AI based Multilingual Cyberbullying Detection System

by

Jakaria Kamal
Maysha Samia Masud
Sakib-E-Saqlain

BACHELOR OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

Department of Information and Communication Technology
Faculty of Science and Technology
Bangladesh University of Professionals

August 2025

APPROVAL

The thesis titled *Multilingual Cyberbullying Detection System Using AI-Based Language Identification and Language-Specific Models for Bangla, Banglish, and English* submitted by Jakaria Kamal, Roll No: 2154901124, Maysha Samia Masud, Roll No: 2154901053, and Sakib-E-Saqlain, Roll No: 2054901126, Session: 2020-2021 has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Information and Communication Engineering.

(Supervisor)
Saeed Mahmud Ullah
Professor
Department of Electrical and Electronics Engineering
University of Dhaka

DECLARATION

We hereby declare that this thesis is our original work, and we have written it in its entirety. We have duly acknowledged all the sources of information which have been used in the thesis. The thesis (fully or partially) has not been submitted for any degree or diploma in any university or institute previously.

Jakaria Kamal
ID: 2154901124
Department of Information and Communication Technology
Faculty of Science and Technology
Bangladesh University of Professionals
August, 2025

Maysha Samia Masud
ID: 2154901053
Department of Information and Communication Technology
Faculty of Science and Technology
Bangladesh University of Professionals
August, 2025

Sakib E Saqlain
ID: 2054901126
Department of Information and Communication Technology
Faculty of Science and Technology
Bangladesh University of Professionals
August, 2025

ACKNOWLEDGEMENTS

First and foremost, praises and gratitude to the Almighty for His blessings during our study work to successfully complete the thesis. We want to give our sincere and honest appreciation for giving us the opportunity to study in this field and provide invaluable guidance during the thesis to our thesis Supervisor, Saeed Mahmud Ullah, Professor, Department of Electrical and Electronics Engineering, University of Dhaka. His energy, vision, candor, and motivation greatly inspired us. He has imparted knowledge to us on how to conduct research and craft a thesis that is as clear as possible. The study conducted under his direction was a true honour and privilege.

We would like to express our gratitude to all of the Bangladesh University of Professionals' Department of Information and Communication Engineering faculty members for sharing their knowledge and expertise with us during the course of our studies, enabling us to stay focused on the thesis. Our deepest gratitude goes to our parents for their unwavering commitment, prayer, and selflessness in educating and preparing us for the future. Lastly, we would want to express our gratitude to all of our classmates who assisted us, either directly or indirectly, to finish the research.

ABSTRACT

The rise of online interactions has given way to an alarming increase in cyberbullying, necessitating advanced systems for detection and prevention. This paper presents a novel approach to multilingual cyberbullying and hate speech detection across three languages: English, Banglish (transliterated Bengali using English script), and Bengali. With the increasing prevalence of code-mixing—where multiple languages are used within a single text—this study addresses the challenges posed by diverse linguistic expressions in online communication. To tackle this, we propose a robust system that incorporates a language identifier, which preprocesses the input text and routes it to the appropriate language-specific model, ensuring more accurate and context-aware detection of harmful content. Using publicly available datasets from platforms like Kaggle and Mendeley, we developed separate models for each language, ensuring the system is tailored to the linguistic nuances of Bangla, English, and Banglish. Our methodology combines traditional machine learning models like Logistic Regression and SVM, with advanced deep learning models such as LSTM and BiLSTM, as well as cutting-edge transformer models like BERT and mBERT. This combination enables the system to effectively balance precision and recall, providing a comprehensive and efficient solution for cyberbullying detection.The system is rigorously evaluated using standard performance metrics—accuracy, precision, recall, F1-score, and ROC AUC—and the results demonstrate its exceptional effectiveness in handling multilingual and code-mixed content. Our approach not only outperforms conventional methods but also offers a scalable, efficient solution for real-time cyberbullying detection, making it suitable for deployment in large-scale online platforms. This work is a significant step forward in improving the safety and well-being of online communities, particularly by addressing the unique challenges of multilingual environments. Our system has the potential to provide more accurate, real-time detection of harmful content across diverse languages and cultures, ultimately paving the way for safer online spaces. By highlighting the importance of context-aware and language-specific models, we offer a solution that can adapt to the complexities of modern digital communication, making it more effective in combating cyberbullying globally.

Keywords: Cyberbullying Detection, Machine Learning, Deep Learning, Transformer Models, BanglaBERT, Multimodal Analysis, Social Media, NLP.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

Bi-LSTM	Bidirectional Long Short-Term Memory
Stack-LSTM	Stacked Long Short-Term Memory
SVM	Support Vector Machine
GRU	Gated Recurrent Unit
NumPy	Numerical Python
ReLU	Rectified Linear Unit
LR	Logistic Regression
RF	Random Forest
DT	Decision Tree
XGBoost	Extreme Gradient Boosting
Bi-GRU	Bidirectional Gated Recurrent Unit
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
F1-score	F1-measure
ROC AUC	Receiver Operating Characteristic - Area Under Curve
TF-IDF	Term Frequency-Inverse Document Frequency
BERT	Bidirectional Encoder Representations from Transformers
mBERT	Multilingual BERT
XLM-RoBERTa	Cross-lingual Roberta

Chapter 1

INTRODUCTION

1.1 Overview

Cyberbullying and hate speech detection has become a critical area of research due to the increasing prevalence of these harmful behaviors across social media platforms. The complexity in detecting such content arises from the diverse linguistic expressions, multilingual environments, and code-mixed languages often used in online interactions. Traditional detection systems have largely depended on machine learning (ML), deep learning (DL), and transformer-based models, which have demonstrated impressive performance in various text classification tasks. However, these models, particularly transformers like BERT and XLM-R, often come with high computational costs, which make them unsuitable for real-time applications or use on devices with limited computational resources. Additionally, many existing models face difficulties when handling code-mixed languages, such as Banglish (Bengali written in English script), which is common in online communities, especially in South Asia.

This paper proposes a novel solution to these challenges by developing a multilingual cyberbullying and hate speech detection system across three languages: English, Banglish, and Bengali. A key aspect of the proposed system is the language identifier model, which preprocesses the input and directs it to the relevant language-specific model for more accurate processing. This approach not only enhances the system's accuracy but also ensures that language-specific nuances are properly addressed. By focusing on lightweight models, our system is able to achieve high accuracy while minimizing resource consumption, making it more practical for deployment in resource-constrained environments. Our methodology leverages traditional machine learning models (e.g., Logistic Regression, SVM), deep learning models (e.g., LSTM, BiLSTM), and transformer models (e.g., BERT, XLM-R) to provide a robust solution for detecting cyberbullying across different languages and contexts. The results of our experiments demonstrate that our system delivers strong performance, particularly in the Banglish context, where traditional models have struggled [5].

1.2 Motivation of the Study

The motivation for this study stems from the limitations observed in previous cyberbullying detection models, particularly in terms of their resource-intensive nature. Transformer-based models like BERT and XLM-R, while achieving state-of-the-art results, require substantial computational resources, which makes them unsuitable for real-time applications or deployment on devices with limited resources. Furthermore, these models often fail to effectively handle code-mixed languages such as Banglish, which are common in social media platforms. Banglish—a hybrid language combining Bengali and English—presents unique challenges due to inconsistent spelling and the blending of two languages, making it difficult for existing models to capture the full linguistic context.

