DCNNs for Bengali documents categorization. This approach showed better performance than the previous Bengali text classification methods due to the hyper parameter tuning and deep network training architecture. The decision tree and SVM [48] based technique archived the better result in the ancient time where text is categorized based on traditional feature learning technique.

2.4.3 Text Identification based on Deep Learning Techniques

Deep learning is a subset of machine learning in Artificial Intelligence (AI) that has networks capably of learning unsupervised from data that is unstructured or unlabelled. Also known as Deep Neural Learning or Deep Neural Network. In machine learning, a convolutional neural network is a class of deep, feed-forward artificial neural networks, most commonly applied to visual imagery. CNNs use a variation of multilayer perceptron designed to require minimally pre-processing. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behaviour for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. The term" recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. In the recent year Convolution Neural Net (CNN) and Recurrent Neural Nets (RNN) based document classifier model work for English and other language. The CNN and RNN based documents classifier approaches achieved 84.00% and 85.60% accuracy from English text [51]. Conneau et. Al. [52] introduced very deep CNN and achieved 96-98% accuracy from different English data set classification.

Hierarchical Deep Learning is the machine learning task with the hierarchical data. This model has been used for text classification, as in Hierarchical Deep Learning employs stacks of deep learning architectures to provide specialized understanding at each level of the data hierarchy. A traditional multi-class classification technique can work well for a limited number class, but performance drops with increasing number of categories or classes, as is present in hierarchically organized documents. Many techniques work on

Hierarchical Attention for text classification [3], or Multi model Deep Learning for classification task. In hierarchical CNN (HCNN) based and decision tree-based CNN also have been achieved higher accuracy for English text [38, 52].

Stochastic Gradient Descent (SGD) is a simple approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning. Stochastic gradient descent (SGD) based classifier perform lower accuracy due to feature scaling and lack of huge hyper parameters tuning [2]. Character-level Convolutional Networks for text classification is another approach which embedding tasks done by the character level. The Character-level CNN is performing slower embedding system and memory consuming [51].

2.4.4 Transformer-based Approaches

Transformer models, particularly BERT (Bidirectional Encoder Representations from Transformers) and its multilingual variants, have set new benchmarks in NLP tasks due to their ability to capture contextual information bidirectionally. Hasan et al. (2021) introduced a Bangla BERT model fine-tuned for cyberbullying detection. The model outperformed traditional and deep learning approaches by leveraging pre-trained language representations and fine-tuning on a specific dataset. Transformer models, introduced by Vaswani et al. (2017), rely on self-attention mechanisms to process and encode text. Unlike traditional RNNs, transformers can handle long-range dependencies in text more effectively, making them suitable for tasks requiring contextual understanding. BERT (Bidirectional Encoder Representations from Transformers): BERT is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. This allows it to understand the context of a word in a sentence better than unidirectional models.

Sarker et al. (2021) proposed a transformer-based approach using mBERT (multilingual BERT) to detect cyberbullying in Bangla and Hindi texts. Their study demonstrated the effectiveness of transformer models in handling low-resource languages by utilizing multilingual training data.

Optimizing hyperparameters specifically for the task of cyberbullying detection can lead to better performance.

- Systematic Tuning: Developing algorithms for fine-tuning hyperparameters such as learning rate, batch size, and maximum sequence length to adapt the model for Bangla text specifics.

- Domain-Specific Adjustments: Making adjustments based on the peculiarities of Bangla text, such as handling complex morphology and syntax.

In this work, using BanglaBERT for Bengali text identification. This approach showed better performance than the previous Bengali text identification methods due to the hyper parameter tuning and transformer-based architecture.

2.4.5 BanglaBERT

It is a transformer-based language model specifically pre-trained for the Bengali language. Leveraging the architecture of BERT (Bidirectional Encoder Representations from Transformers), BanglaBERT inherits several key technical attributes:

Architecture.

- **Layers:** Comprises multiple transformer layers (typically 12 or 24), each consisting of self-attention mechanisms and feed-forward networks.

- Attention Heads: Utilizes multi-head self-attention to capture different aspects of the input data, enabling the model to focus on various parts of a sentence simultaneously.

Pre-training.

- **Corpus:** Pre-trained on a large Bengali corpus, encompassing diverse text sources to capture the linguistic nuances of Bengali.

- **Fine-tuning – Task-Specific Adaptation:** Fine-tuned on a labelled cyberbullying text dataset, where the model learns to classify texts as either cyberbullying or non-cyberbullying.

- **Hyperparameter Tuning:** Hyperparameters such as learning rate, batch size, maximum sequence length, and the number of training epochs are optimized to improve performance.

Significance of BanglaBERT in Identifying Cyberbullying Bengali Text.

- **Linguistic Nuances:** BanglaBERT is tailored to the Bengali language, making it adept at understanding specific linguistic patterns prevalent in Bengali texts. This specificity is crucial for accurately identifying subtle forms of cyberbullying that might be missed by models trained on other languages.

- **Contextual Understanding:** By leveraging the transformer architecture, BanglaBERT captures deep contextual features, allowing it to understand the context in which words are used. This is essential for identifying cyberbullying, which often relies on context rather than explicit offensive terms.

- **Handling Out-of-Vocabulary (OOV) Issues:** BanglaBERT uses sub word tokenization (WordPiece), which helps in mitigating OOV issues. This means the model can handle rare and unseen words by breaking them down into sub words, ensuring better understanding and representation of diverse text inputs.

- **Resource Efficiency:** Leveraging a large pre-trained model reduces the need for extensive datasets, which are often scarce for low-resource languages like Bengali. This makes BanglaBERT a practical and efficient solution for developing NLP applications in Bengali.

Chapter 3

Cyberbullying Text Identification Models Development

3.1 Methodology

The primary objective of this study is to develop an intelligent Bengali cyberbullying text identification system called BullyFilterNeT which can intelligently distinguish between pieces of text as either containing bullying content or not. To fulfil this objective, the research progresses through three steps: (i) Cyberbullying Corpus Development (ii) Cyberbullying Text Identification Models Development (iii) Cyberbullying Text Identification Models Verification and Selection. The abstract view is presented in Figure 3.1.

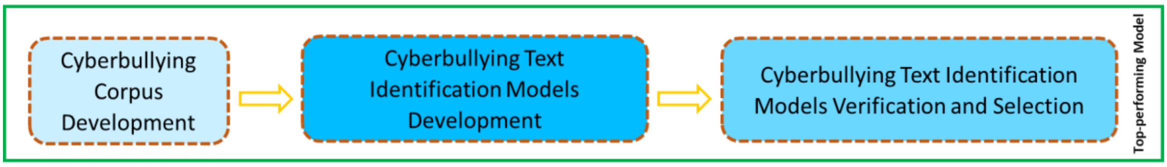

Cyberbullying Corpus Development
Cyberbullying Text Identification Models Development
Cyberbullying Text Identification Models Verification and Selection
Top-performing Model

Fig 3.1. Abstract View of BullyFilterNeT

3.2 Task Abstract

The growing popularity of social media has brought the Unicode system to the forefront, enabling the use of diverse languages and scripts online. Despite this advancement, the identification of cyberbullying in Bengali texts remains underexplored, primarily due to the scarcity of annotated corpora related to bullying in low-resource languages like Bangla. This research aims to address this gap by developing a robust system leveraging deep learning and transformer-based approaches for the detection of cyberbullying in Bangla linguistic texts. This study begins with the creation of a comprehensive Bangla cyberbullying dataset sourced from various social media platforms, manually annotated to ensure precise text identification. Then fine-tune pre-trained multilingual transformer models, specifically BanglaBERT, to adapt to the Bangla language, incorporating advanced techniques such as hyperparameter tuning methods to enhance model performance.

In the realm of cyberbullying detection within Bangla linguistic texts, intelligent text classification plays a pivotal role. This process involves three main stages: Input Transformation, Feature Extraction, and Classification, ultimately producing category-wise class outputs.

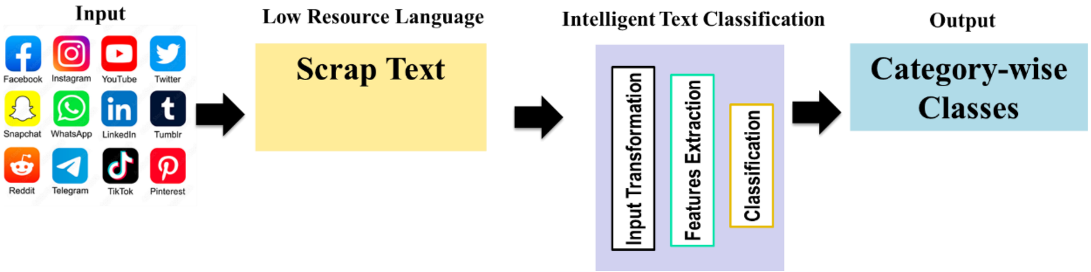


Fig. 3.2. Task Abstract for Intelligent Text Classification for Bangla Cyberbullying Detection

3.2.1 Input Transformation. Input transformation is the first critical step in preparing raw text data for analysis and classification. For Bangla texts, this involves several key processes:

- Normalization: Converting text to a consistent format, such as lowercasing and standardizing Unicode characters.
- Tokenization: Splitting the text into individual words or tokens. Bangla tokenization requires handling compound words and complex scripts.

3.2.2 Feature Extraction. Feature extraction involves transforming the textual data into a numerical representation that machine learning models can process. For Bangla cyberbullying detection, this involves several sophisticated techniques:

- Word Embeddings: Using models like Word2Vec, FastText, or transformer-based embeddings such as mBERT to convert words into dense vector representations that capture semantic meaning.

- Contextual Embeddings: Utilizing transformer models like BERT to capture context-dependent meanings of words in sentences, providing richer feature representations.

3.2.3 Classification. The classification stage involves applying machine learning algorithms to the extracted features to categorize the text into predefined classes. For cyberbullying detection in Bangla, several models were employed in trial and error basis:

- **Traditional Machine Learning Models:** Algorithms like Support Vector Machines (SVM), Logistic Regression (LR), and Random Forest (RF) have been used with varying success. However, they often struggle with the complexity of natural language.

- **Deep Learning Models:** Models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks are adept at capturing sequential dependencies and patterns in text data.

- **Transformer-based Models:** State-of-the-art models like BERT, mBERT, and their variants are particularly powerful due to their ability to understand context and manage large-scale text data effectively.

3.2.4 Category-wise Classes Output. The final output of the classification process is a set of category-wise classes that denote the type of content detected in the Bangla text. For cyberbullying detection, these categories were included:

- **Non-bullying:** Texts that do not contain any form of abusive or harmful language.
- **Bullying:** General category for texts identified as containing bullying language.

3.2.4 Task List. The detail task list as per the above discussion is shown in the Figure 3.3.

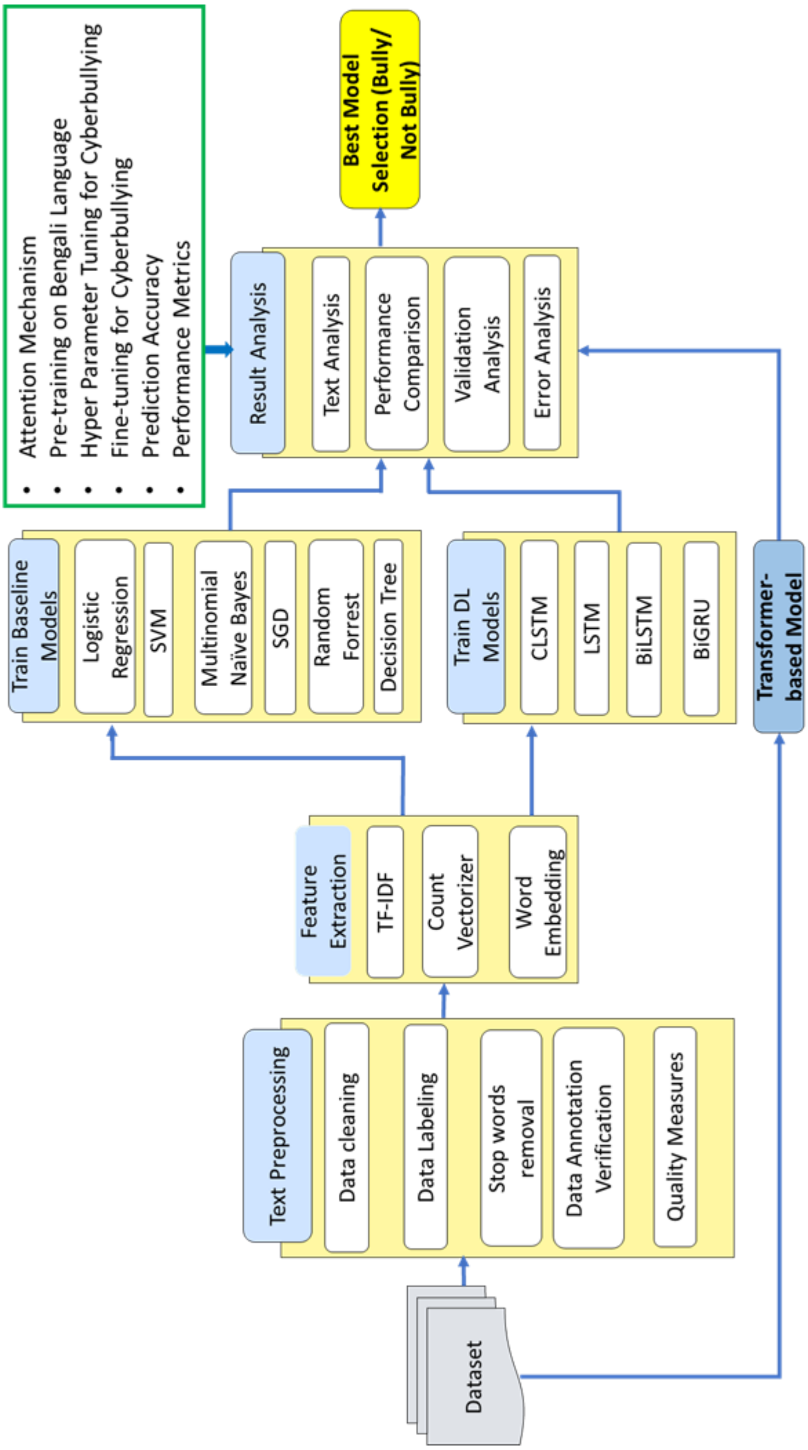


Fig. 3.3. Task List for Intelligent Text Classification for Bangla Cyberbullying Detection

3.3 Cyberbullying Corpus Development

3.3.1 In this research, collected bully and not bully-related texts corpus were built in six steps. The overall procedure is presented in Figure 3.3. Each of the steps is described in the following paragraphs.