Our goal is to develop a lightweight solution that provides high accuracy while reducing computational overhead. This will ensure the scalability and real-time capability of the system, making it suitable for deployment in a variety of environments. By designing three language-specific models—one for English, one for Bangla, and another for Banglish—we aim to overcome the challenges posed by code-mixed data and multilingual text. The inclusion of a language identifier to route input to the correct model is a key innovation that allows our system to perform well across different linguistic contexts, offering a more efficient and accurate solution compared to traditional models. This lightweight approach ensures that our system can run efficiently without the need for large-scale computational resources, thus making it suitable for large-scale deployment and real-time detection of cyberbullying content [6].

1.3 Problem Statement

Despite the advances made in cyberbullying detection, existing models still face significant limitations that hinder their practical application. Transformer-based models like BERT and XLM-R have achieved high accuracy in text classification tasks, but they are also computationally expensive, making them impractical for real-time applications or environments with limited resources. In addition, these models struggle with multilingual and code-mixed text, which is a common feature in online platforms. Banglish, a blend of Bengali and English, poses a unique challenge for existing models due to the lack of standardized spelling conventions and the inherent complexities of code-switching between languages. Moreover, many studies have focused on monolingual datasets, which limits the generalizability of their models in multilingual environments.

Another key issue is overfitting, which occurs when models, particularly deep learning-based ones, have a high number of parameters relative to the available data, leading to poor generalization on unseen data. Existing models also lack a real-time deployment evaluation, making them unsuitable for large-scale applications where quick detection of harmful content is essential. Our work aims to address these limitations by developing lightweight, language-specific models for Bangla, English, and Banglish, enabling high accuracy while reducing the computational overhead of previous approaches. By using these language-specific models, we can improve the performance of multilingual and code-mixed language detection without the need for resource-heavy transformer models, thus making the system scalable and real-time capable [16].

1.4 Objective of the Study

The primary goal of this study is to develop an efficient, scalable, and accurate multilingual cyberbullying detection system that addresses the limitations of previous works, especially concerning resource consumption, real-time applicability, and the challenges posed by multilingual and code-mixed data. The specific objectives of this study are as follows:

1. **To develop a language identifier model:** Accurately classify text as Bangla, Banglish, or English and route it to the appropriate language-specific model.
2. **To create three language-specific models for Bangla, English, and Banglish:** Develop distinct models for each language to capture unique linguistic features and provide contextual detection of harmful content.
3. **To utilize lightweight models for better efficiency:** Show that lightweight models can achieve high accuracy while reducing computational overhead, making them suitable for real-time detection.
4. **To address the challenges posed by code-mixed data:** Specifically improve detection in Banglish, a mixture of Bengali and English, by creating a dedicated model for transliterated text.
5. **To evaluate the system using standard performance metrics:** Use metrics such as accuracy, precision, recall, F1-score, and ROC AUC to assess the models' performance in detecting cyberbullying.
6. **To ensure real-time applicability and efficiency in resource-constrained environments:** Ensure the system operates effectively in real-time and on platforms with limited computational resources.
7. **To compare the lightweight models with existing models:** Compare the performance of the lightweight models with BERT and XGBoost to demonstrate the advantages of using lightweight models for multilingual detection.
8. **To develop a scalable and efficient solution for multilingual cyberbullying detection:** Create a system capable of detecting cyberbullying across multiple languages and running efficiently on large-scale platforms.

These objectives aim to provide a resource-efficient, accurate, and scalable solution for multilingual cyberbullying detection, particularly focusing on real-time applicability.

1.5 Contribution of the Study

This study makes several important contributions to the field of cyberbullying detection:

First, our research introduces a multilingual detection system that handles English, Banglish, and Bangla content effectively. Unlike previous systems that either focused on monolingual datasets or struggled with code-mixed data, our approach utilizes three language-specific models, significantly improving detection accuracy across different linguistic contexts.

Second, we propose the use of lightweight models that achieve high performance while minimizing computational overhead, in contrast to traditional transformer-based models such as BERT and XLM-R, which are computationally expensive. This makes our solution scalable and suitable for real-time deployment.

Third, our system excels at detecting cyberbullying in code-mixed languages like Banglish, an area where existing models have performed poorly. By leveraging the strengths of language-specific models, we offer a more efficient and effective solution to the challenges posed by multilingual and code-mixed content.

Finally, the study demonstrates that, while models like XGBoost and BERT perform well, our lightweight models deliver comparable or better results with significantly less resource consumption, making them more suitable for large-scale deployment in real-time applications [5, 6].

1.6 Outline of the Thesis

There are six chapters in this thesis. In the following sections, the research methodology, experimental setup, results, and analysis will be discussed in detail, offering a comprehensive overview of the study and its implications. The remainder of the thesis is structured as follows:

- **Chapter 1:** Provides an introduction to the issue of cyberbullying, its societal impact, the motivation behind the research, and outlines the objectives, problem statement, and overall contributions of the study.
- **Chapter 2:** Discusses the background and related works, offering a review of existing literature on cyberbullying detection using machine learning, deep learning, and transformer-based approaches. It includes studies conducted on both Bangla and multilingual datasets.
- **Chapter 3:** Details the tools, techniques, models, and datasets employed in the implementation of this study. It explains the design of hybrid frameworks and the use of augmentation, preprocessing, and feature extraction methods.
- **Chapter 4:** Presents a comparative analysis of the implemented models, including performance metrics and evaluation across different data types and architectures.
- **Chapter 5:** Discusses the experimental results in depth, highlighting key findings, limitations, and the effectiveness of each model in real-world cyberbullying detection scenarios.

- **Chapter 6:** Concludes the research by summarizing the major outcomes, outlining the limitations of the current study, and proposing potential directions for future work.

Chapter 2

Literature Review

2.1 Overview of Cyberbullying Detection

Cyberbullying detection has become a crucial area of research, particularly with the rise of social media platforms where abusive behavior often occurs in online interactions. Various techniques have been proposed for detecting cyberbullying, with machine learning (ML), deep learning (DL), and transformer-based models emerging as the most prominent methods.

In the case of machine learning, models like Decision Trees, Random Forests, and Support Vector Machines (SVM) are commonly used, typically relying on manually engineered features like n-grams and syntactic patterns for classification tasks. However, these models often struggle with larger, unstructured datasets common in social media and fail to capture the complex linguistic features associated with cyberbullying.

Deep learning models, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and BiLSTM, have shown superior performance by automatically extracting features and identifying deeper semantic patterns in the text. For example, Farjana Akter et al. [3](2025) used CNN and LSTM models to detect cyberbullying in Bangla social media comments, achieving accuracy levels of 99.80% with LSTM and 99% with CNN.

More recently, transformer-based models like BERT, mBERT, and XLM-R have dominated the field due to their contextual understanding of language. These models have been fine-tuned on task-specific datasets, making them highly effective for tasks such as cyberbullying detection. Transformer models are especially advantageous for multilingual tasks, where they can handle text in different languages with high accuracy. For instance, Khalid Saifullah et al.[16] (2024) used BanglaBERT, a fine-tuned transformer model, and achieved an accuracy of 88.04% for detecting cyberbullying in Bengali texts. However, the use of multilingual and code-mixed data (e.g., Banglish) remains a challenge for these models, often limiting their effectiveness in real-world applications.

Furthermore, multimodal models, which combine text, image, and audio data, have been explored to increase accuracy. For example, Neha Minder Singh et al. [18](2023) developed a multimodal framework using text, audio, and image data, achieving an impressive 98.23% accuracy for cyberbullying detection.

2.2 Conventional Cyberbullying Detection Approaches

Traditionally, cyberbullying detection relied on feature-based machine learning models, where researchers would manually extract features from text data, such as word frequencies, sentiment scores, and specific linguistic patterns associated with bullying behavior. Models like SVM, Naive Bayes, and Logistic Regression were commonly used for binary classification tasks (bullying vs. non-bullying).

These models typically performed well on smaller datasets, but their performance significantly dropped as the volume of data increased or when the text became more complex. Furthermore, conventional methods did not account for the subtleties in code-mixed languages (like Banglish) or multilingual data, leading to significant limitations in real-world applications.

Early deep learning methods, such as CNN and RNN, provided some improvements by leveraging their ability to learn hierarchical features and handle large-scale unstructured data. However, they still lacked the contextual understanding that transformer-based models now provide. The shift towards transformers marked a significant leap in the field, allowing models to capture the nuances of language and context more effectively, especially for cyberbullying detection. Despite these advancements, many studies continued to focus on text-only data, neglecting the multimodal nature of real-world social media interactions, where posts often combine text with images, videos, or audio.

2.3 Existing Work

There has been considerable progress in cyberbullying detection research, with numerous studies applying machine learning, deep learning, and transformer models. For instance, Iurii Krak et al.[5] (2025) combined BERT with visual explanation tools like LIME to detect cyberbullying, achieving an accuracy of 95.65%. This method focused on English text and showed a significant improvement over previous models by incorporating explainability features for end-users. In contrast, Farjana Akter et al.[3](2025) built a Bangla cyberbullying detection system, using CNN and LSTM, achieving near-perfect accuracy with LSTM (99.80%) and CNN (99%). Khalid Saifullah et al.[16](2024) also worked on cyberbullying detection using BanglaBERT, achieving an 88.04% accuracy on a dataset of Bengali social media comments, highlighting the model's success in low-resource languages. However, while transformer-based models like XLM-R have shown superior performance in multilingual contexts (with an accuracy of 82.61% for Bangla), they still struggle with the intricacies of code-mixed languages, where multiple languages are interspersed in the same text. Neha Minder Singh et al.[18] (2023) presented a multi-modal detection framework that combined text, audio, and image data, achieving 98.23% accuracy. However, this approach's resource-intensive nature makes it impractical for real-time applications or environments with limited computational power. Existing studies have mainly focused on single-language datasets or multilingual models that often fail in code-mixed data scenarios, such as Banglish, which is prevalent in Bengali-English social media posts.

Paper Title	**Year**	**Author(s) / Dataset**	**Model Name**	**Result**
Method for neural network cyberbullying detection [5]	2025	Iurii Krak, Olena Sobko, Maryna Molchanova / Cyberbullying Classification (Kaggle)	BERT	Accuracy: 95.65%, Precision: 96.37%, Recall: 95.65%
Comprehensive Framework for Multimodal Hate Speech Detectio[12]n	2025	R. Prabhu, V. Seethalakshmi / HateXplain, Kaggle Toxic Comments	CNN, RNN/LSTM, BERT	Accuracy: 98.53%
Cyberbullying Detection on Social Media Platforms[3]	2025	Farjana Akter et al. / Bangla social media (10,254 comments)	CNN, LSTM	LSTM: 99.80%, CNN: 99%
Cyberbullying Detection Using Decision Fusion Classifier[18]	2023	Neha Minder Singh et al. / Text, Audio, Image (unspecified)	BiLSTM-AHCNet, AEB0, MMDFC	Accuracy: 98.23%, F1: 98.22%
Cyberbullying Detection for Low-resource Languages[7]	2023	Tanjim Mahmud et al. / Chittagonian dialect	BanglaBERT, mBERT	No fixed metrics, review paper
Cyberbullying Detection Using Deep Neural Networks[1]	2022	Md Faisal Ahmed et al. / 44,001 Bangla comments	Deep Neural Networks	Binary accuracy: 87.91%, Multiclass: 85%
Cyberbullying Text Identification[16]	2024	Khalid Saifullah et al. / 34,422 Bengali texts	Hybrid: ML, DL, Transformers	Accuracy: 88.04%
Cyberbullying Detection Using Machine Learning and Deep Learning[4]	2023	Aljwharah Alabdulwahab et al. / 47,692 tweets	CNN-LSTM	96% accuracy
An Application to Detect Cyberbullying Using ML and DL[13]	2022	Mitushi Raj et al. / English, Hindi, Hinglish	CNN-BiLSTM	Up to 94.94% accuracy
Cyberbullying Detection Based on Hybrid Ensemble Method[2]	2023	Md. Tofael Ahmed et al. / Bangla social media comments	Hybrid ensemble (LR, KNN, LSTM)	Accuracy: 85%
Cyberbullying Detection of Resource-Constrained Language[17]	2024	Syed Sihab-Us-Sakib et al. / 2,751 Bangla texts	SVM, RF, m-BERT	XLM-R: 82.61%
A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection[9]	2020	Amgad Muneer et al. / 37,373 tweets	Logistic Regression, SVM, Naive Bayes	Logistic Regression: 92% (Formspring)
EnsMulHateCyb: Multilingual hate speech detection[6]	2024	Esshaan Mahajan et al. / Multilingual (English, Spanish, Hindi)	XLM-RoBERTa	Accuracy: 90.7%, Recall: 0.89
Multimodal Hate Speech Detection[15]	2025	Furqan Khan et al. / Urdu-English dataset (11,000 tweets)	BiLSTM + EfficientNetB1	Urdu: 81.5%, English: 75.3%
Deep Learning Based Cyberbullying Detection in Bangla[11]	2024	Sristy Shidul Nath et al. / 12,282 Bangla social media comments	BiLSTM	Accuracy: 95.08%, F1: 95.23%

Table 2.1: Summary of existing cyberbullying detection studies.

2.4 Limitations of Existing Work

While previous work has made significant strides in cyberbullying detection, there are notable limitations that remain. Many studies, such as Farjana Akter et al. [3](2025), focused solely on monolingual datasets, primarily in Bangla or English, limiting applicability in multilingual settings. Transformer-based models like XLM-R and mBERT struggle with code-mixed languages, leading to lower accuracy. They are also resource-intensive, making them unsuitable for real-time applications.

Multimodal approaches, while promising, face challenges in integrating data types effectively and require high computational power, as in Neha Minder Singh et al.[18] (2023). Additionally, many studies lack real-time deployment validation.

By addressing these gaps, your approach of using language-specific models (Bangla, Banglish, English) improves detection accuracy in mixed-language contexts and offers more efficient real-time deployment.