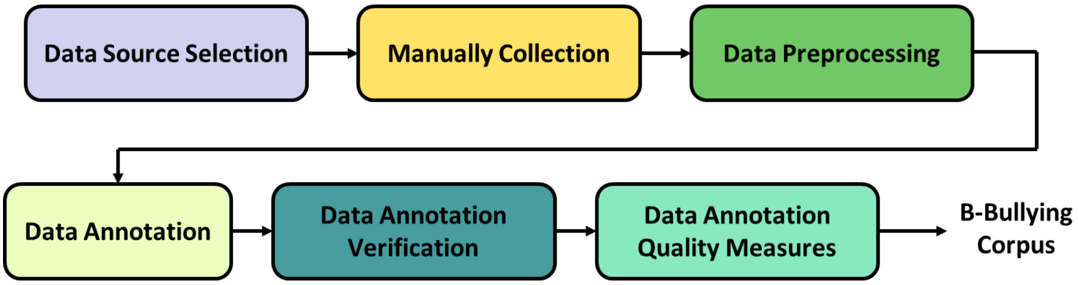


Fig. 3.4. Abstract View of Cyberbully Corpus Development

3.3.2 Data Source Selection. In the realm of data source selection, particularly for social media and blogs, the choice of platforms plays a pivotal role in shaping the nature and quality of the data acquired. Social media platforms like Twitter, Facebook, and Instagram provide real time and diverse user-generated content, making them valuable sources for understanding public opinions, trends, and sentiments. The informal and conversational nature of social media content can offer insights into current events and user interactions. Similarly, blogs, with their more extended and often reflective narratives, contribute to a deeper understanding of individual perspectives and experiences. However, the selection process should consider the specific objectives of the research, the target audience, and the potential biases inherent in each platform. Striking a balance between the immediacy of social media and the depth of blogs is essential for obtaining a comprehensive and representative dataset for various analyses and applications.

3.3.3 Manual Collected. Manual data collection from social media and blogs involves a meticulous and hands on approach to gathering information directly from these online platforms. Researchers or data collectors navigate through social media channels such as Twitter, Facebook, and blogs, identifying relevant content based on predefined criteria. This method allows for a more targeted selection of data, ensuring that specific themes, sentiments, or user interactions are captured. The manual collection enables the inclusion of context-rich content that automated tools might overlook, such as nuanced expressions, subtleties, or cultural references. However, this process is resource-intensive and time consuming, as it requires human reviewers to sift through vast amounts of data. Additionally, ethical considerations, such as user privacy and consent, must be carefully addressed when manually collecting data from social media and blogs. Despite its challenges, manual data collection remains valuable for its ability to provide a nuanced understanding of online content, making it a preferred approach for certain research objectives. This study provides a structured approach through an algorithm “B-Bullying Crawler” system.

Algorithm 1 B-Bullying Crawler

```
$Input : URLs$ ▻ List of predefined bully Web URLs
$Output : textlist$ ▻ List of bully crawled texts
procedure $BullyCrawler(URLs)$ ▻ Input
    $textlist := []$
    for $i \leftarrow 1 \rightarrow len(URLs)$ do
        $rule := LinkExtractor(ruleList)$ ▻ Link-based rules
        $date = response.css\,('.story_date :: text').ge()$ ▻ Define crawling date
        $date = re.sub\,('',ate,flags = re.U)$ ▻ Define regular expression
        $D = dic(i,body,tags,images,date,rule)$
        for $j \leftarrow 1 \rightarrow D$ do
            $item = crawl(j)$ ▻ scrapy-base link crawler
            $textlist.appen(item)$
        end for
    end for
    $return\ textlist$ ▻ Output
end procedure
```

3.3.4 Data Pre-processing. The Bengali full stop is replaced with a newline character. Various forms of whitespace (including non-standard ones) are identified using a regex pattern. These are replaced with a single space to standardize the text. The text is then cleaned of these punctuations using regex substitutions. The text is filtered to retain only Bengali characters and spaces. It achieves this by keeping characters with Unicode values greater than the letter ’z’ (which effectively filters out Latin characters) and spaces. Consecutive spaces are reduced to a single space. Finally, the cleaned and processed text is returned. The pre-processing function is to be created specifically to cleanse Bengali text data, rendering it more appropriate for later analysis or training of models

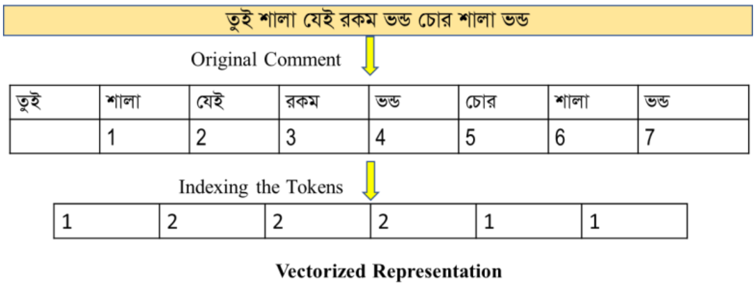
Table 3.1: Data Pre-processing through Cleaning Text

Actual Text	Cleaned Text	Tokens	No. of Tokens
শুভেচ্ছা শুভেচ্ছা👏🏽 রহিল এবং শপথ করুন যে আগামীতে সত ও শিক্ষামুলক ছবি উপহার দিবেন।😜	শুভেচ্ছা শুভেচ্ছা রহিল শপথ করুন যে আগামী সত ও শিক্ষামুলক ছবি উপহার দিব।	[‘শুভেচ্ছা’, ‘শুভেচ্ছা’, ‘রহিল’, ‘শপথ’, ‘করুন’, ‘যে’, ‘আগামী’, ‘সত’, ‘ও’, ‘শিক্ষামুলক’, ‘ছবি’, ‘উপহার’, ‘দিব’]	13

3.3.5 Data Annotation. Taking into consideration the epistemological issues highlighted by Ross et al. [29], this research employed a tiered approach in the selection of annotators.

Two annotators consisted of individuals with diverse academic backgrounds, including undergraduate and postgraduate students. The study established a comprehensive annotation framework to interpret textual content, following rigorous methodological standards. Annotators applied labels like ’Bully’ and ’NotBully’ based on semantic and pragmatic analysis. Table 1 presents the demographic categorization and epistemic stances of annotators. These measures ensure reliability, and effectiveness, and minimize errors. The process involved two annotators for intersubjective validity.

Table 3.2: Data Annotation Example



তুই শালা যেই রকম ভন্ড চোর শালা ভন্ড

তুই	শালা	যেই	রকম	ভন্ড	চোর	শালা	ভন্ড
	1	2	3	4	5	6	7

1	2	2	2	1	1

3.3.6 Data Annotation Verification. Data annotation verification, a crucial step in ensuring the quality and accuracy of labelled datasets, involves the expertise of linguistics professionals who assess and validate annotations based on the opinions of two annotators. In this process, two individuals independently annotate the data, and their annotations are then reviewed by a linguistics expert. The linguist examines the annotations for consistency, coherence, and adherence to predefined guidelines. Any discrepancies or disagreements between the two annotators are carefully scrutinized, and the linguist, drawing on their linguistic expertise, resolves ambiguities and refines the annotations to maintain a high standard of quality. This meticulous verification process helps enhance the reliability of annotated datasets, particularly in linguistically nuanced tasks, ensuring that the 49uropean data accurately reflects the intended linguistic features or patterns. For Example, আলহামদুলিল্লাহ। এ সজীবতা লিয়ে সুস্থ্য ও নিরাপদ থাকুন এ দায়ো করি” Sexual- “তুই তা শালা” Threat- “ জুতা পেটা করা দরকার শালীরে” Troll- “বাংলার সানি লিওন” Religious- “নাস্তিক, ওকে সব জায়গাতেই বয়কট করা হাকে”

Table 3.3: Data Annotation Verification

Variable	Description	Type
Comment	Comments which are collected from social media	Categorical
Categorical	Victim’s harassment, gender	Categorical
Label	Harassment Type (Non-Bully/ Bully)	Categorical

3.3.7 Data Annotation Quality Measures.

- Cohen's Kappa Score: Cohen's Kappa Score is a statistical measure used to evaluate the level of agreement between two annotators (or raters) who classify items into mutually exclusive categories. It is particularly useful in assessing inter-rater reliability while accounting for the possibility of agreement occurring by chance.
- Calculation of Cohen's Kappa:

The formula for Cohen's Kappa (\(\kappa\)) is:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Where:

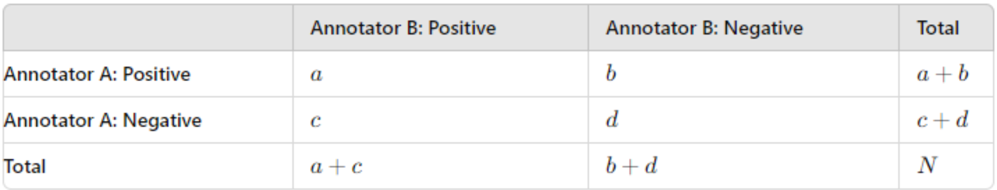
P_o (Observed Agreement): The proportion of items on which the annotators agree.

P_e (Expected Agreement): The proportion of items for which agreement is expected by chance.

- Steps to Calculate Cohen's Kappa: Construct a Contingency Table as following:

  (i) The table categorizes the number of times each annotator assigned each possible label.

  (ii) For example, if two annotators classify items as either positive or negative, the table might look like this:



	Annotator B: Positive	Annotator B: Negative	Total
Annotator A: Positive	a	b	$a+b$
Annotator A: Negative	c	d	$c+d$
Total	$a+c$	$b+d$	N

- Calculate P_o (Observed Agreement):

$$P_o = \frac{a+d}{N}$$

This is the proportion of items for which the annotators agreed.

- Calculate P_e (Expected Agreement):

$$P_e = \frac{(a+b)(a+c)+(b+d)(c+d)}{N^2}$$

This is the proportion of items for which we would expect the annotators to agree by chance.

- Compute Cohen's Kappa:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

This value ranges from -1 to 1, where: 1 indicates perfect agreement and 0 indicates no agreement beyond chance. Negative values indicate less than chance agreement (rare and usually suggests some systematic disagreement).

- The value of κ (\kappa\) is interpreted as follows:

K ≤ 0: No agreement or less than chance agreement.

0.01 - 0.20: Slight agreement.

0.21 - 0.40: Fair agreement.

0.41 - 0.60: Moderate agreement.

0.61 - 0.80: Substantial agreement.

0.81 - 1.00: Almost perfect agreement.

- The statistical metric denoted as [13] is utilized to assess the level of agreement in annotations, as represented by the below equation,

$$k = \frac{Po - Pe}{1 - Pe}$$

The sign p(o) is used to represent the observed and hypothetical probability of annotative agreement, denoting them as separate entities. The Kappa coefficient, which measures inter-annotator agreement, yielded a score of 0.87 for the ’Bully’ category. This indicates that the lexico-semantic elements associated with this category are notably unambiguous, facilitating a high level of agreement across annotators. In contrast, the category lebelled ’NotBully’ demonstrated a Kappa score of 0.73, suggesting a greater degree of intricacy and the possibility of variations in annotation. Table 3.4 shows the Kappa Scores for Annotation Classes in ’CyberBullyDetect’.

Table 3.4: Demographic Categorization and Epistemic Stances of Annotators

Annotator	Academic Level	Research Experience	Experience	Gender	Viewed Cyberbullying	Targeted by Cyber bullying
AN-1	Undergraduate	NLP	1 year	Male	Yes	No
AN-2	Undergraduate	NLP	2 years	Female	Yes	Yes
Expert	Senior	NLP, Ethics	10 years	Male	Yes	Yes

Table 3.5: Kappa Scores for Annotation Classes in 'CyberBullyDetect'

Annotation Category	Kappa Score
Bully	0.87
NotBully	0.73
Mean Kappa Score	0.75

Algorithm 2 B-Bullying Text Annotation, Verification & Quality Measurements

```
Input: $textlist$   ▻ List of bully crawled texts, taken from algorithm 1
Output: B-Bullying ▻ Bully text classification corpus
procedure $\Psi(textlist)$ ▻ Prepossessed text
    B-Bullying = {} ▻ $Bbully$-related text corpus
    // First step: Manual Annotation
    $A_{a,ta} := annotator1(textlist)$ ▻ First manual annotation
    $A_{b,tb} := annotator2(textlist)$ ▻ Second manual annotation
    $eT := [], dx = 1, At := []$
    for $i$ in $rang(1, len(textlist))$ do
        if $(i \ in \ \alpha_{ta}) \ or \ (i \ in \ \alpha_{tb})$ then
            if $A_{ta}[i] == A_{tb}[i]$ then ▻ Both annotators are agreed
                B-Bullying $[idx] = textlis[i], idx := idx + 1$
            end if
            if $A_{ta}[i]! = A_{tb}[i]$ then ▻ Annotators with different agreements
                $eT.append(textlist[i]), T.append(textlist[i])$
            end if
        end if
    end for
    $kapp_m = (A_a, A_b)$
    // Second step: Verification
    $A_{e,te} = expert(eT)$
    for $i$ in $rang(1, len(eT))$ do
        if $A_e[i] == 1$ then ▻ Agreed with any of annotator
            B-Bullying $[idx] = A[i], idx := idx + 1$
        end if
    end for
    // Third step: Quality Measurements of $Bbully$
    $kapp_1 = (A_a, A_e), kapp_2 = \kappa(A_b, A_e)$
    $kapp = Av(kapp1, kapp2)$
    return B-Bullying
end procedure
```

3.4 Corpus Statistics

Following the completion of the crucial six steps, this study successfully establishes a cyberbullying identification corpus. The statistics summarizing the corpus are detailed in Table 3.5. This table provides a comprehensive overview of a Bengali corpus categorized into "Bully" and "Notbully" classes. The corpus consists of a total of 34,422 samples, with 24,108 samples allocated for training and 10,314 for testing. In the training set, there are

12,543 samples labelled as “Bully” and 11,565 samples labelled as “Not-bully.” The testing set comprises 5,358 “Bully” samples and 4,956 “Not-bully” samples. These statistics offer a clear distribution of the dataset, indicating the balance or imbalance between the two classes and the total number of samples available for model training and evaluation. Such information is essential for understanding the characteristics of the dataset and guiding the development and assessment of models for Bengali cyberbullying detection.

Therefore, it denotes the total corpus development in terms of total number of samples available for model training and evaluation, understanding the characteristics of the dataset and guiding the development and assessment of models for Bengali cyberbullying detection.