Paper Title	**Year**	**Model Name**	**Contributions**	**Limitations**
Method for neural network cyberbullying detection[5]	2025	BERT	Combines BERT with visual explanations for interpretability	Focused on English only, no real-time deployment
Comprehensive Framework for Multi-modal Hate Speech Detection[12]	2025	CNN, RNN/LSTM, BERT	Multi-modal framework, attention-based fusion for better interpretability	High resource demand, limited exploration of low-resource languages
Cyberbullying Detection on Social Media Platforms[3]	2025	CNN, LSTM	Created Bangla cyberbullying dataset, compared classifiers	No use of transformers, struggles with future pattern shifts
Cyberbullying Detection Using Decision Fusion Classifier[18]	2023	BiLSTM-AHCNet, AEB0, MMDFC	Multi-modal detection framework combining text, audio, and image	Lack of dataset size details, complex fusion
Cyberbullying Detection for Low-resource Languages[7]	2023	BanglaBERT, mBERT	Released dataset, surveyed low-resource challenges	Limited benchmarks, no fixed metrics
Cyberbullying Detection Using Deep Neural Networks[1]	2022	Deep Neural Networks	Proposed binary and multiclass classification models for Bangla	Limited to Bangla Facebook comments, no real-time validation
Cyberbullying Text Identification[16]	2024	Hybrid (ML, DL, Transformers)	Compared 13 models, proved BanglaBERT's efficacy	Resource-heavy transformer fine-tuning
Cyberbullying Detection Using Machine Learning and Deep Learning[4]	2023	CNN-LSTM	Combined ML and DL for high accuracy	Limited to English, no multilingual evaluation
An Application to Detect Cyberbullying Using ML and DL[13]	2022	CNN-BiLSTM	Multilingual real-time detection framework	Dataset imbalance, no per-language breakdown
Cyberbullying Detection Based on Hybrid Ensemble Method[2]	2023	Hybrid ensemble	Proposed hybrid approach combining ML and DL	Limited diversity in dataset, no external validation
Cyberbullying Detection of Resource-Constrained Language[17]	2024	XLM-R, m-BERT	Released Bangla dataset, benchmarked transformers	Small dataset, high resource demand
A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection[9]	2020	Logistic Regression, SVM	Compared 7 ML classifiers on Twitter data	Focused on English, limited to textual features
EnsMulHateCyb: Multilingual hate speech detection[6]	2024	XLM-RoBERTa	Multilingual transformer model for hate speech detection	Poor performance in code-mixed languages
Multimodal Hate Speech Detection[15]	2025	BiLSTM + EfficientNetB1	Developed image-text fusion model	Low quality image-text fusion, small subset
Deep Learning Based Cyberbullying Detection in Bangla[11]	2024	BiLSTM	Optimized deep learning framework for Bangla	Overfitting risk, malformed Bangla words

Table 2.2: Summary of contributions and limitations of existing work.

2.5 Summary

This literature review discusses various approaches to cyberbullying detection, focusing on ML, DL, and transformer-based models. While traditional ML models struggled with complex, unstructured, and multilingual data, DL models like LSTM and CNN achieved higher accuracy (up to 99.80%). Transformer models improved contextual understanding but face challenges with code-mixed languages and are resource-intensive. Multimodal approaches have shown promise but are computationally expensive. Existing research highlights limitations in handling code-mixed languages and real-time deployment, which your work addresses by using language-specific models for Bangla, Banglish, and English, offering a more efficient and accurate solution.

Chapter 3

BACKGROUND TOOLS AND TECHNOLOGIES

3.1 Overview

This chapter gives an overview of the essential components required to use our system. It covers key concepts in machine learning and the algorithms implemented in our system.

3.2 Tools

This section provides a brief understanding of the tools used for the work.

3.2.1 Machine Learning Libraries

Our system utilizes several Python machine learning libraries, including NumPy, Pandas, Matplotlib, TensorFlow, Keras, and Scikit-learn (Sklearn), which are essential for data manipulation, visualization, and model development.

NumPy

Python numerical computations are built on top of NumPy, short for Numerical Python. It contains strong object derivatives such multi-dimensional arrays, masked arrays, and matrices. Additionally, the library offers functions for I/O operations, logical operations, statistical functions, linear algebra, sorting, shape manipulation, discrete Fourier transforms, and mathematical operations.

Pandas

For Python data analysis and manipulation, Pandas is a robust and popular open-source toolkit. Its capacity to manage real-world data effectively and without linguistic constraints makes it extremely valued. A wide range of data formats, including tabular data from Excel spreadsheets and SQL databases, time series data frequently used in scientific and financial applications, arbitrary matrix data for intricate structures, and statistical data for efficient analysis, are especially well-suited for the library. Pandas facilitates data manipulation by making it simple and easy for users to carry out operations including data transformation, filtering, and cleaning. It is a vital tool for every data-driven project due to its adaptability and widespread use in the data science community.

Matplotlib

A robust and popular Python toolkit for producing static, interactive, and animated visualizations, Matplotlib is mainly used for 2D data representation. It is compatible with the whole SciPy stack because it is a cross-platform library. Because it simplifies complex data and aids in decision-making, visualization is an essential component of data analysis. You may build a variety of graphs with Matplotlib, including bar charts, line graphs, and scatter plots, which gives you the freedom to select the best depiction for your data. Matplotlib's versatility is one of its best features. Almost everything is customizable, including the colors and axes' labels, as well as the addition of interactive graph elements. It can be readily incorporated into other frameworks like WxPython, Tkinter, and PyQt because it also includes an object-oriented API and a Python GUI toolkit. Matplotlib is compatible with online apps and technologies such as Jupyter Notebooks, which allow visualizations to be directly integrated into the notebook interface for a more engaging experience. All the tools you need to successfully communicate your data are provided by Matplotlib, regardless of whether you're working on straightforward plots or more complex visual representations. Anyone wishing to produce high-quality visualizations for data science, machine learning, or research uses it because of its extensive interoperability, customizable capabilities, and ease of use.

Seaborn

A Python framework called Seaborn, which is based on Matplotlib, makes it simpler to produce eye-catching and educational visualizations. It is perfect for analyzing and analyzing data since it easily connects with Pandas data structures. Seaborn works well with arrays and data frames to help produce graphs that are both aesthetically pleasing and intelligible through the use of statistical aggregation and semantic mapping. A feature of Seaborn is its data-oriented API, which allows you to concentrate on the message your visualizations are attempting to convey rather than the technical aspects of their creation. It makes it easier to analyze and visualize trends in your data, which making it an excellent tool for data exploration. If you are familiar with Pandas, Matplotlib, and NumPy, you will be well-equipped to make the most of Seaborn.

TensorFlow

TensorFlow is an open-source framework for large-scale numerical computation and machine learning. Several deep learning and machine learning techniques and frameworks are combined into a single, potent platform. TensorFlow offers an intuitive front-end API that facilitates the development and implementation of applications. With its extensive application in data gathering, model training, and prediction, it provides a complete solution for machine learning processes.

Keras

An intuitive, high-level API, Keras was created with a focus on simplicity and ease of usage. Its main objective is to lighten the cognitive burden on developers so they may concentrate more on creating models rather than handling intricate code. Keras guarantees that its interface is user-friendly and easy to comprehend by following industry standards. The API is made to manage common use cases with little input and to give developers concise, early, and useful error notifications so they can troubleshoot more rapidly. The deep learning community's preferred option for both novice and seasoned practitioners, it also provides a wealth of documentation and developer tools.

3.3 Key Concepts

To comprehend the training of machine learning models, it is imperative to first understand fundamental principles in deep learning.

3.3.1 Activation Function

The activation function is essential in converting input signals into output signals by applying a nonlinear function to the linear input. It adds nonlinearity to the network, enabling the model to learn more complicated patterns. Some of the most popular activation functions are sigmoid, softmax, ReLU, Leaky ReLU, Tanh, and Softplus.

Sigmoid Function:

In machine learning, the sigmoid function, which generates an S-shaped curve, is commonly known as the logistic function. Its primary function is to transform real-valued numbers into probabilities, which is especially beneficial when the output must be between 0 and 1. In machine learning models, the sigmoid function is frequently used in the final layer to convert the model's output into a probability score, making it easier to interpret. It is commonly employed in logistic regression models for binary categorization, such as predicting whether a customer will buy. The formula for the sigmoid function is

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

ReLU (Rectified Linear Unit) Function:

ReLU is an activation function that ensures piecewise linearity. If the input is positive, ReLU just returns the input value. If the input is negative, it returns zero. ReLU has been the default activation function for many neural networks due to its speed and efficiency during training. It works particularly well in deep network hidden layers, where it speeds up training by keeping a constant derivative for all positive inputs. ReLU is frequently related with "rectified networks." The formula for ReLU is:

$$\text{ReLU}(x) = \max(0, x)$$

Leaky ReLU Function:

Leaky ReLU is a type of ReLU that lets a small, non-zero gradient through when the input is negative. This helps fix the ”dying ReLU” problem, which is when neurons stop working. This little change lets the network keep learning, even when the inputs are bad. The formula for Leaky ReLU is:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

Tanh (Hyperbolic Tangent) Function:

The Tanh function is another activation function that works like sigmoid but gives outputs from -1 to 1 instead of 0 to 1. This means that it is zero-centered, which can be useful in some situations because it helps the model converge faster. For tasks that involve sequences, Tanh is widely utilized in Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Tanh's equation is:

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

Softmax Function:

In issues with more than one class, the softmax function is often utilized. It takes a vector of real numbers and turns it into a probability distribution where the total of all the probabilities is 1. It gives bigger input values a higher chance of happening and smaller input values a lower chance of happening. Softmax is basically an extension of the sigmoid function that works for projects with more than one class. It's especially helpful when the classes don't overlap. The softmax function's equation is:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

Softplus Function:

Softplus is a smooth alternative to ReLU that works everywhere. ReLU has a sharp corner at zero, which makes it hard to differentiate. When avoiding problems with non-differentiability is important, Softplus is a smart solution. It is typically used instead of ReLU in deep learning models. Softplus's formula is:

$$\text{Softplus}(x) = \ln(1 + e^x)$$

3.3.2 Data Processing

The process of data processing involves converting raw data into a form that computers can readily comprehend and utilize. Proper data processing is critical for a model's accuracy since it allows the algorithm to identify trends and generate accurate predictions. In practice, datasets originate from a variety of places, which might contribute noise and inconsistency. Because the model may have a more difficult time detecting significant patterns due to this noise, the quality of the results may suffer. It is critical to process data correctly to guarantee that it is clean and prepared for analysis. To make machine learning models more accurate and reliable, this is a crucial step.

3.4 Related Algorithms

3.4.1 Logistic Regression Algorithm

One popular approach to binary classification is logistic regression, which seeks to forecast the likelihood that a specific data point is part of a specific class. One goal of logistic regression is to determine the likelihood that the class label Y is dependent on the input features X. The following equation shows the result of applying Bayes' theorem to the calculation of this probability:

$$P(M|N) = \frac{P(N|M)P(M)}{P(N)}$$

/In this case, P(M—N) signifies the likelihood of the data point being a member of class M given the features N, P(N—M) denotes the likelihood of the data point, P(M) denotes the prior probability of class M, and P(N) denotes the prior probability of the features.

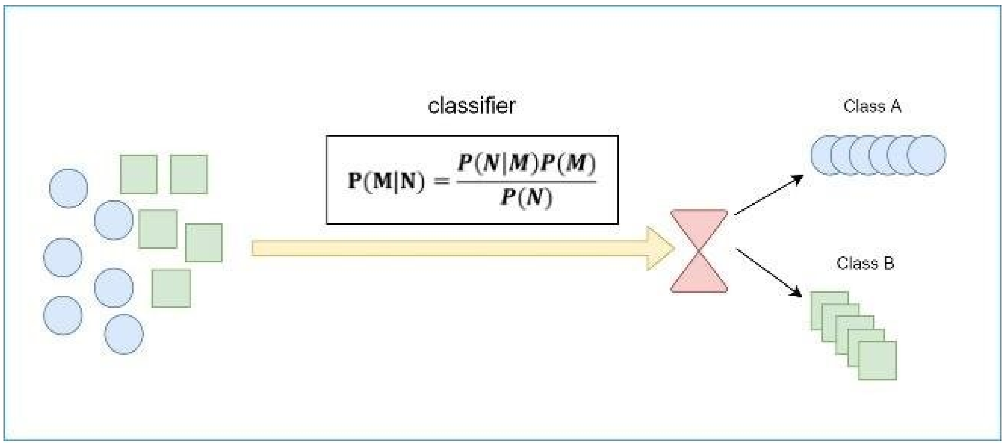

Figure 3.1: Logistic Regression Architecture

Logistic regression architecture is best understood as an elementary procedure. To begin, weights are applied to the input data, which consists of features. A function is then called upon to compute the weighted sum of these attributes using this combination. A Sigmoid function is applied to the total, converting it to a probability value between 0 and 1. At last, a choice is made: data points are assigned to Class A if the likelihood is more than 0.5, and to Class B if it is less than 0.5. This method aids logistic regression in classifying data points according to their likelihood of belonging to a certain set.

3.4.2 Support Vector Machine (SVM)

One common machine learning approach for classification and regression is Support Vector Machine (SVM). It divides data points into several categories by drawing a decision boundary, often known as a hyperplane. To maximize the margin between the two classes, or the space between the nearest data points on either side, the SVM forms a straight line (or hyperplane) for linear data. A greater margin indicates that the classifier is performing better. When dealing with non-linear data, support vector machines employ a kernel trick. This method creates an illusion of a higher-dimensional space where linear boundaries are achievable, even though this is not the case in the original space.

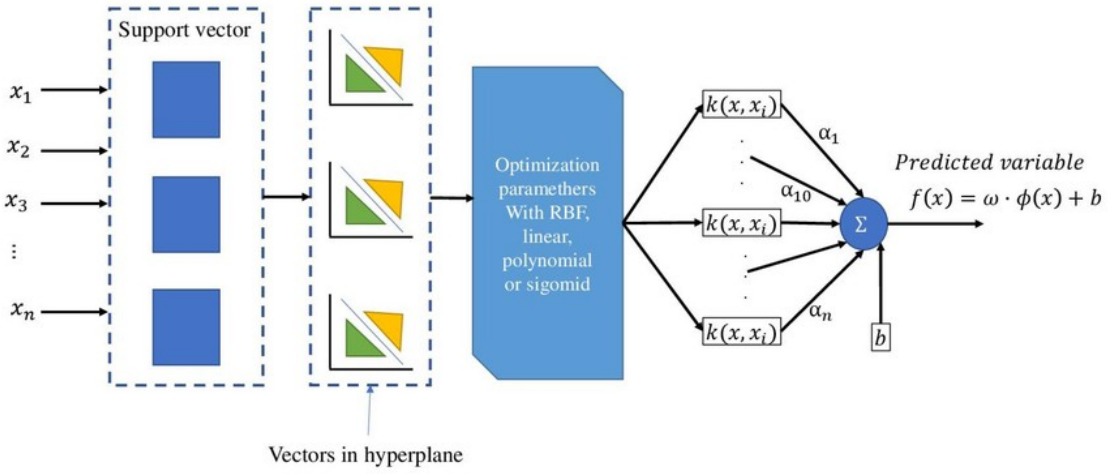

Figure 3.2: SVM Architecture

Support vectors are the most important data points that are closest to the decision border in SVM design. A kernel function sends these support vectors through a higher-dimensional space. This change makes it possible for a linear hyperplane to separate the data in this new space. The program then strives to improve the hyperplane so that it gives the two classes the biggest possible gap. The SVM model employs the optimized hyperplane to produce predictions by adding together the weighted outputs of the kernel function and the bias term. This gives reliable classification results.