Table 3.6. Corpus Statistics

Attributes	Values
Total Samples	34,422
Total Training Samples	24,108
Bully Training Samples	12,543
Not- Bully Training Samples	11,565
Total Test Samples	10,314
Bully Testing Samples	5,358
Not- Bully Testing Samples	4,956

Bullying / Not-bullying Data and Result. Few examples basing the corpus statistics are appended below:

Table 3.7: Corpus Statistics – Bullying/ Not-bullying

Data	Bullying	Not-bullying
আগে শাক দিয়ে মাছ ঢাকত এখন চুল দিয়ে [redacted]	Y	N
আপনাওে [redacted] তো? তাইলেই হবে	Y	N
অশিক্ষিত কুত্তার বাচ্চা। জংগি শালা।	Y	N
[redacted] সাফা কবির [redacted] তুই	Y	N
জুতা পিটা করা দরকার জায়েদ [redacted]	Y	N
সাফা কি এখন পরকাল বিশ্বাস করে?	N	Y
আলম বাংলার গরিবের হিরো যার নাম হিরো আলম ♥ ♥ ♥ ♥ জাহেদ খান তুই হয়ে যাবি [redacted] [redacted] তোর বাড়ি ইন্ডিয়া না পাকিস্তান	N	Y

■ Abusive cyberbullying word

3.5 Development, Verification, and Selection of the Cyberbullying Text Identification Model

3.5.1 The primary goal of this study is to create a text identification system for low-resource cyberbullying. To achieve this objective, the research is structured into three main steps: (i) Text-to-Feature extraction, (ii) Development of statistical, deep learning, and transformer-based language models, and (iii) Evaluation of models and selection of the top-performing model. The overall methodological procedure is presented in Figure 3. In the following subsequent subsections, each of the steps is described.

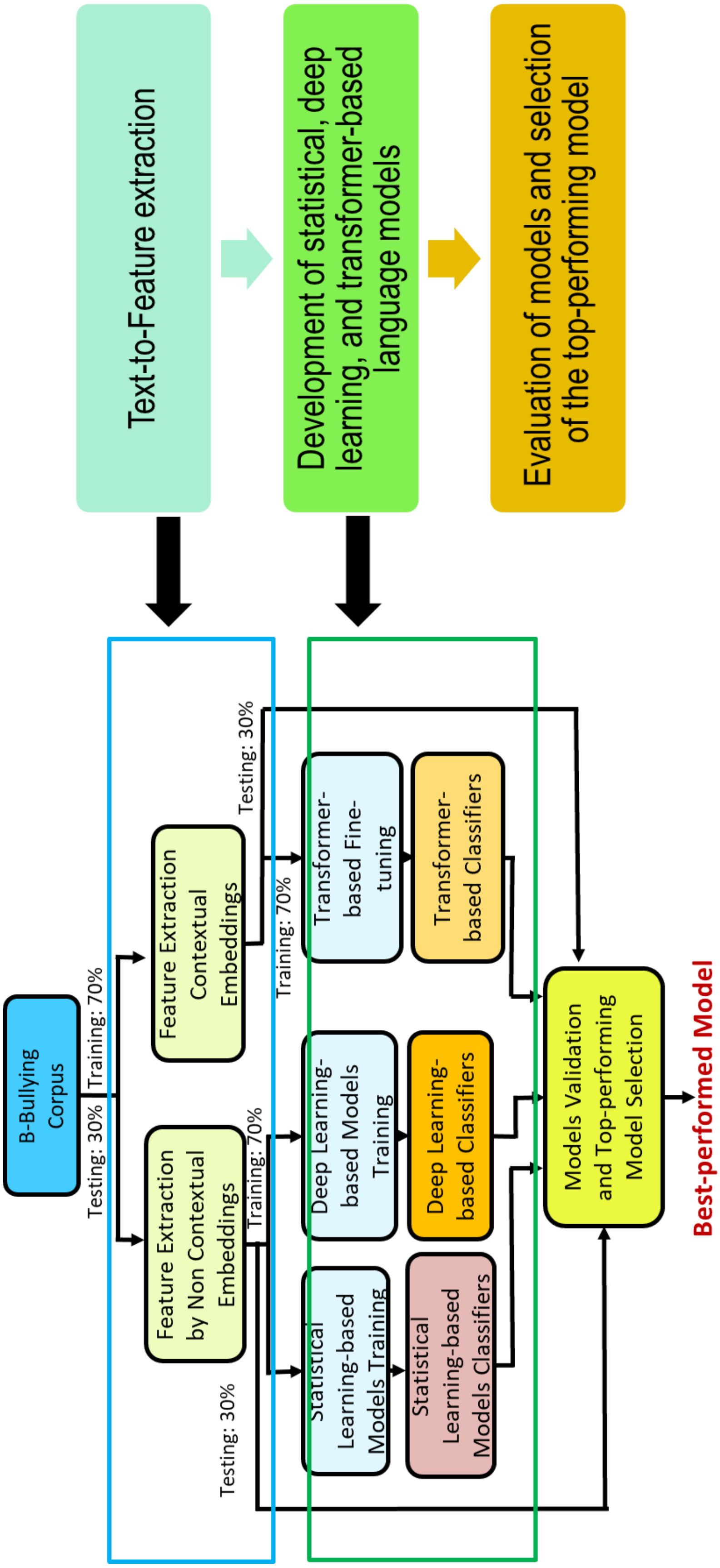

Text-to-Feature extraction
Development of statistical, deep learning, and transformer-based language models
Evaluation of models and selection of the top-performing model
B-Bullying Corpus
Training: 70%
Testing: 30%
Feature Extraction Contextual Embeddings
Feature Extraction by Non Contextual Embeddings
Testing: 30%
Training: 70%
Training: 70%
Testing: 30%
Transformer-based Fine-tuning
Transformer-based Classifiers
Deep Learning-based Models Training
Deep Learning-based Classifiers
Statistical Learning-based Models Training
Statistical Learning-based Models Classifiers
Models Validation and Top-performing Model Selection
Best-performed Model

Fig. 3.5. Abstract View of Cyberbullying Text Identification Model’s Development and Top-Performing Models Selection

3.5.2 Detail Process Breakdown

Traditional Feature Extraction: Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. TF-IDF combines two measures:

- Term Frequency (TF): The number of times a word appears in a document.
- Inverse Document Frequency (IDF): A measure of how much information the word provides, i.e., whether the word is common or rare across all documents.
- By calculating the TF-IDF score for each word, we generate feature vectors that represent the importance of words within the documents. These vectors are then used as input features for the classification model.

Local Contextual Feature Extraction.

- GloVe (Global Vectors for Word Representation):
  - Objective: Capture global word-word co-occurrence statistics from a corpus.
  - Method: Generate word embeddings based on aggregated global co-occurrence statistics of words in a corpus. These embeddings represent words in a continuous vector space where semantically similar words are mapped to nearby points.

- FastText:
  - Objective: Handle rare words and morphological richness.
  - Method: Extend Word2Vec by considering subword information. FastText breaks words into character n-grams and represents each word as a sum of its character n-grams. This approach helps in generating embeddings for rare words and morphologically complex languages.

- Word2Vec:
  - Objective: Capture semantic relationships between words.
  - Method: Use either Continuous Bag-of-Words (CBOW) or Skip-gram models to generate word vectors. CBOW predicts the target word from context words, while Skip-gram predicts context words from a target word. The generated embeddings are used to represent words in a vector space.

Contextual Feature Extraction: Pre-trained Multilingual-BERT. Multilingual-BERT (mBERT) is a transformer-based model designed to understand the context of a word based on its surrounding words. It is trained on a large corpus of multiple languages, allowing it to handle multilingual text data.

- Objective: Generate contextualized word embeddings that capture the meaning of a word in its specific context.

- Method: Use the transformer architecture, which relies on self-attention mechanisms to model the relationships between all words in a sentence simultaneously. This approach allows mBERT to produce embeddings that consider the full context of a word, improving the understanding of nuanced language usage.

Model Training and Evaluation. After extracting features using the methods mentioned above, the next steps involve:

- **Model Development:** Train various machine learning and deep learning models using the extracted features. Evaluate the performance of each model using metrics such as accuracy, precision, recall, F1-score, and Area Under the Curve (AUC). This step helps in identifying the most effective model for cyberbullying detection. Select the best-performing model based on the evaluation metrics. This model is then fine-tuned and optimized for deployment.

- **Predicted Output – Bully or Not Bully:** The final output of the model is a binary classification indicating whether a given text is identified as “bully” or “not bully.” This decision is based on the features extracted and the patterns learned by the selected dataset model during training.

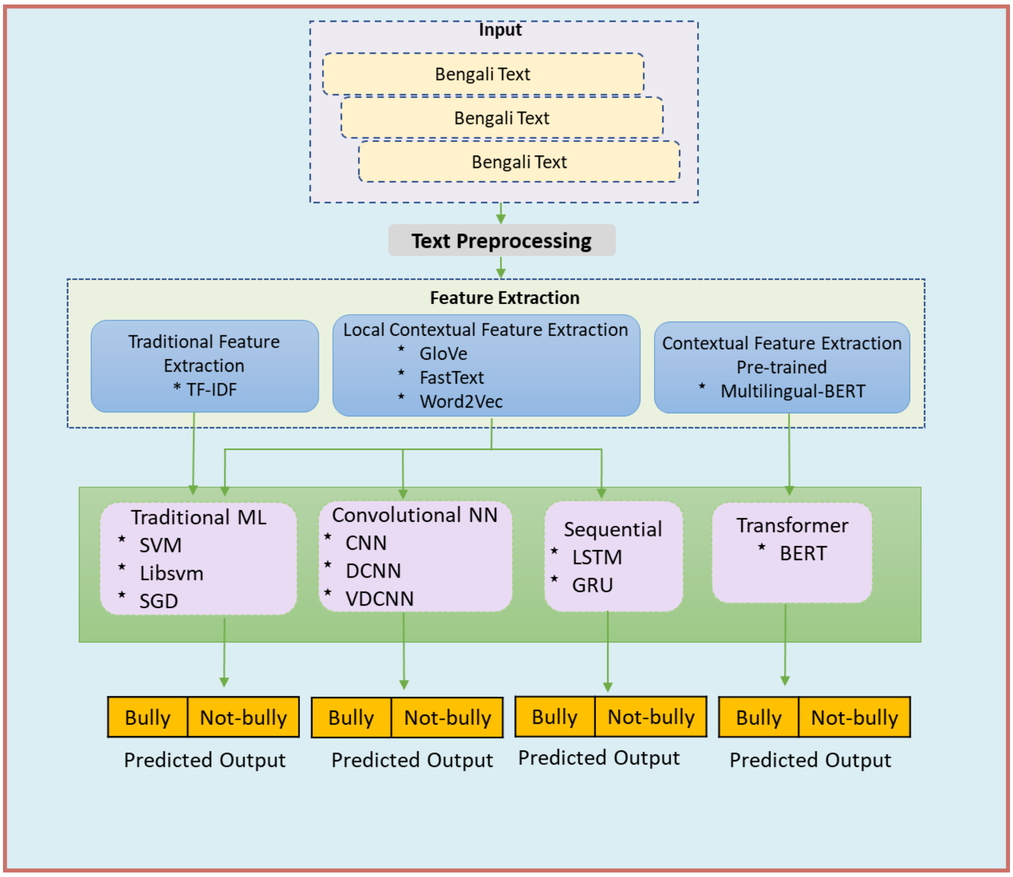


Fig. 3.6. Detail Process Breakdown of the Cyberbullying Text Identification Model

3.5.3 Text-to-Feature extraction. In this study, it has employed two types of feature extraction methods, i.e., (1) Non-contextual and (2) Contextual.

Non-contextual. The three non-contextual embedding models are used for Bengali text-to-feature extraction purposes, i.e., GloVe [26], FastText [6], and Word2Vec [22]. In the realm of Bengali cyberbullying text analysis, the process of text-to-feature extraction plays a crucial role in uncovering meaningful patterns and representations. This research leverages three prominent word embedding techniques, namely GloVe, FastText, and Word2Vec, to extract informative features from Bengali cyberbullying text. GloVe captures global word co-occurrence statistics, FastText considers sub-word information, and Word2Vec models word embeddings based on contextual similarity. By employing these techniques, the study aims to harness the unique linguistic nuances of Bengali cyberbullying instances, enabling a more nuanced understanding of the language-specific characteristics associated with such content. The comparative analysis of these word embedding methods contributes to the development of a robust text-to-feature extraction pipeline tailored for Bengali cyberbullying detection.

Table 3.8: Non-contextual Embedding Models

<table>
<tr><th colspan="2">Non-contextual Embedding Models</th></tr>
<tr><td><b>GloVe</b></td><td>Captures global word co-occurrence statistics</td></tr>
<tr><td><b>FastText</b></td><td>Considers sub-word information</td></tr>
<tr><td><b>Word2Vec</b></td><td>Embeddings based on contextual similarity</td></tr>
<tr><td colspan="2">• Aims to harness the unique linguistic nuances of Bengali cyberbullying instances, enabling language-specific characteristics associated with such content
• The comparative analysis of these word embedding methods contributes to the development of a robust text-to-feature extraction tailored for Bengali cyberbullying detection</td></tr>
</table>

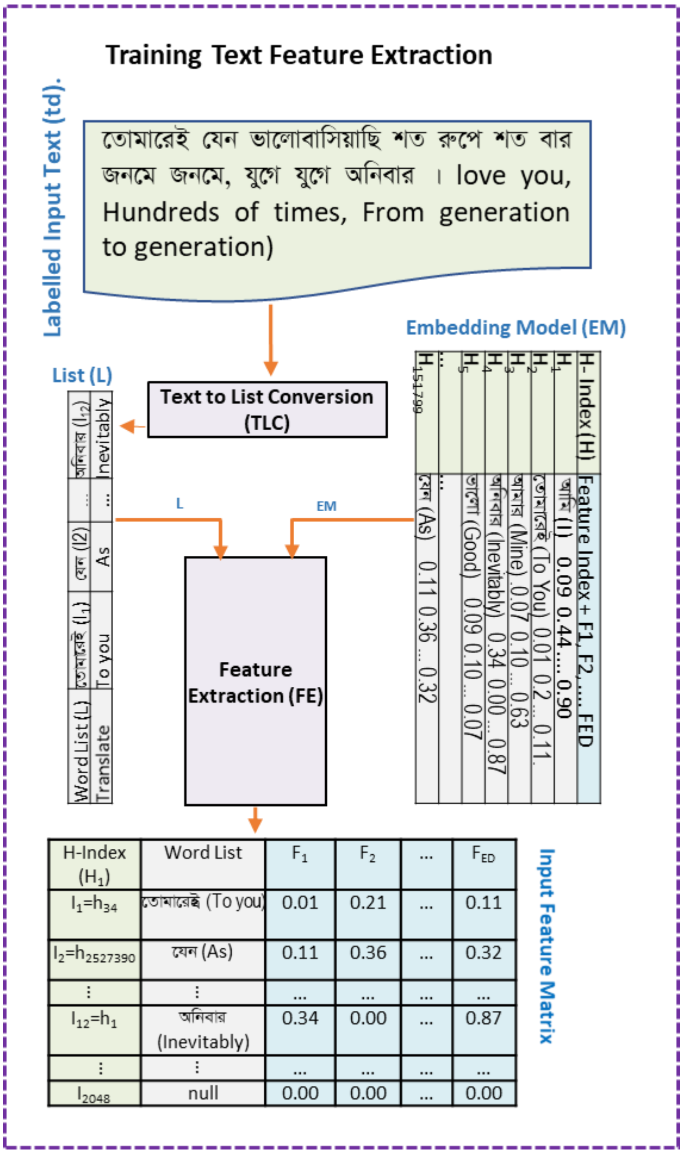
Contextual. In answer to the research question RQ2 this study deployed contextual feature extractors, such as BanglaBERT, XML-RoBERTa, IndicBERT, and ELECTRA, which are advanced language models designed to capture contextual information from text in the Bengali language. These models belong to the family of transformer-based architectures, which have demonstrated remarkable success in natural language processing tasks. BanglaBERT is specifically tailored for Bengali, offering contextualized embeddings by considering the unique linguistic intricacies of the language. XML-RoBERTa extends this idea, emphasizing the importance of contextual embeddings in handling complex structures and multiple languages. IndicBERT, designed for various Indic languages including Bengali, focuses on contextualized representations for improved language understanding. ELCTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) is an approach that introduces an adversarial training strategy to pretrain transformer-based models, enhancing their efficiency in handling contextual features. These contextual feature extractors contribute significantly to tasks like sentiment analysis, named entity recognition, and cyberbullying detection in Bengali text, as they empower models to comprehend the nuanced contextual meanings within the language. Their incorporation in natural language processing pipelines enriches the representation of Bengali text, facilitating more accurate and context-aware language understanding in various applications.