3.4.3 Random Forest Algorithm

Random Forest is a strong machine learning method that may be utilized for both classification and regression tasks. During training, it builds several decision trees and then makes predictions based on the trees that get the most votes. Each tree in the forest learns from a random part of the data. The model usually employs Gini Impurity to pick which features to use at each node. Bootstrap sampling is one of the most important parts of Random Forest. This means that each tree is trained on a slightly different version of the data, which was made by randomly selecting and replacing parts of the original dataset. This is where the idea of Out-of-Bag (OOB) mistake comes in. Some data points are left out of the training for some trees, and these points are subsequently employed as a kind of validation set. We can get a good idea of how the model will do on data it hasn't seen before by looking at how the trees forecast these OOB points. This takes the predictions from all the trees and makes a final answer. The approach enables the model produce predictions that are more accurate and reliable than those made by only one decision tree. It is very helpful for dealing with complicated data and classes that aren't evenly distributed, like when you need to find hate speech, where you need a model that can work well in many various situations.

3.4.4 Decision Tree

One powerful machine learning tool with dual classification and regression capabilities is the decision tree. In order to make decisions in a tree-like structure, it continually partitions the dataset into smaller groups according to the feature values. The internal nodes represent feature-based choices, while the branches indicate the outcomes of those choices. In this diagram, the leaf nodes represent the final result or forecast. When deciding the feature to use for data splitting at each node, the algorithm takes quality metrics like Gini Impurity and Entropy into account. Even though they are simple to implement, Decision Trees run the risk of overfitting if trained to a depth of too many layers. Pruning or establishing a depth limit for the tree are typical solutions to this problem. Because of its versatility, Decision Trees can solve a wide range of problems using either structured or unstructured data. However, its instability is a major drawback. Even little modifications to the data could have a significant impact on the structure of the tree. In spite of these issues, decision trees are effective in discovering complex nonlinear relationships. When it comes to classification and regression, a Decision Tree is one of the most potent machine learning algorithms available. Datasets are subsetted recursively according to feature values, resulting in a decision-tree structure. Every internal node stands for a feature-based choice, and every branch signifies the decision's outcome. In this diagram, the leaf nodes stand for the ultimate forecast or result. Using measures like entropy or Gini impurity to assess the quality of the splits, the algorithm chooses the optimal feature to divide the data at each node. Overfitting is a problem with Decision Trees, even though they are easy to understand and use, particularly when the tree depth increases. Pruning or establishing a maximum depth for the tree are common methods used to reduce this effect. Decision trees are useful for many different types of situations because they can process both continuous and categorical data. One disadvantage, though, is that they are unstable; even little changes to the data can cause the tree's structure to change dramatically. Decision trees are nonetheless helpful for capturing complicated nonlinear interactions, even with these limitations.

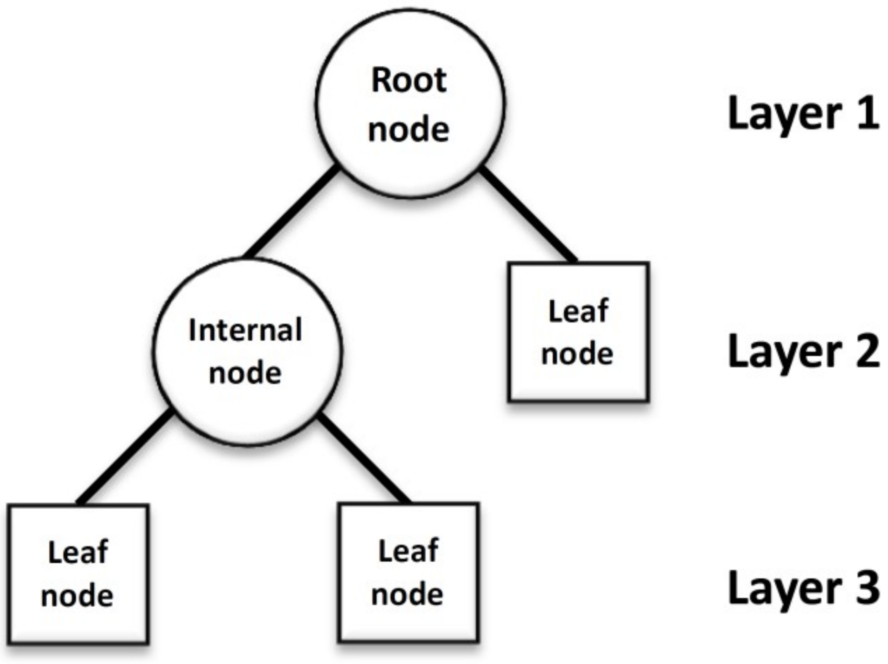

Figure 3.3: Decision Tree Architecture

This graphic illustrates a basic tree form that is frequently used in data structures like decision or binary trees. At the top of the tree (Layer 1) is a root node that acts as the structure's entrance. An internal node (Layer 2) sits beneath the root and branches off to two leaf nodes. The ultimate data or judgments are contained in these leaf nodes, which are terminal nodes in Layer 2. The leaf nodes, which stand in for the tree's ends, reappear in the third layer, or Layer 3. The root, internal nodes, and leaf nodes are all arranged hierarchically in the structure, with internal nodes acting as middlemen to direct the flow of choices or information to the leaf nodes.

3.4.5 XGBoost

A popular machine learning technique for classification and regression applications, XGBoost (Extreme Gradient Boosting) is a member of the gradient boosting family and is incredibly effective and scalable. It functions by successively constructing decision trees, with each new tree aiming to fix the mistakes of the ones that came before it. In order to avoid overfitting, the algorithm minimizes a regularized objective function, which combines a regularization term with a loss function (such log loss or mean squared error). This can be stated as follows:

$$L(\theta) = \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{y}_i) + \lambda \sum_{k=1}^{K} \|\theta_k\|_2^2$$

The loss function $\mathcal{L}(y_i, \hat{y}_i)$ measures the difference between the actual values y_i and the predicted values $\hat{y}_i$. θ_k represents the model parameters, and λ is the regularization strength that controls overfitting. XGBoost is notable for using L1 (Lasso) and L2 (Ridge) regularization to make models less complicated. It also allows parallel processing, which speeds up training, especially when working with big datasets. XGBoost can deal with missing values and employs tree pruning to make the decision tree grow faster. Key hyperparameters, like `colsample_bytree`, `learning_rate`, `max_depth`, `n_estimators`, and `subsample`, offer flexibility for tuning the model and improving its performance.

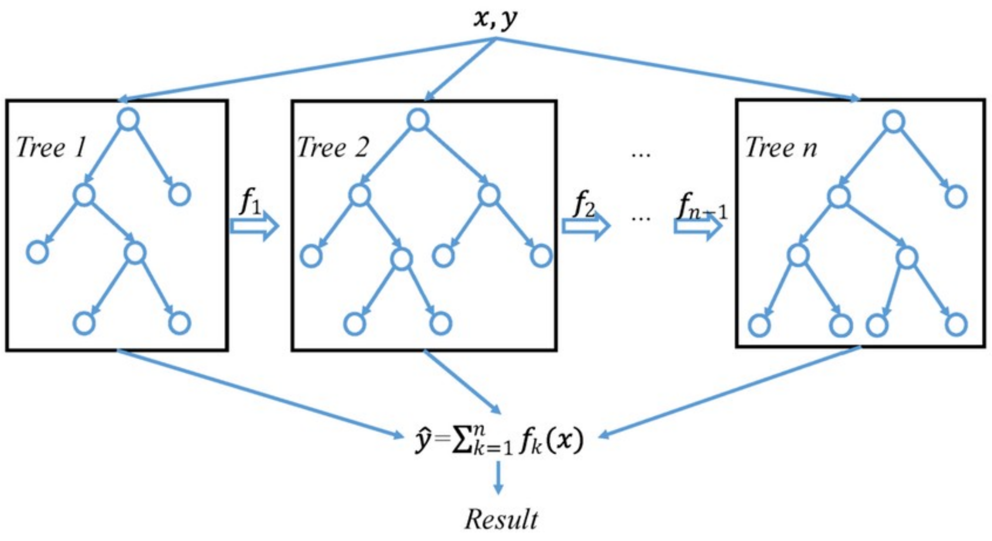

Figure 3.4: XGBoost Architecture

The graphic demonstrates an ensemble learning model in which several decision trees (Tree 1, Tree 2, ..., Tree n) work together to create predictions. Every tree develops its own guess based on the data it gets. Then, all the guesses are put together to make the final answer. This strategy makes things more accurate by using the best parts of numerous trees instead of just one.

3.4.6 LSTM

Long Short-Term Memory (LSTM) is a type of neural network designed to handle data sequences, such as time series or phrases, where anticipating future events requires knowledge of past events. Because LSTM has a clever way of "remembering" or "forgetting" things over time, it performs better than standard neural networks at complex tasks like language translation or market price prediction. The input gate determines what fresh information should be added to memory, the output gate determines what information should be sent on to the following stage, and the forget gate determines what should be forgotten from previous memories. By serving as filters, these gates enable the model to retain important information while forgetting less important ones. By altering these gates during training, LSTM gains the ability to more accurately predict what will happen next in a sequence. For tasks like text classification, speech recognition, and even handwriting analysis—where past information influences future outcomes—LSTM is excellent. This is due to its ability to retain key characteristics over time. The LSTM architecture is seen below.

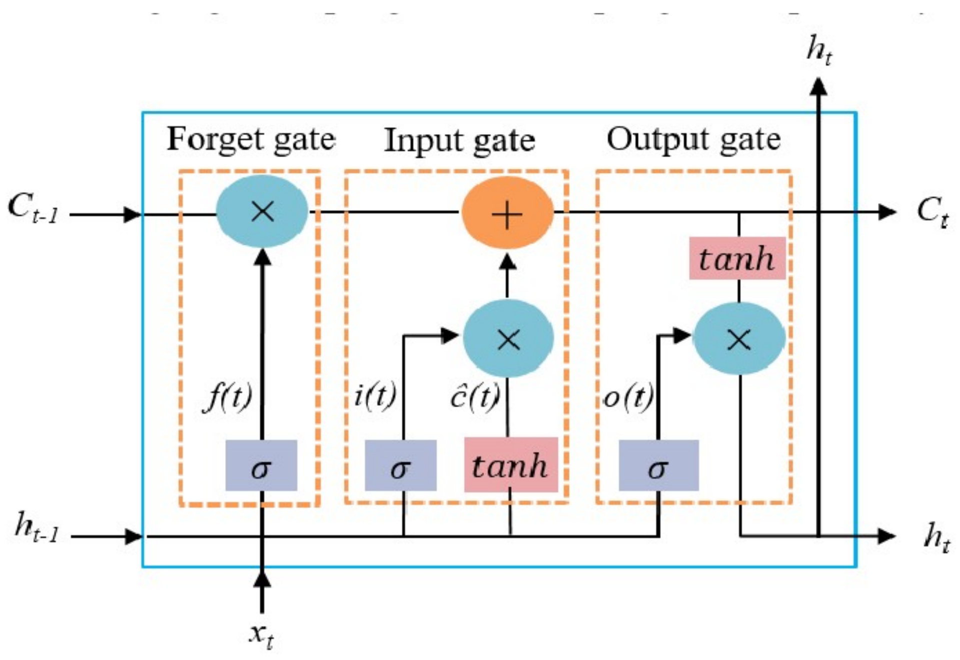

Figure 3.5: LSTM Architecture

The diagram shows the Long Short-Term Memory (LSTM) unit, which is a type of neural network that can handle sequential input like text or time-series data while solving issues like the vanishing gradient problem. There are three main gates that make it work: the forget gate, the input gate, and the output gate. The forget gate, which is controlled by a sigmoid function, decides which previous information should be ignored. The input gate controls the flow of new data into the memory by updating the memory cell with both a sigmoid and a tanh activation. The output gate decides what data will be sent out by combining the current memory with a tanh function. The LSTM is very useful for tasks like language processing and time-series forecasting because the gates work together to keep and change the memory, which helps the system remember vital information and forget useless information. The diagram shows how a Long Short-Term Memory (LSTM) unit is set up. This is a specific kind of neural network that can handle sequential input, like text or time-series data, while avoiding problems like the vanishing gradient problem. The forget gate, the input gate, and the output gate are the three key parts that make it work. The forget gate, which is controlled by a sigmoid function, decides which old information should be thrown away. The input gate controls how new information gets into the memory. It does this by utilizing both a sigmoid and a tanh activation to update the memory cell. The output gate uses a tanh function to mix the current memory with the information that will be sent out. These gates operate together to keep and change the memory, which lets the LSTM remember vital information and forget unimportant information. This is why it's so valuable for things like language processing and forecasting time series.

3.4.7 BiLSTM

An improved form of the traditional Long Short-Term Memory (LSTM) architecture is called Bidirectional Long Short-Term Memory (BiLSTM). It enables the model to comprehend context from both the past and the future by processing sequential input in both directions. Two separate LSTM layers are used to achieve this. The sequence is processed by the forward LSTM from left to right, absorbing previous information, and by the backward LSTM from right to left, absorbing future information. To create a single representation that incorporates data from both sides, the outputs of these two layers are often combined or merged together. As a result, each token in the series has a more comprehensive and complete image. BiLSTMs excel at tasks where context from both sides of a token is crucial, like speech recognition, emotion analysis, machine translation, and named entity recognition, where a word's meaning can vary depending on the words that come before and after it. For example, in order to understand what "bank" means in the sentence "The bank will open tomorrow," you must understand the meaning of the terms that come before and after "bank." Because they can identify intricate associations, resolve meaning ambiguities, and improve the efficiency of tasks using sequential or time-series data, BiLSTMs are highly valuable in Natural Language Processing (NLP). However, because two LSTM layers must be trained and maintained, this two-way processing has a greater memory and processing cost. Even yet, BiLSTMs are frequently used in NLP applications due to their superior performance over unidirectional models and ability to create representations that are rich in context. The fundamental structure and operation of BiLSTM are illustrated in the following diagram:

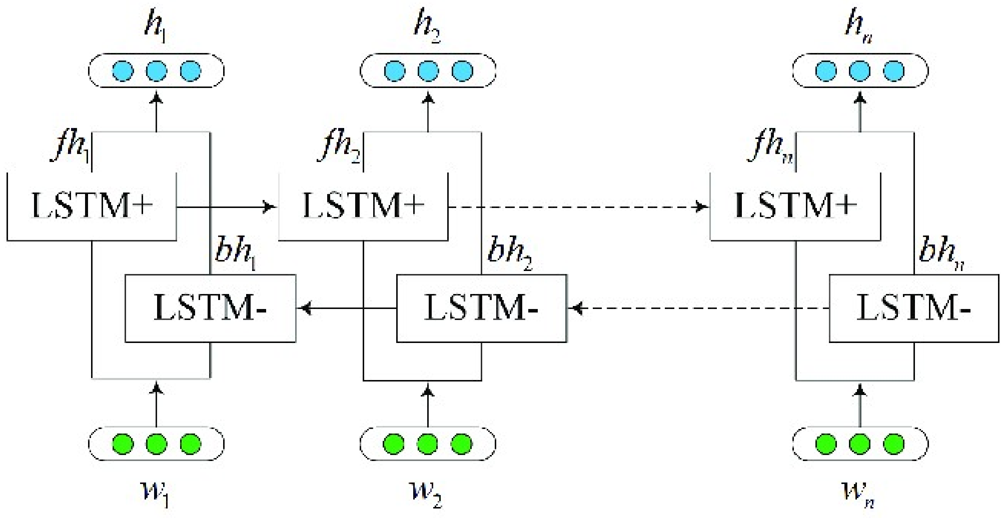

Figure 3.6: BiLSTM Architecture

This figure shows how a rudimentary Bidirectional Long Short-Term Memory (BiLSTM) network is set up. It can handle data in both directions. The LSTM+ blocks manage the forward pass by looking at the data from start to finish, while the LSTM-blocks handle the backward pass by looking at the data from start to finish. This lets the BiLSTM understand the whole context of a sequence, both its history and its future. After getting inputs, the network makes hidden states as outputs. These are affected by biases and weights that help guide the data through the system. This dual-processing setup is especially useful for tasks like sentiment analysis and language translation where it's important to understand the context both before and after a certain point.