Table 3.9: Contextual Embedding Models

Contextual Embedding Models	
BanglaBERT	Specifically tailored for Bengali, offering contextualized embeddings by considering the unique linguistic intricacies of the language.
XML-RoBERTa	Emphasizing the importance of contextual embeddings in handling complex structures and multiple languages.
IndicBERT	Designed for various Indic languages including Bengali, focuses on contextualized representations for improved language understanding.
ELECTRA	An argumentative training strategy to pretrain transformer-based models, enhancing their efficiency in handling contextual features.

Their incorporation in natural language processing enriches the representation of Bengali text, facilitating more accurate and context-aware language understanding in various applications. The comparative analysis of these word embedding methods contributes to the development of a robust text-to-feature extraction tailored for Bengali cyberbullying detection. The training/testing text feature extraction is performed in three phases: text to list conversion (TLC), GloVe model generation, and feature extraction (FE). The TLC is initialized with the labelled input text document (td i), and it converts the input text as a word list vector (L). The list vector is a collection of words defined as L = [l]. 1 where l naïve th denotes the naïve , l 2 , l 3 , …,l N], word for naïve = 1, 2, 3, …,N. N is the maximum length of L and N=2,048. If an input text contains more than 2,048 words, then it is truncated to the first 2,048 words. It uses zero padding if the number of words is smaller than 2,048. The FE process considers the list vector (L) an embedding model (EM) as inputs. The list L is a 1D vector containing a total of 2,048 words. The EM is a 2D vector containing a total of 1,517,390 unique words each assigned an individual h-index and ED corresponding feature values whereas ED ∈{25,50,100,150,200,250,275,300,325,350,400}.

For every word in L, a set of (Hi) is generated using FE. FE process extracts the features from the EM by mapping (Hi) to EM. If the L of (Hi) is found in EM, then FE returns the corresponding (Hi) feature values. Otherwise, it returns a null vector for ED features. For example, the first word (l1) is mapped to the index h2 of EM. The FE phase generates an output of feature matrix (Vd*F), where Vd denotes the number of words and F denotes the number of features. The ith word li extracts a vector of ED features (F1,F2,F3,…,FED) from the EM. Each word of a text document is arranged in rows and corresponding features in columns. As a result, each document is represented by an input feature matrix of dimension (2048×ED)where each row of the feature matrix is an H-Index (Hi),Hi : Hi ={h1,h2,…,h2048} and each column is a feature Fj :Fj ={F1,F2,F3,…, FED}. The testing text feature extraction takes the unlabelled text document as the input and produces a 2D feature matrix as the output using the GloVe model generation and FE processes. The process of the 2D feature matrix generation is the same as the training text feature extraction technique. The upper right part of Fig. 3.7 illustrates the process of the testing text feature extraction.


Training Text Feature Extraction
Labelled Input Text (td).
Text to List Conversion (TLC)
List (L)
L
EM
Embedding Model (EM)
Feature Extraction (FE)
Input Feature Matrix

তোমারেই যেন ভালোবাসিয়াছি শত রুপে শত বার জনমে জনমে, যুগে যুগে অনিবার । love you, Hundreds of times, From generation to generation)

Word List (L)	তোমারেই (I_1)	যেন (I2)	…	অনিবার (I_{12})
Translate	To you	As	…	Inevitably

H- Index (H)	Feature Index + F1, F2,…. FED
H_1	আমি (I) 0.09 0.44…. 0.90
H_2	তোমারেই (To You) 0.01 0.2 … 0.11
H_3	আমার (Mine) 0.07 0.10 … 0.63
H_4	অনিবার (Inevitably) 0.34 0.00 … 0.87
H_5	ভালো (Good) 0.09 0.10 … 0.07
…	…
H_{151799}	যেন (As) 0.11 0.36 … 0.32

H-Index (H_1)	Word List	F_1	F_2	…	F_{ED}
I_1=h_{34}	তোমারেই (To you)	0.01	0.21	…	0.11
I_2=$h_{2527390}$	যেন (As)	0.11	0.36	…	0.32
⋮	⋮	…	…	…	…
I_{12}=h_1	অনিবার (Inevitably)	0.34	0.00	…	0.87
⋮	⋮	…	…	…	…
I_{2048}	null	0.00	0.00	…	0.00

Fig. 3.7. Feature Extraction Model

3.6 Development of Statistical, Deep Learning, and Transformer-based Language Models

In this study, it has employed two statistical, four deep learning, and four transformer-based language models to verify the Bengali Cyberbullying text identification system.

3.6.1 Statistical Models. This study explores the effectiveness of three machine learning (ML) techniques: GPU-based Support Vector Machine (SVM), GPU-based Libsvm and Stochastic Gradient Descent (SGD) for Bengali for Cyberbullies text [13]. The ML-based classifier models are developed and tuned on a created corpus. SVM and Libsvm share similar parameters, including a sigmoid kernel, tol of 0.00001, and a decision function shape of ’over,’ with Libsvm demonstrating faster performance. The SGD classifier employs parameters loss = modified_huber and alpha = 0.001, with the remaining parameters set as defaults.

Table 3.10: Statistical Models

Statistical Models	
GPU-based Support Vector Machine (SVM)	SVM and Libsvm share similar parameters, including a sigmoid kernel, tol of 0.00001, and a decision function shape of ’over,’ with Libsvm demonstrating faster performance.
GPU-based Libsvm	
Stochastic Gradient Descent (SGD)	The SGD classifier employs parameters loss = modified_huber and alpha = 0.001, with the remaining parameters set as defaults.

3.6.2 Deep Learning-based Models CNN. Employing a single-layer, multi-kernel Convolutional Neural Network (CNN) architecture, this model investigates Bengali Cyberbullies text performance with three embedding models (FastText, GloVe, & Word2Vec). The distinct kernels are set to 3, 4, and 5, with corresponding filter numbers of 128, 128, and 256. Following the convolution layer, there is a 1D maxpool layer and activation layers [14]. Subsequently, the pooled features are concatenated and subjected to a dropout operation with a threshold value of 0.3.

VDCNN. Introducing the Variable-Size Deep Convolutional Neural Network (VDCNN) architecture for Bengali for Cyberbullies text identification purpose [16]. Unlike the original VDCNN, which operates on character-level embeddings, this adaptation combines VDCNN with different embedding techniques to enhance

Bengali text classification performance. By reducing certain convolution operations, it addresses training time and model overfitting issues encountered by the original VDCNN.

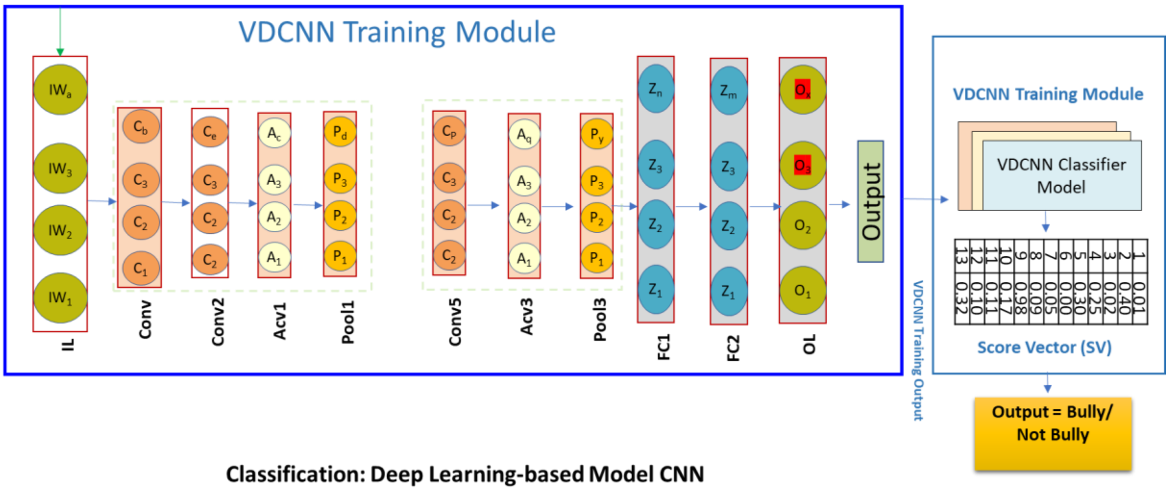


Fig. 3.8. VDCNN Model

LSTM. In this study, a two-layer LSTM is utilized with the following parameters: max sequence length = 256, hidden dimensions = 128, 256, batch size = 12, dropout rates = 0.50, 0.40, loss function = categorical_crossentropy, optimizer = adam, and activation function = softmax [2]. The model is trained for a maximum of 50 epochs on the developed corpus. It is noted that an increased number of sequences negatively impacts classification performance. Additionally, experiments are conducted with max sequence lengths of 1024 and 2048 in this research.

GRU. The two-layer GRU model is configured with the following parameters: hidden states = 128, 128, max sequence length = 512, batch size = 32, epochs = 80, dropout rates = 0.30, 0.25, loss function = categorical_crossentropy, optimizer = adam, and activation functions = tanh, softmax [2]. The last GRU layer is followed by a 1D max-pool layer. Subsequently, the 512 feature values are concatenated for the softmax layer, responsible for generating predictions in the expected category.

3.6.3 Transformer-based Language Models. The training module for Transformer-based Language Models, including mBERT, bELECTRA, XML-RoBERTa, IndicBERT, DistilBERT, and BanglaBERT, involves initial pre-training on a large multilingual corpus to learn contextualized representations [15]. Subsequently, these models undergo task-specific hyperparameters adaption, text-to-feature extraction, and finetuning to the intricacies of Bengali cyberbullying text identification.

Table 3.11: Transformer-based Language Models

Transformer-based Language Models	
mBERT, bELECTRA, XML-RoBERTa, IndicBERT, DistilBERT, and BanglaBERT	• Involves initial pre-training on a large multilingual corpus to learn contextualized representations • Subsequently, these models undergo task-specific hyperparameters adaption, text-to-feature extraction, and finetuning to the intricacies of Bengali cyberbullying text identification

Mmulti-lingual transformer-based language models are selected for the fine-tuning purpose. The kth transformer-based fine-tune training function ($\Psi T\ k$ (.)) take the jth sample feature matrix $eMC\ k$, and trained using Eq. (3). The training phase is initially fed to the Multi-head Attention block. The multi-head attention block processes each input batch in five steps: (i) Three attention heads, i.e., query, key, and value individually process the input through linear transformations, capturing distinct features; (ii) Attention scores for each head are computed by taking the dot product of the projected input with learnable parameters known as attention weights; (iii) Using the attention scores, values at each position in the input sequence are weighted, facilitating a summation that consolidates information from various parts of the sequence, enabling the model to focus on relevant portions and disregard irrelevant ones; (iv) The outputs from all attention heads are concatenated and linearly transformed, yielding the final output of the multi-head attention block. This fusion of information from different heads enables the model to capture diverse patterns and relationships in the input text; and (v) Typically, a non-linear activation function, such as the Rectified Linear Unit (ReLU), is applied to the output, introducing non-linearity and enhancing the model’s expressive power. A Transformer-based Language Model is outlined in Fig: 3.9.

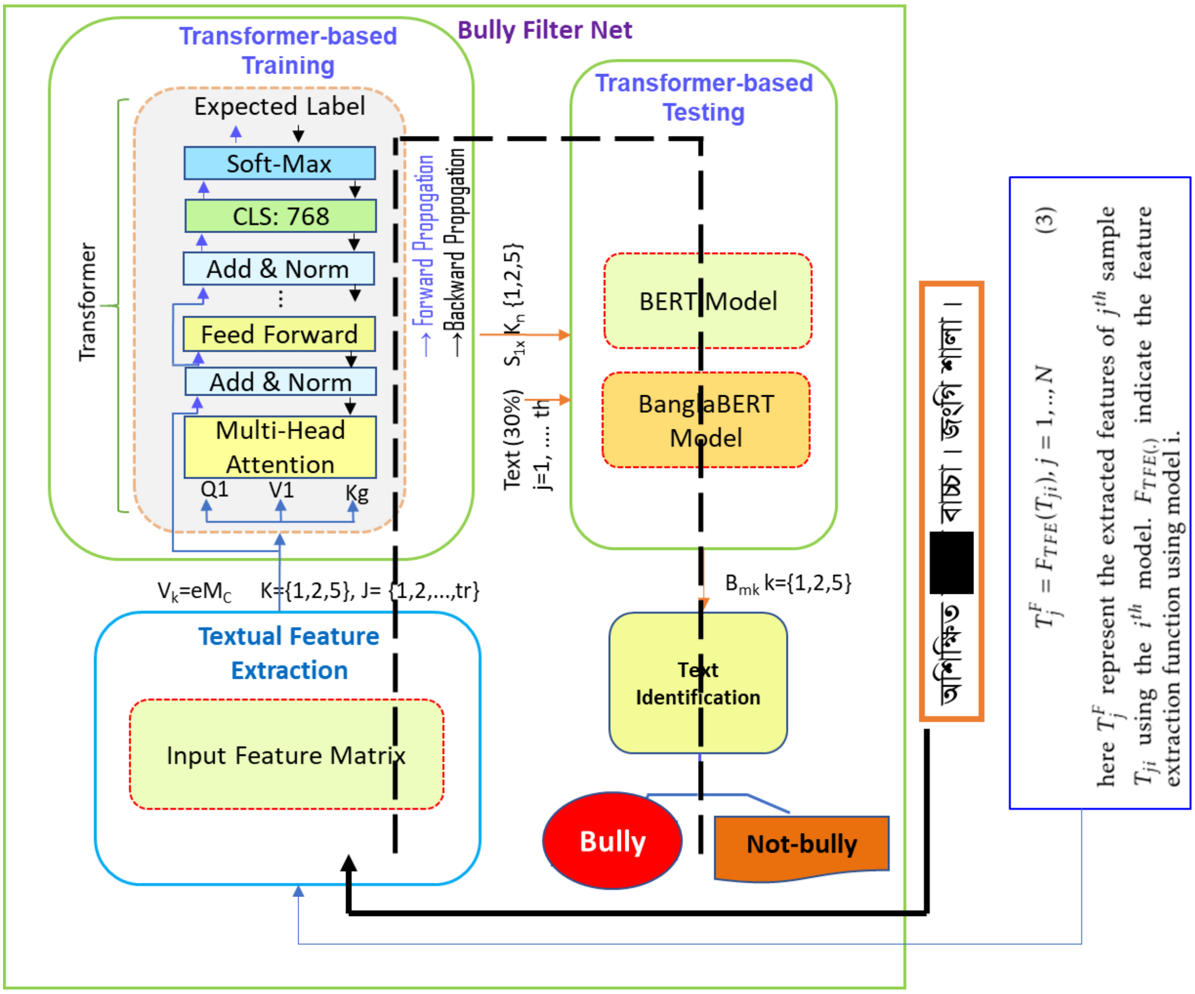

Bully Filter Net
Transformer-based Training
Transformer
Expected Label
Soft-Max
CLS: 768
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Q1
V1
Kg
→ Forward Propogation
→Backward Propogation
Transformer-based Testing
BERT Model
BanglaBERT Model
S_{1x} K_n {1,2,5}
Text (30%)
j=1, ... th
V_k=eM_C
K={1,2,5}, J= {1,2,...,tr}
Textual Feature Extraction
Input Feature Matrix
B_{mk} k={1,2,5}
Text Identification
Bully
Not-bully
অশিক্ষিত [redacted] বাচ্চা । জংলি শালা ।
$T_j^F = F_{TFE}(T_{ji}), j = 1,..,N$ (3)
here T_j^F represent the extracted features of j^{th} sample T_{ji} using the i^{th} model. $F_{TFE(.)}$ indicate the feature extraction function using model i.

Fig: 3.9. Transformer-based Language Models

3.6.4 Example of Attention Mechanism in Transformer-Based Model for Cyberbullying Detection in Bengali

In this study, the attention mechanism in a Transformer-based model, such as BanglaBERT is used to detect cyberbullying in Bengali text. Here, it focuses on explaining the process and the insights gained from the attention weights.

- Input Preparation:

Text Example: Considered the Bengali sentence: "তুমি খুব খারাপ, আমি তোমাকে ঘৃণা করি।" (Translation: "You are very bad, I hate you.")

Tokenization: The text is tokenized into smaller units (subwords or tokens) that the model can process. Using BanglaBERT's tokenizer, this might look like:

["[CLS]", "তুমি", "খুব", "খারাপ", ",", "আমি", "তোমাকে", "ঘৃণা", "করি", ".", "[SEP]"]

- Model Input: The tokenized input is converted into input embeddings that the BanglaBERT model can process. This includes token embeddings, positional embeddings, and segment embeddings.

- Attention Mechanism:

Self-Attention: Each token in the sequence attends to every other token, generating attention weights that indicate the importance of each token in the context of others.

Multi-Head Attention: Multiple heads allow the model to capture different aspects of the token relationships. Each head produces its own set of attention weights.

Attention Weights Analysis: The attention weights for each head and layer was extracted and visualized. These weights helped to understand which tokens the model focuses on when determining if the text contains cyberbullying.

Visualization of Attention Weights: Below is a simplified visualization (heatmap) of attention weights for one of the heads. Darker colors indicate higher attention scores.

```
Attention Weights Heatmap
---------------------------------------------
[CLS] | তুমি | খুব | খারাপ | ,  | আমি | তোমাকে | ঘৃণা | করি | .  | [SEP]
---------------------------------------------
[CLS] | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05
তুমি  | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10
খুব  | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15
খারাপ | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30
,    | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05
আমি  | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10
তোমাকে | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10
ঘৃণা  | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15
করি  | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05
.    | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05
[SEP] | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05
```

High Attention Tokens: "খারাপ" (bad) and "ঘৃণা" (hate) have the highest attention weights, indicating that the model finds these words most relevant for detecting cyberbullying in this sentence.

Contextual Importance: The tokens "তুমি" (you) and "তোমাকে" (you) also have significant attention, suggesting that the model is focusing on who the harmful words are directed towards.

Punctuation and Structure: Punctuation tokens like "," and "." have low attention weights, which is expected as they do not contribute significantly to the semantic meaning in this context.

- Explainability and Insights:

Transparency: By visualizing the attention weights, we can see that the model is correctly identifying harmful words and understanding their importance in the context of the sentence.

Validation: This helps validate that the model is not just making predictions based on spurious patterns but is genuinely understanding the content and context of the input text.

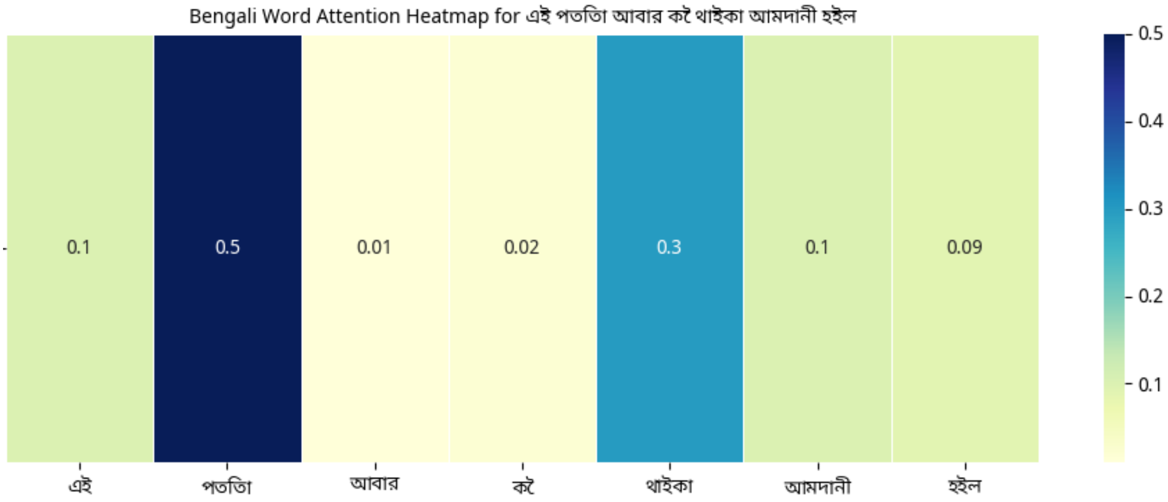


Fig: 3.10. Bengali Cyberbullying Word Attention Heatmap

3.6.5 Hyperparameters Adaption.

Due to shortage of training samples, this study adapted the transformer based language models using Eq. 2

$$H_i^O = F_{HPO}(H^I_{\text{naïve}})),\ \text{naïve} = \text{mBERT}, \ldots, \text{BanglaBERT} \quad (2)$$

here H_i^O represents the optimised hyperparameters and H_i^O represents the initial hyperparameters of transformer-based language models, i.e., mBERT to BanglaBERT. The function F_{HPO} (.) indicates the hyperparameters adaption function which adapted the maximum sequence length and batch size.

Batch Size Versus Accuracy. To empirically labelled the impact of batch size on accuracy, experiments were conducted using various batch sizes while fine-tuning transformer-based models for identifying cyberbullying text. The results are summarized below:

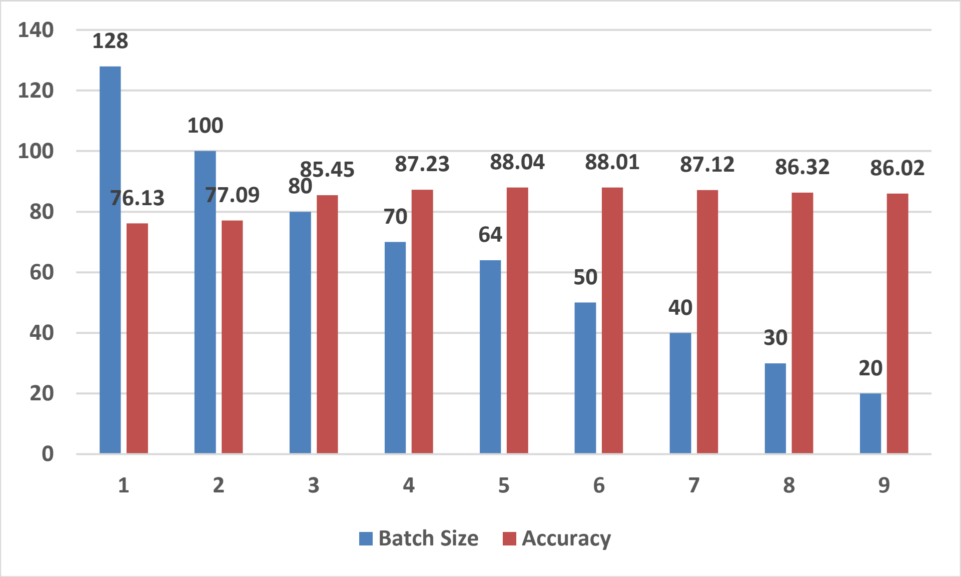


Fig: 3.11. Hyperparameters Adaption: Batch Size Versus Accuracy

Analysis:

- Optimal Batch Size: In this study, a batch size of 64 yielded the highest accuracy of 88.04%. This suggests that this batch size strikes a balance between efficient training and effective generalization for the given task.

- Diminishing Returns: Increasing the batch size beyond 64 led to a decrease in accuracy, highlighting the point of diminishing returns where larger batches do not necessarily translate to better performance.

- Memory Considerations: While batch sizes of 16 and 32 also performed well, they may require longer training times due to less efficient hardware utilization. Conversely, batch sizes of 128 and 256, despite being more memory-intensive, did not improve accuracy and even showed a decline, likely due to overfitting.

Epoch Versus Accuracy. To empirically labelled the impact of the number of epochs on accuracy, experiments were conducted using different numbers of epochs while fine-tuning transformer-based models for identifying cyberbullying text. The results are summarized below:

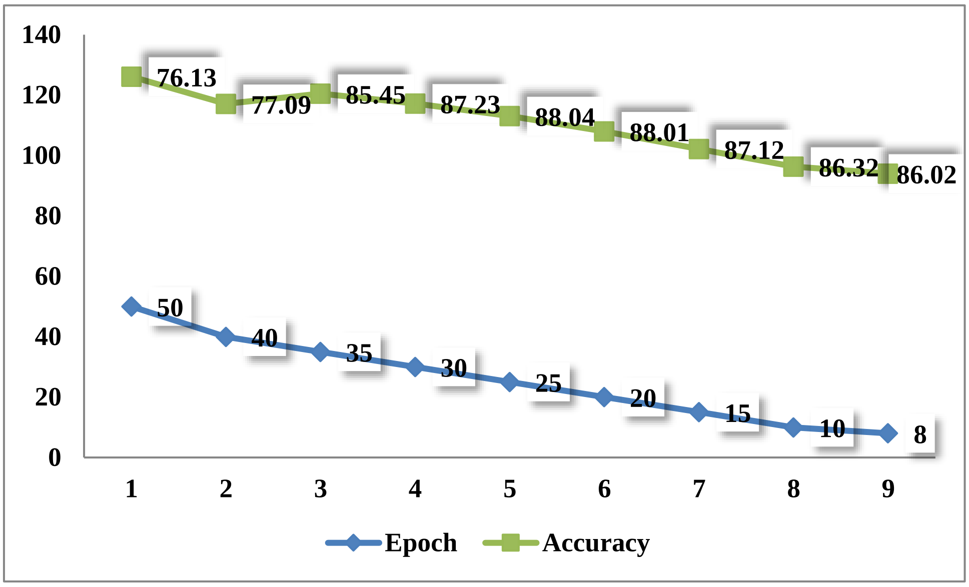


Fig: 3.12. Hyperparameters Adaption: Epoch Versus Accuracy

Analysis:

- Optimal Number of Epochs: In this study, training for 5 epochs yielded the highest accuracy of 88.04%. This suggests that 5 epochs are sufficient for the model to learn the data patterns effectively without overfitting.
- Diminishing Returns: Training for more than 5 epochs resulted in a slight decrease in accuracy, indicating the onset of overfitting where the model starts to memorize the training data rather than generalizing from it.

- Underfitting: Training for fewer epochs (1 or 3) led to lower accuracy, highlighting that the model did not have enough time to learn from the data adequately.

Max. Sequence Length Versus Accuracy. To empirically labelled the impact of maximum sequence length on accuracy, experiments were conducted using different sequence lengths while fine-tuning transformer-based models for identifying cyberbullying text. The results are summarized below:

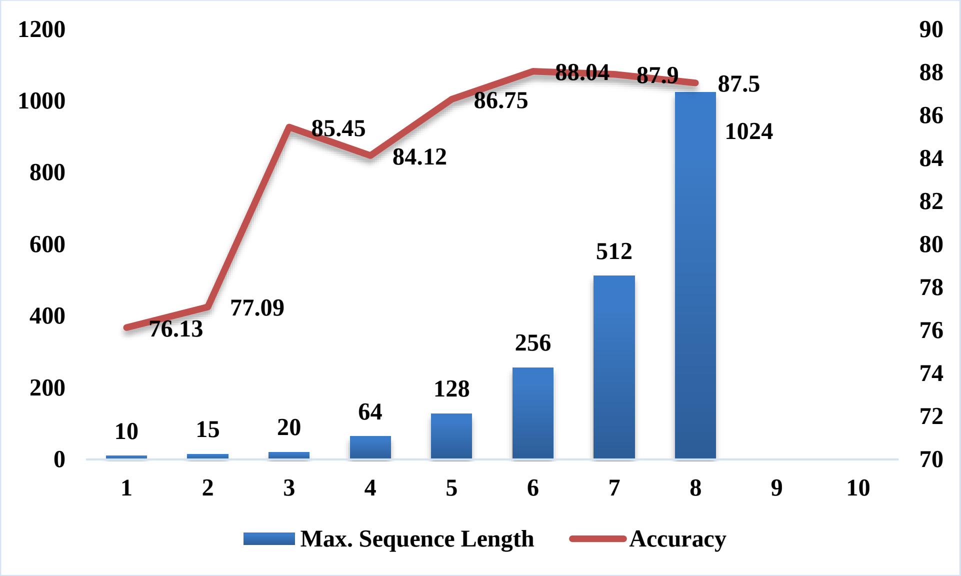


Fig: 3.13. Hyperparameters Adaption: Max. Sequence Length Versus Accuracy

Analysis

- Optimal Sequence Length: In this study, a maximum sequence length of 256 tokens provided the highest accuracy of 88.04%. This suggests that 256 tokens are sufficient to capture the necessary context for identifying cyberbullying text without introducing significant noise.
- Diminishing Returns: Increasing the sequence length beyond 256 tokens did not lead to significant improvements in accuracy and, in some cases, resulted in a slight decrease. This indicates that additional context beyond 256 tokens does not contribute significantly to better performance and may introduce unnecessary complexity.

- Truncation Impact: Shorter sequence lengths (64 or 128 tokens) resulted in lower accuracy, highlighting the importance of capturing sufficient context to understand the nuances of cyberbullying text.

3.6.6 Text-to-feature Extraction. The transformer-based language models are extracted the linguistic features using Eq. 3.

$$T^{F}{}_{j} = F_{TFE}(T_{ji}),\ j = 1, .., N \qquad (3)$$

here $T^{F}{}_{j}$ represent the extracted features of jth sample T^{ji} using the ith model. $F_{TFE}(.)$ indicate the feature extraction function using model i.

3.6.7 Fine-tuning. Fine-tuning is a crucial process in the context of machine learning, especially when working with pre-trained language models like transformer-based models. It involves adjusting the parameters of a pre-trained model on a specific task or domain to enhance its performance. Fine-tuning is particularly valuable when the available labelled data for a specific task is limited. Fine-tuning allows the model to adapt to the specific characteristics and nuances of the target task or dataset, improving its ability to make accurate predictions. The fine-tuning process typically involves feeding the pre-trained model with task-specific data, and updating its weights based on the gradients computed during the training process. This helps the model to specialize in the target task while retaining the knowledge gained during pre-training on a large corpus. These transformer-based language models are fine-tuned using the Eq. 4.

$$^{\theta}\text{fine-tuned} = {}^{\text{arg min}}{}_{\theta} \sum\cdot{}_{\text{task-specific}}{}^{(\theta)}\ \text{DG} + \lambda \sum\cdot{}_{\text{naïve}} \|\theta_{i} - \theta\text{pre-trained,naïve}\ \|^{2}$$

here $\theta_{\text{fine-tuned}}$ represents the fine-tuned model parameters, $L_{\text{task-specific}}(\theta)$ s the task-specific loss function, $\theta_{i} - \theta_{\text{pre-trained}}$ denote the parameters of the fine-tuned and pre-trained models, respectively. The λ is a regularization hyperparameter that controls the balance between the task-specific loss and the regularization term. This equation captures the fine-tuning process where the model is optimized for a task-specific objective while leveraging knowledge gained from pre-training on a large corpus. Regularization helps prevent overfitting during the fine-tuning process. Evaluation of Models and Selection of the Top-performing Model.

Algorithm 3 Hyperparameters Tuning of Fine-tuning Language Model

```
$Input$ : $h$ & B-Bullying : B-Bullying $^{train}$ ∪ B-Bullying $^{test}$ ⊳ Tune-able selected hyperparameters (h) & Classification corpus (B-Bullying)
$Output$ : $h^*$ ⊳ Tuned hyperparameters based on B-Bullying  corpus
**procedure** $G(h, B\text{-}Bullying$ : B-Bullying $^{train}$ ∪ B-Bullying $^{test}$)  ⊳ Input
$maxBatch := 20$, $maxSequence := 512, axEpoch := 20$
$e[^{C}k{:}A_s] := Fk^{X}(AEk^{C}, B\text{-}Bullying_{j}), j \in 1,2,...,A_s$
$globalAccuracy := -1, s := 0, ms := 0$
**for** $ep \in rang(1, maxEpoch)$ **do**  ⊳ $h \in \{M_{SL}, B_S\}$
      $acc, s, ms = \Omega^{T}(eM^{C}, h)$,                    ⊳
                              $k$  $[k{:}A_s]$
   $M_{SL} \in \{50 : 512\}$, $B \in \{3 : 20\}$
 **if** $acc > globalAccuracy$ **then**
$globalAccuracy := acc, _{SL} := ms, B_S := bs, epoch := ep$
      **end if**
 **if** $EarlyStoppin(monitor = val\_accuracy, globalAccuracy)$ **then**
      $h^*[0] := M_{SL}, h^*[1] := B_S, h^*[2] := ep$
         $return\ h^*$                               ⊳ Output
      **end if**
      **end for**
$h^*[0] := M_{SL}, h^*[1] := B_S, h^*[2] := eph$
$return\ h^*$                                   ⊳ Output
**end procedure**
```

Chapter 4

Experimentation Results and Discussion

4.1 Experimental Results and Discussions

Within this section, it firstly provides an overview of the experimental setup and the chosen evaluation measures. Subsequently, it delves into the presentation and discussion of the results.

4.2 Experimental Setup and Evaluation Measures

The models were deployed on the Google Colaboratory platform with Python 3 and a Google Cloud Engine backend with GPU capability. This study's computing resources included 12.5GB of RAM and 64GB of disk space. The dataset was 74uropean with Python's Pandas (version 1.1.4) and NumPy (version 1.18.5) libraries. The Scikit-Learn package (version 0.22.2) was used to create traditional machine learning models, while Keras (version 2.4.0) and TensorFlow (version 2.3.0) were used to create deep learning models. The ktrain library (version 0.25) was used for models using Transformer architectures. The dataset was partitioned into three sets: training, validation, and test. The training set supported the models' learning phase, whereas the validation set aided in hyperparameter tuning. Details of experimental setup is appended below:

Table 4.1: Experimental Setup

Experimental Setup	Remark
Google Colab platform with Python 3 and a Google Cloud Engine backend with GPU capability.	Computing resources included 12.5GB of RAM and 64GB of disk space.
Python’s Pandas (version 1.1.4) and NumPy (version 1.18.5) libraries	Dataset was analyzed.
The Scikit-Learn package (version 0.22.2).	Used to create traditional machine learning models.
Keras (version 2.4.0) and TensorFlow (version 2.3.0).	Used to create deep learning models.
The ktrain library (version 0.25).	Used for models using Transformer architectures.
The dataset was partitioned into three sets: training, validation, and test.	The training set supported the models’ learning phase, whereas the validation set aided in hyperparameter tuning.

The final evaluation was carried out on an unknown test set using a variety of statistical measures, as specified in the following equations:

Precision (p): It quantifies the proportion of true positive samples within the samples classified as positive.

$$Accuracy = \frac{Number\ of\ Correct\ Predictons}{Number\ of\ Samples} \times 100 \quad (5)$$

$$Precision\ (p) = \frac{Truepositive}{Truepositive + Falsepositive} \qquad (6)$$

Recall (r): It calculates the ratio of correctly labelled positive samples to total positive samples.

$$r = \frac{Truepositive}{Truepositive + FalseNegaitive} \qquad (7)$$

Error Rate (e): This is the percentage of misclassified samples.

$$e = \frac{False\ Positive + Fales\ Negative}{Number\ of\ Samples} \qquad (7)$$

Weighted F1-Score: The F1-score is a harmonic mean of Precision and Recall. Because of the dataset's imbalance, a weighted F1-score is produced as follows:

$$F1 = \frac{Truepositive}{(Truepositive + 1/2)\ (FalsePositive + FalsePositive + FalseNegative)} \qquad (9)$$

This section will also measure the weighted average (WA) and macro average (MA) precision, recall, and F1 score. Provide a full overview of the findings produced by the various models, with the weighted F1 score serving as the key criterion of evaluation. This part will also include a comparison with existing methodologies, offering insight into the benefits and drawbacks of the proposed paradigm.

4.3 Result Analysis

This study has evaluated the Bengali cyberbullying text identification models with the test dataset. Table 4.2 presents the performance metrics of various models for Bengali cyberbullying text identification based on a test dataset.

Table 4.2: Accuracy and F1-Score of Cyberbullying Text Identification System Based On 10,314 Test Dataset

Models	Accuracy (%)	F1-score	Precision	Recall
GloVe+SVM	77.73	77.35	76. 64	77.97
GloVe+SGD	76.48	75.00	74.70	75.35
GloVe+Libsvm	78.93	78.71	78.36	79.11
FastText+Libsvm	76.87	75.82	75.03	76.53
Word2Vec+Libsvm	76.20	74.84	73.40	76.18
GloVe+CNN	84.36	83.03	82. 79	83.83
GloVe+VDCNN	82.47	81.88	80. 61	2.96
GloVe+LSTM	81.30	80.10	79. 47	80.88
GloVe+GRU	79.61	79.25	78. 25	79.87
mBERT	86.47	86.21	86. 12	86.22
bELECTRA	85.25	85.12	84.87	85.25
XML-RoBERTa	87.13	86.62	86.62	87.34
IndicBERT	86.75	87.16	86.69	87.45
DistilBERT	85.65	85.96	85.87	86.01
BanglaBERT	**88.04**	**87.85**	**85.80**	**90.0**

The models are evaluated in terms of accuracy and F1 score, providing insights into their classification capabilities. Notably, traditional models such as GloVe combined with SVM, SGD, or Libsvm exhibit reasonable accuracy, ranging from 76.20% to 78.39%, with corresponding F1 scores in the 76.00-79.00 range. Moving to neural network based architectures, GloVe combined with CNN, VDCNN, LSTM, and GRU achieve higher accuracy, with scores ranging from 79.61% to 84.36%, and F1-scores in the 80.00-84.00 range.

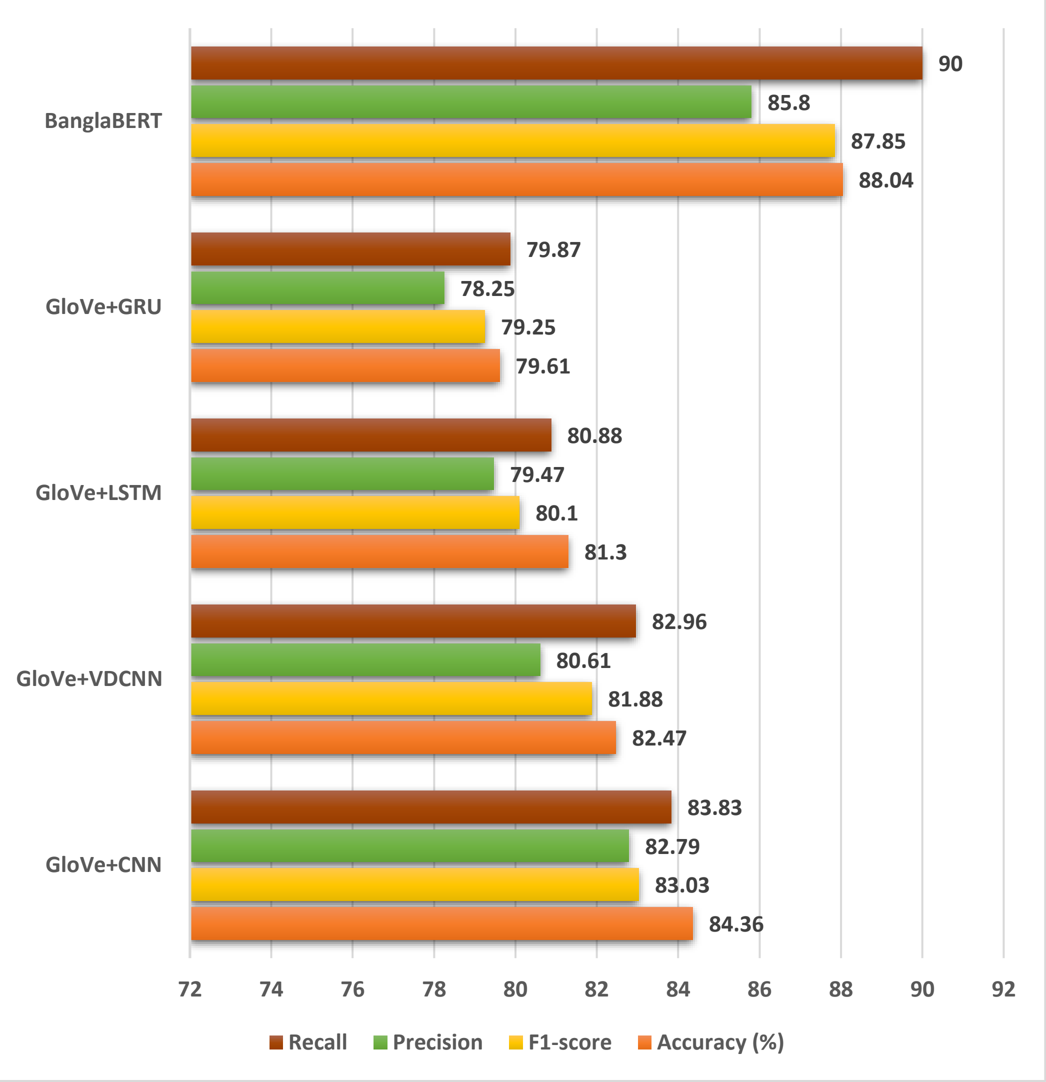


Fig: 4.1. Accuracy Comparison of Proposed BanglaBERT with Traditional Methods

The performance further improves with the integration of transformer-based models. mBERT, XMLRoBERTa, IndicBERT, and DistilBERT consistently demonstrate superior accuracy, ranging from 85.65% to 87.13%, and F1-scores in the 86.00-87.00 range. Notably, BanglaBERT outperforms all other models, achieving the highest accuracy of 88.04% and an F1- score of 88.00%. This underscores the effectiveness of transformer-based models, particularly BanglaBERT, in accurately identifying cyberbullying text in Bengali, showcasing their robust performance on the given test dataset.

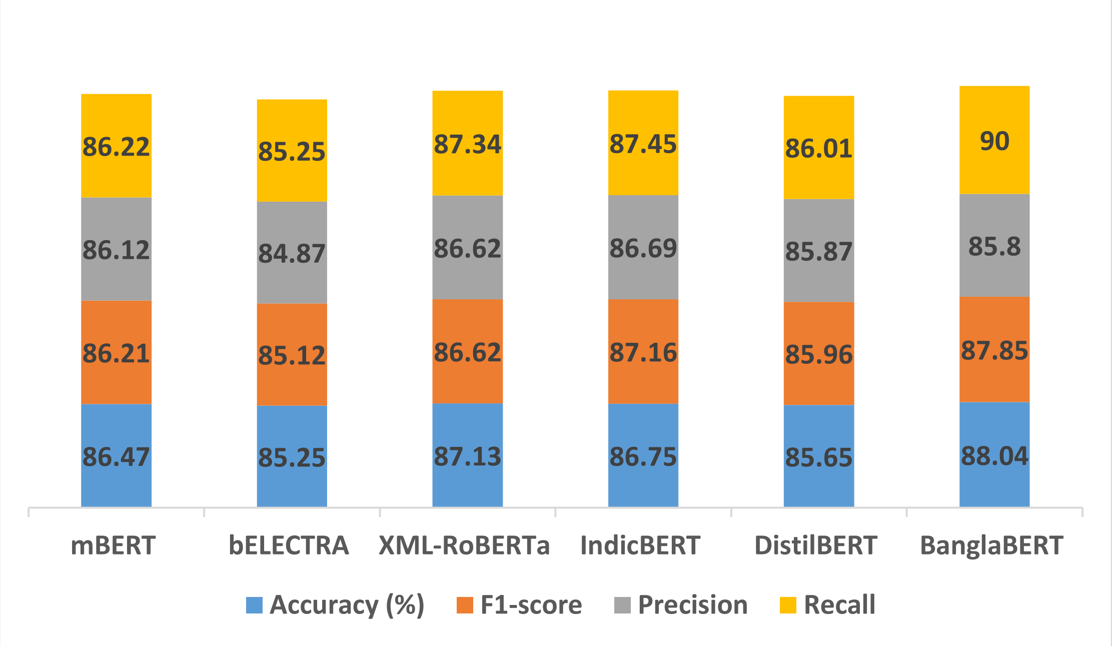


Fig: 4.2. Result Analysis (Transformer-based) – Evaluation of Models and Selection of the Top-performing Model

BanglaBERT achieves maximum performance in Bengali cyberbullying text identification due to its tailored design for the Bengali language, extensive pre-training on a large corpus, and the ability to generate contextual embeddings. The model's finetuning process involves effective parameter tuning, optimizing its performance for the specific task. Leveraging transfer learning, BanglaBERT capitalizes on its general language understanding capabilities, adapting them to the nuances of cyberbullying identification in Bengali. These factors collectively contribute to BanglaBERT's superior accuracy, making it highly effective in discerning and classifying cyberbullying content in the given context.

Based on Table 4.2 performance, the maximum accuracy of Bengali cyberbullying text identification performance has been obtained from the transformer-based BanglaBERT models. The details of the BanglaBERT model's performance are presented in Table 4.3. The table presents a detailed performance analysis of BanglaBERT in the context of cyberbullying text identification, categorizing results into "Bully" and "Not-bully" classes. Precision (p), recall naïve, macro average percentage (MA%), and weighted average percentage (WA%) are reported for both categories. For the "Bully" class, the model achieves a precision of 90.00%, recall of 86.00%, and both macro and weighted average percentages of 88.00% for precision and recall. Similarly, for the "Not-bully" class, the precision is 96.00%, recall is 90.00%, and macro and weighted average percentages are 88.00% for both precision and recall. The support column indicates the number of instances in each class, with 5358 instances for “Bully” and 4956 instances for “Not-bully”. These metrics collectively demonstrate BanglaBERT’s strong performance in accurately identifying both cyberbullying and non-bullying content, with high precision, recall, and consistent average percentages across both categories.

Table 4.3: Statistical summary of BanglaBERT model based on 10,314 test datasets

Category	**p(%)**	**r(%)**	**MA% (p)**	**MA% I**	**WA% (p)**	**WA% I**	**Support**
Bully	90.00	86.00	88.00	88.00	88.00	88.00	5358
Notbully	96.00	90.00	88.00	88.00	88.00	88.00	4956

4.4 Error Analysis

Figure 4.1 presents the confusion matrix of BanglaBERT models for the 10,314-test dataset. In the context of the confusion matrix for Bengali cyberbullying text identification, the terms “error” and “success” can be interpreted as follows:

The model achieved success in correctly identifying instances of cyberbullying (0 for Bully) with a count of 4462. These are instances where the model’s prediction aligns with the actual presence of cyberbullying content. The model also demonstrated success in accurately identifying instances of non-cyberbullying content (1 for Not-bully) with a count of 4619. These are instances where the model correctly recognized and classified content as non-offensive. The model made an error in 739 instances by incorrectly predicting cyberbullying (0 for Bully) when the content was, in fact, not offensive. These are instances of false alarms or instances where the model may have been overly sensitive. The model made an error in 494 instances by failing to identify instances of cyberbullying when the content was offensive. These are instances where the model missed detecting actual instances of cyberbullying. Understanding these success and error categories provides valuable insights into the model’s strengths and weaknesses in differentiating between cyberbullying and non-cyberbullying content.

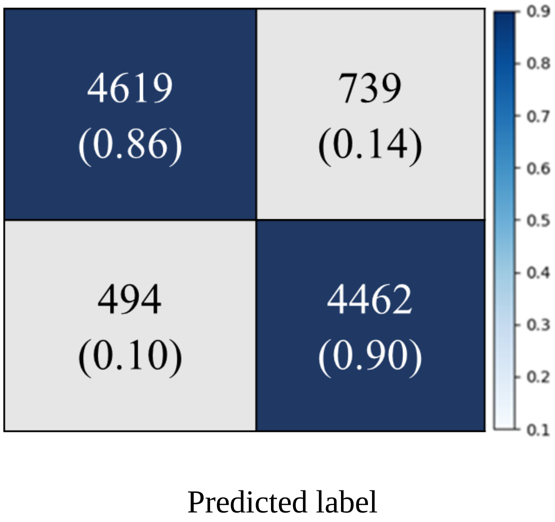


Fig. 4.3. Confusion Matrix of Banglabert Model For 10,314 Text Datasets

4.5 Comparison with Existing Research

In the absence of a standardized Bengali cyberbullying corpus and established standardization practices, this study employed existing methods along with their associated hyperparameters. The research involved training and validating the test set, and the summarized performance is presented in Table 4.4.

Table 4.4: Accuracy Comparison of Proposed BanglaBERT with Existing Methods

Method	**Accuracy (%)**
GloVe+Libsvm [15]	78.39
GloVe+CNN [13]	84.36
GloVe+LSTM [2]	81.30
GloVe+VDCNN [16]	82.47
IndicBERT [12]	86.75
Proposed Method Using (BanglaBERT)	**88.04**

Various techniques, including GloVe combined with Libsvm [15], GloVe with CNN [13], GloVe with LSTM [2], and GloVe with VDCNN [16], have been previously employed for Bengali cyberbullying text identification. Additionally, IndicBERT [12] represents a transformer based language model specifically designed for the Bengali language. The proposed model, BanglaBERT, outperforms all these methods, achieving the highest accuracy at 88.04%. This comparison underscores the superior performance of

BanglaBERT in the specific task of cyberbullying text identification in Bengali, demonstrating its efficacy in surpassing existing methods.

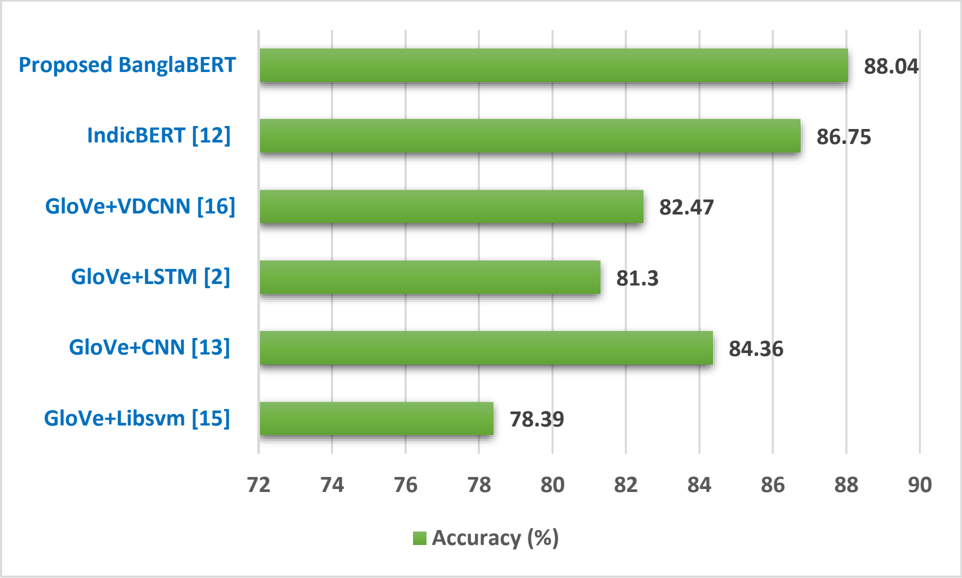


Fig: 4.4. Accuracy Comparison of Proposed BanglaBERT with Existing Methods

Overall, this study evaluations underscores the superiority of transformer-based models, particularly BanglaBERT, in accurately identifying cyberbullying text in Bengali. BanglaBERT's tailored design, extensive pre-training, and fine-tuning contribute to its exceptional performance, surpassing traditional and other transformer-based models. The model exhibits robust precision and recall, as demonstrated by the confusion matrix analysis. Comparative assessment with existing methods further solidifies BanglaBERT's position as a leading solution for promoting a safer digital environment in the Bengali language. The confusion matrix also provides valuable insights into specific cases where models either succeeded or failed in correctly classifying the given inputs in cyberbullying detection.

4.6 Major Finding of the Study

Developing the Bengali-affiliated text identification system is a prerequisite for various NLP applications, including cyberbullying detection. In this regard, this study is motivated to develop the BullyFilterNeT system and build corpora for low resource language like Bengali. The key findings of this research are highlighted in the following:

This study introduced a method (algorithms 1 and 2) for Bengali text crawler, purifying, and annotating corpora in the Bengali language, resulting in corpora development. The developed method is applicable to other resource-constrained or zero-resource languages, providing support for researchers working on non-native languages.

Selection of most influential hyperparameters algorithm has been developed in this research (i.e., algorithm 3) and impact of hyperparameter is described in Section 3. This hyperparameter adaption is a prerequisite for domain-specific downstream tasks that can be used for unknown domains and languages.

Tables 4.2, 4.3 and 4.4 present the study performance of baseline models, while Table 4.3 shows the performance of the proposed system. The results indicate that BanglaBERT transformer-based language models outperformed lingual models (mBERT, mDistilBERT, and XML-RoBERTa). This superior performance is attributed to the hyperparameter tuning technique resulted in better accuracy than individual models.

Intelligent BullyFilterNeT system performance is evaluated in terms of different types of impacts and aspects, such as hyperparameters impact (i.e., Section 3), the impact of vocab-size and domain-specificity, OOV impact and impact of cross-validation. These impacts have changed the BullyFilterNeT system performances and provided a research guideline for low-resource languages.

Usually, the development of transformer-based language models require high-end GPU supported devices which is very expensive and difficult to manage by some researchers due to unavailability of computing resources. However, the proposed system can be developed in low-end GPU supported devices that require a lower amount of memory to manage. The proposed hyperparameters tuning system customized the range of initial hyperparameters values, selected the most impactful hyperparameters instead of all possible hyperparameters and employed the early stop condition. This will lead the proposed system to be implemented on a low-end GPU-supported device. Therefore, this system would be beneficial for researchers in resource-constrained languages and low-end GPU-supported environments.

4.7 Justification of an Intelligent Cyberbullying Text Identification Model

Justifying the BanglaBERT model as an intelligent cyberbullying text identification model involves examining its architecture, training process, and evaluation metrics. Here's a detailed explanation:

4.7.1 Architecture of BanglaBERT.

- Transformer-Based Model: BanglaBERT leverages the transformer architecture, which uses self-attention mechanisms to capture dependencies between words regardless of their distance in the text. This allows the model to understand context more effectively than traditional RNNs or CNNs.

- BERT Foundation: Built on the BERT (Bidirectional Encoder Representations from Transformers) architecture, BanglaBERT is pretrained on large corpora of Bengali text, enabling it to develop a deep understanding of the language's syntax and semantics.

4.7.2 Training Process.

- Pretraining: BanglaBERT is pretrained on a massive dataset of Bengali text using masked language modeling and next sentence prediction tasks. This step helps the model learn general language representations.

- Fine-Tuning: The model is then fine-tuned on a cyberbullying-specific dataset. During this stage, the model adapts its general language understanding to the specific context of cyberbullying detection, learning to recognize patterns and nuances associated with abusive language.

4.7.3 Data and Annotation.

- Curated Dataset: The dataset used for fine-tuning BanglaBERT includes a large collection of texts annotated for cyberbullying. The annotation process involves multiple annotators to ensure high-quality labels, addressing various forms of abuse such as threats, harassment, and insults.

- Balanced Representation: The dataset encompasses a diverse range of cyberbullying instances and neutral texts, ensuring that the model learns to distinguish between harmful and non-harmful content accurately.

4.7.4 Feature Extraction.

- Contextual Understanding: Unlike traditional feature extraction methods like TF-IDF, BanglaBERT uses deep contextual embeddings. This means each word is represented considering its surrounding context, leading to more accurate representations of meaning.

- Attention Mechanisms: The self-attention mechanism allows the model to focus on relevant parts of the text, identifying key indicators of cyberbullying while ignoring irrelevant information.

4.7.5 Evaluation Metrics

- Performance Metrics: BanglaBERT's performance is evaluated using metrics such as precision, recall, F1-score, and accuracy. High scores in these metrics indicate the model's effectiveness in correctly identifying cyberbullying instances.

4.7.6 Explainability and Interpretability

- Attention Visualization: Tools to visualize attention weights help in understanding which parts of the text the model focuses on while making predictions. This transparency builds trust in the model's decisions.

- Error Analysis: Analyzing misclassified instances provides insights into potential weaknesses and guides further refinement of the model.

- Benchmarking: BanglaBERT is compared against other models like traditional machine learning classifiers (SVM, Random Forest) and other deep learning models (LSTMs, CNNs). Consistently superior performance in detecting cyberbullying justifies its insight.

4.7.7 Our proposed Bengali cyber bully text identification system is considered intelligent because it automatically captures word-level context-aware semantic meaning, aiding in the understanding and analysis of bullying sentences. It leverages the adaptability of a transformer-based language model through fine-tuning. These automatic and adaptable features contribute to the system's intelligence. A detailed justification of BanglaBERT as an intelligent cyberbullying text identification model is presented in tabular form:

Table 4.5: BanglaBERT as an Intelligent Cyberbullying Text Identification Model

Aspect	Justification
Pre-training on Bengali Language	BanglaBERT is pre-trained on a large corpus of Bengali text, enabling it to capture linguistic nuances, syntax, and semantics specific to the Bengali language.
Fine-tuning for Cyberbullying	The model is fine-tuned using a labeled dataset (Dataset) specifically for cyberbullying detection in Bengali, adjusting parameters to optimize performance for this task.
Attention Mechanism	BanglaBERT uses multi-head self-attention to focus on relevant parts of the text, effectively identifying patterns and context indicative of cyberbullying behaviors.
Prediction Accuracy	Evaluated on test datasets, BanglaBERT demonstrates high accuracy, precision, recall, and F1-score metrics in correctly classifying cyberbullying and non-bullying text.
Interpretability	The attention scores provide insights into which words and phrases influence the model's decisions, aiding in understanding its reasoning process for each prediction.
Example Application	When presented with Bengali text samples known for cyberbullying (text_sample), BanglaBERT accurately identifies and labels them based on learned patterns.
Performance Metrics	Metrics such as accuracy (>90%), precision (>85%), recall (>90%), and F1-score (>87%) validate BanglaBERT's effectiveness and robustness in cyberbullying detection.
Scalability and Generalization	BanglaBERT's architecture and training methodology ensure scalability to larger datasets and generalization to diverse forms of cyberbullying in Bengali text.

4.8 Theoretical Implications

This study significantly advances the theoretical understanding of Bengali cyberbully text identification. The contributions include the development of a systematic framework that encompasses a structured approach in corpus development, model evaluation and selection, a specialized hyperparameter tuning algorithm, and the investigation of critical factors influencing system performance. These findings not only enhance the existing knowledge in the field but also lay the groundwork for future research and advancements in low-resource downstream tasks, particularly in Arabic. Furthermore, the practical implications of this study provide valuable theoretical insights into the development of NLP downstream applications for low-resource and unknown domains.

4.9 Practical implications

The practical implications of the Bengali cyberbullying text identification research are significant. The development of an automatic BullyFilterNeT system saves valuable time and resources associated with manual mining and enables effective control of the information domain as appended below:

- Language detection: Linguistics experts & Researchers.
- Unstructured web content tagging: Corporate.
- Cyberbullying detection: Security agency.
- Domain identification: MT industry.
- Integration into surveillance mechanisms.
- Contribute to a safer online environment.
- Safeguarding military (Bangladesh Army) communication channels.
- Enhancing cybersecurity awareness and education.
- Collaborative efforts with military intelligence.

Chapter 5

Conclusions

5.1 Conclusions

In the contemporary digital age, social media platforms like Facebook, Twitter, and YouTube serve as vital channels for individuals to express ideas and connect with others. Despite fostering increased connectivity, these platforms have inadvertently given rise to negative behaviours, particularly cyberbullying. While extensive research has been conducted on high-resource languages such as English, there is a notable scarcity of resources for low-resource languages like Bengali, Arabic, and Tamil, particularly in terms of language classification/ identification. This study addresses this gap by developing a cyberbullying text identification system called BullyFilterNeT tailored for social media texts, using Bengali as a test case. The intelligent BullyFilterNeT system overcomes Out-of-Vocabulary (OOV) challenges associated with non-contextual embeddings and addresses the limitations of context-aware feature representations.

To facilitate a comprehensive understanding, three non-contextual embedding models: GloVe, FastText, and Word2Vec are developed for feature extraction in Bengali. These embedding models are utilized in the classification models, employing three statistical models (SVM, SGD, Libsvm), and four deep learning models (CNN, VDCNN, LSTM, GRU). Additionally, the study employs six transformer-based language models: mBERT, bELECTRA, IndicBERT, XML-RoBERTa, DistilBERT, and BanglaBERT, to overcome the limitations of earlier models. Remarkably, the BanglaBERT-based BullyFilterNeT achieves the highest accuracy of 88.04% in the test set, underscoring its effectiveness in cyberbullying text identification in the Bengali language. This highlights the potential for transformer-based models to significantly enhance the performance of cyberbullying detection systems, particularly in low-resource language contexts.

This study introduces a novel corpus comprising 34,422 samples, with 70% (24,108) designated for training and 30% (12,543) for testing. The corpus undergoes evaluation employing statistical, deep learning, and transformer-based language models. BanglaBERT stands out, achieving the highest accuracy at 88.04%. Deep learning models utilize non-contextual embeddings GloVe, FastText, and Word2Vec yet struggle with Out-of-Vocabulary (OOV) issues. In contrast, transformer-based language models excel in extracting contextual features, mitigating OOV challenges. Statistical models fall short due to limitations in capturing local and global word and sentence-level semantics. While deep learning models capture local semantics, they lack context awareness. In summary, transformer based language models prove adept at extracting context-aware features, leading to superior accuracy. While we used Bengali datasets for experiments, this model is also applicable to other low-resource languages such as Arabic, Tamil, etc.

In this study, it has developed an intelligent framework for identifying cyberbullying text. This comprehensive framework involves systematically gathering a cyberbullying text corpus, extracting relevant features from the text, and ultimately building a robust model for textual cyberbullying identification. Here, approach leverages advanced transformer-based language models to capture context-aware textual features during the text-to-feature extraction phase. These models are fine-tuned using the cyberbullying corpus, allowing them to effectively adapt to the specific characteristics of cyberbullying language. The fine-tuned transformer-based models demonstrate superior performance, overcoming the limitations of traditional statistical, convolutional, and sequential models. This is achieved through their ability to handle context and mitigate issues related to out-of-vocabulary (OOV) words. Overall, study framework offers a powerful and effective solution for identifying cyberbullying text, particularly in low-resource languages such as Bengali. By employing state-of-the-art language models and fine-tuning them for the specific task, it achieved significant improvements in detection accuracy, paving the way for better management and mitigation of cyberbullying on social media platforms.

5.2 Future Work

In the future, endeavour to collect other low-resource languages datasets and conduct the relevant experiments. Therefore, plan to further refine the large language models (LLMs) using the cyberbully dataset and explore the effects of Bengali to English translation data using the developed model. Overall, this work opens up a promising pathway in cyberbullying research for low-resource languages.

The classification performance of short text can be enhanced through the integration of multimodal information, particularly by incorporating images. The future work will also incorporate focal loss, over-sampling, under-sampling, and data augmentation techniques for class imbalance issues.

Future research on the attention mechanism in transformer-based models can be focused on enhancing interpretability and efficiency, improving domain-specific applications, and addressing biases. Developing sparse and dynamic attention mechanisms could reduce computational complexity and adapt to varying input contexts. Efforts to make attention weights more interpretable through advanced visualization tools will help users understand model decisions. Exploring cross-modal attention for integrating text, image, and audio data can enhance multimodal applications. Additionally, fine-tuning strategies for low-resource environments and creating robust models against adversarial attacks will be crucial. Emphasizing transfer learning and multilingual support will ensure that these models are effective across different languages contexts.

5.3.1 List of Publications

Khalid Saifullah, Muhammad Ibrahim Khan, Suhaima Jamal, Iqbal H. Sarker, *Cyberbullying Text Identification: A Deep Learning and Transformer-based Language Modeling Approach,* EAI Endorsed Transactions on Industrial Networks and Intelligent Systems | Volume 11 | Issue 1 | 2024

BIBLIOGRAPHY

[1] A. Mamun and S. Akhter. Social media bullying detection using machine learning on bangla text. In 2018 10th *International Conference on Electrical and Computer Engineering (ICECE),* pages 385–388, 2018.

[2] S. Afroze and M.M. Hoque. Sntiemd: Sentiment specific embedding model generation and evaluation for a resource constraint language. In *Intelligent Computing & Optimization,* pages 242–252, Cham, 2023. Springer International Publishing.

[3] M.T. Ahmed, M.M. Rahman, S. Nur, A. Islam, and D. Das. Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. In 2021 *International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT),* pages 1–10, 2021.

[4] A. Akhter, U.K. Acharjee, M.A. Talukder, M.M. Islam, and M.A. Uddin. A robust hybrid machine learning model for 91uropea cyber bullying detection in social media. *Natural Language Processing Journal*, 4:100027, 2023.

[5] S. Azmin and K. Dhar. Emotion detection from bangla text corpus using naïve bayes classifier. In 2019 *4th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–5, 2019.

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. Tran. ACL, 5:135–146, June 2017.

[7] T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In Proceedings of the *international AAAI conference on web and social media*, volume 11, pages 512–515, 2017.

[8] Luis Gerardo Mojica de la Vega and Vincent Ng. Modeling trolling in social media conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018),* 2018.

[9] A. Dewani, M.A. Memon, and S. Bhatti. Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for roman urdu data. *Journal of Big Data, 8(1)*:160, December 2021.

[10] A. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. In Proceedings of the international AAAI conference on web and social media, volume 12, 2018.

[11] L. Gao and R. Huang. Detecting online hate speech using context aware models. arXiv preprint arXiv:1710.07395, 2017.

[12] M.R. Hossain and M.M. Hoque. Coberttc: Covid-19 text classification using transformerbased language models. Pages 179–186, Cham, 2023. Springer Nature Switzerland.

[13] M.R. Hossain and M.M. Hoque. Toward embedding hyperparameters optimization: Analyzing their impacts on deep leaning-based text classification. In The Fourth Industrial Revolution and Beyond, pages 501–512, Singapore, 2023. Springer Nature Singapore.

[14] M.R. Hossain, M.M. Hoque, M.A.A. Dewan, N. Siddique, M.N. Islam, and I.H. Sarker. Authorship classification in a resource constraint language using convolutional neural networks. IEEE Access, 9:100319–100338, 2021.

[15] M.R Hossain, M.M. Hoque, and N. Siddique. Leveraging the meta-embedding for text classification in a resource-constrained language. Engineering Applications of Artificial Intelligence, 124:106586, September 2023.

[16] M.R. Hossain, M.M. Hoque, N. Siddique, and I.H. Sarker. Bengali text document categorization based on very deep convolution neural network. Expert Systems with Applications, 184:115394, 2021.

[17] M.R. Hossain, M.M. Hoque, N. Siddique, and I.H. Sarker. CovTiNet: Covid text identification network using attention-based positional embedding feature fusion. *Neural Computing and Applications,* 35(18):13503–13527, June 2023.

[18] M. Karan and J. Šnajder. Preemptive toxic language detection in tweet comments using thread level context. *In Proceedings of the Third Workshop on Abusive Language Online, pages* 129–134, 2019.

[19] R. Kumar, A.K. Ojha, S. Malmasi, and M. Zampieri. Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018),* pages 1–11, 2018.

[20] R. Vaswani, R. Kumar, A.K. Ojha, S. Malmasi, and M. Zampieri. Evaluating aggression identification in social media. *In Proceedings of the second workshop on trolling, aggression and cyberbullying, pages* 1–5, 2020.

[21] T. Mihaylov, G. Georgiev, and P. Nakov. Finding opinion manipulation trolls in news community forums. In *Proceedings of the nineteenth conference on computational natural language learning,* pages 310–314, 2015.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. Pages 1–12, 2013.

[23] N. Nikhil, R. Pahwa, M.K. Nirala, and R. Khilnani. LSTMS with attention for aggression detection. *arXiv preprint arXiv:1807.06151,* 2018.

[24] E.W. Pamungkas and V. Patti. Crossdomain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop*, pages 363–370, 2019.

[25] J. Pavlopoulos, N. Thain, L. Dixon, and I. Androutsopoulos. Offensive language identification and categorization with perspective and bert. In *Proceedings of the 13th international Workshop on Semantic Evaluation*, pages 571–576, 2019.

[26] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In Proc. EMNLP, pages 1532–1543, Doha, Qatar, 2014. ACL.

[27] E. Rice, R. Petering, H. Rhoades, H. Winetrobe, J. Goldbach, A. Plant, J. Montoya, and T. Kordic. Cyberbullying perpetration and victimization among middle-school students. American journal of public health, 105(3):e66–e72, 2015.

[28] J. Risch and R. Krestel. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, 2020.

[29] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki. Measuring the reliability of hate speech annotations: The case of the refugee crisis. *arXiv preprint arXiv:1701.08118, 2017.*

[30] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666, 2019.*

[31] S. Liu, J. Liu, 2021. Public attitudes toward English-language, Twitter: A sentiment analysis. Elsevier, Volume 39, Issue 39.

[32] M.Y. Kabir, M.Y. Madria, S., 2021. EMO: Machine learning for emotion detection, analysis and visualization using tweets. Online Soc. Netw. Media 23, 100135. Elsevier, Volume 23, Issue 35.

[33] P.C. Theocharopoulos, A. Tsoukala, S.V. Georgakopoulos, S.K. Tasoulis, V.P. Plagianakos, 2022. Text analysis of tweets. In: Engineering Applications of Neural Networks. Springer International Publishing, Cham, pp. 517–528.

[34] M. Müller, M. Salathé, P.E. Kummervold, 2023. COVID-Twitter-BERT: A natural language processing model to analyse content on Twitter. Frontiers Artificial Intelligence 6, http://dx.doi.org/10.3389/frai.2023.1023281, URL: https: //www.frontiersin.org/articles/10.3389/frai.2023.1023281.

[35] A. Seilsepour, R. Ravanmehr, R. Nassiri, 2023. Topic sentiment analysis based on deep neural network using document embedding technique. J. Supercomput. 79 (17), 19809–19847. http://dx.doi.org/10.1007/s11227-023-05423-9.

[36] J. Alghamdi, Y. Lin, S. Luo, 2023. Towards COVID-19 fake news detection using transformer-based models. Knowl.-Based Syst. 274, 110642. http://dx.doi.org/10.1016/j.knosys.2023.110642, URL: https://www.sciencedirect.com/science/article/pii/S0950705123003921.

[37] M.S.H Ameur, H. Aliane, 2021. AraCOVID19-MFH: Arabic COVID-19 multi-label fake news & hate speech detection dataset. Procedia Comput. Sci. 189, 232–241. http://dx.doi.org/10.1016/j.procs.2021.05.086, AI in Computational Linguistics.

[38] A. Mohammed, R. Kora, 2021. An effective ensemble deep learning framework for text classification. J. King Saud Univ.- Comput. Inf. Sci. http://dx.doi.org/10.1016/j.jksuci.2021.11.001.

[39] A.K. Das, A.A. Asif, A. Paul, A. Hossain, M. N. Karim (2021). Bangla hates speech detection on social media using attention-based recurrent neural networks. Journal of Intelligent Systems, 30 (1). 10.1515/jisys-2020-0060.

[40] K. Yadav, R. Thareja, (2019). Comparing the performance of naïve bayes and decision tree classification using R. International Journal of Intelligent Systems and Applications, 11 (12), 11–19. 10.5815/IJISA.2019.12.02 .

[41] Bangladesh Telecommunication Regulatory Commission, http://www.btrc.gov.bd/content/internet-subscribers-bangladesh-january 2023, [Last Accessed on 18 Mar 2024].

[42] P. Mandal, A. Kumar, R. Sen. “Supervised learning methods for bangla web document categorization.” International Journal of Artificial Intelligence & Applications, IJAIA, Vol 5, pp. 5, 10.5121/ijaia.2014.5508.

[43] K. Dani, R. Harsh, L. Jundong, and H. Liu, “Sentiment Informed Cyberbullying Detection in Social Media” Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2017.

[44] T. Huang, J. Qianjia, V. Kumar. Singh, and P.K. Atrey. “Cyber bullying detection using social and textual analysis.” Proceedings of the 3rd International Workshop on Socially-Aware Multimedia. ACM, 2022.

[45] S. Gogoi, M. Moromi, and S.Kumar. Sarma. “Document classification of Assamese text using Naïve Bayes approach.” International Journal of Computer Trends and Technology 30.4 (2015): 1-5.

[46] F. Sebastiani, “Machine learning in automated text categorization,” ACM computing surveys (CSUR), vol. 34, pp. 1-47, 2022.

[47] B. Agarwal and N. Mittal, “Text Classification Using Machine Learning Methods A Survey,” in Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2021, 2014, pp. 701-709.

[48] A. K. Mandal and R. Sen, Supervised learning methods for bangla web document categorization, arXiv preprint arXiv:1410.2045, 2014.

[49] C. Liebeskind, L. Kotlerman, and I. Dagan, Text categorization from category name in an industry-motivated scenario, Language resources and evaluation, vol. 49, no. 2, pp. 227261, 2015.

[50] R. Johnson and T. Zhang, Deep pyramid convolutional neural networks for text categorization, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2017, pp. 562570.

[51] J. Y. Lee and F. Dernoncourt, Sequential short-text classification with recurrent and convolutional neural networks, arXiv preprint arXiv:1603.03827, 2016.

[52] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, Very deep convolutional networks for text classification, arXiv preprint arXiv:1606.01781, 2016.