3.4.8 Gated Recurrent Unit

A type of neural network called the Gated Recurrent Unit (GRU) was developed to address some of the issues with standard Recurrent Neural Networks (RNNs), particularly with regard to recalling information across long periods. Although GRUs process data in the same sequence as other RNNs, they are better at handling long-term dependencies because of their gating mechanism. The two primary gates they employ are the update gate and the reset gate. The update gate determines how much of the previous data should be retained before proceeding to the following phase. Over time, this aids the model in remembering important details. In contrast, when the model receives new input, the reset gate instructs it how much of the existing memory to erase. This aids the model in concentrating on the most crucial data. By combining these two gates, GRUs can quickly decide which data should be kept and which should be deleted. They don't require LSTMs' more intricate structure. This increases their computational efficiency, but they are still capable of performing tasks like language translation, speech recognition, and time series prediction that need for a comprehension of long-range dependencies. Because they strike a fair balance between simplicity and efficiency, GRUs are a common solution for many sequential data problems.

3.4.9 BERT

The transformer design is what makes BERT (Bidirectional Encoder Representations from Transformers) work. It reads text from both left to right and right to left at the same time. The input to BERT is tokenized text, with each token represented by a WordPiece embedding. There are a few different self-attention mechanisms that are utilized to process the input tokens. The mathematical operation for self-attention is:

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

where Q is the query, K is the key, V is the value, and d_k is the dimension of the key vectors. This makes it possible for every token to pay attention to every other token, which aids in its comprehension of their relationships. BERT is pre-trained using the Masked Language Model (MLM) goal. In this model, certain input tokens are hidden at random, and the model learns to infer the meaning of these hidden tokens from their surrounding context. The loss function of the MLM is:

$$\mathcal{L}(\hat{y}_i, y_i) = -\sum_i \log P(\hat{y}_i \mid X)$$

The probability of correctly identifying the original token given its context is denoted by the phrase P(word— X). A task-specific layer, such as a classification head, is added by BERT during fine-tuning, and the model is modified to minimize a task-specific loss function. The final result is utilized for sentence-level tasks like classification, particularly when it comes to the [CLS] token.

$$\text{CLS embedding} = f(\text{final hidden state of } [CLS])$$

BERT can do a lot of different natural language processing tasks well since it uses both pre-training and bidirectional context.

3.4.10 RoBERT

An enhanced version of BERT called RoBERT (Robustly optimized BERT approach) modifies key aspects of BERT's training procedure to increase its performance on tasks involving natural language processing. Like BERT, RoBERT makes use of the transformer architecture. This indicates that every token in the sequence is aware of every other token. However, by eliminating the next sentence prediction (NSP) task and concentrating solely on masked language modeling (MLM), RoBERT simplifies the pre-training objective. In multilevel marketing (MLM), a certain number of tokens are concealed at random, and the model learns to infer their identity from their surroundings. Because RoBERT is trained on a larger dataset than BERT, with more iterations and dynamic masking, it learns more robust language representations. Following pre-training, RoBERT is modified for certain tasks, such as classification, by overlaying the [CLS] token's final concealed state with a layer specific to that task. Because RoBERT has more training data, dynamic masking, and longer training, it outperforms BERT on a number of benchmarks. Applications such as named entity recognition, question answering, and text categorization benefit greatly from this.

3.4.11 DistilBERT

In DistilBERT, a smaller "student" model learns from a bigger "teacher" model. It is a shorter, faster, and better version of BERT that was created by knowledge distillation. The self-attention mechanism based on transformers and much of BERT's architecture are still there, but it is much lighter because it has fewer layers (typically 6 instead of 12 in BERT). It is pre-trained on a large corpus using a masked language model (MLM) aim, just as BERT. It is taught to copy BERT's outputs, nonetheless, by using both distillation loss and cross-entropy. This lets it learn from BERT's soft targets or probabilistic outputs while yet keeping its performance competitive. It is faster, uses less memory, and is great for situations where there isn't a lot of computing power available. It's around 60% smaller than BERT but still does about 97% of the same things well, such classifying text and answering questions. It is a good choice for many NLP applications because it is smaller than BERT but yet gives results that are almost as accurate. It is a smaller, faster, and more efficient variant of BERT. It was made by taking a smaller "student" model and teaching it what it needs to know by using a bigger "teacher" model. It keeps most of BERT's structure, like the transformer-based self-attention mechanism, but cuts the number of layers in half (usually 6 layers instead of 12), making it significantly lighter. It is pre-trained on a large corpus using a masked language model (MLM) objective. However, it is trained to imitate BERT's outputs using both cross-entropy and distillation loss. It learn from BERT's soft aims (probabilistic outputs) while still doing well.

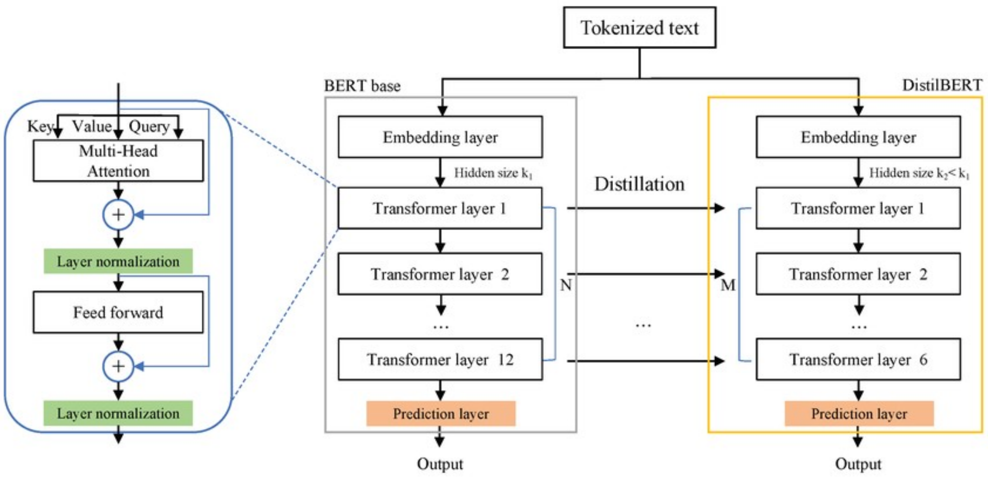

Figure 3.7: DistilBERT Architecture

The graphic illustrates how DistilBERT is a condensed form of BERT that is intended to be faster and smaller while maintaining a large portion of BERT's functionality. The architecture is shown, with two models processing the tokenized text: the distilled DistilBERT model with six transformer levels and the full BERT model with twelve transformer layers. In order to achieve distillation, DistilBERT transfers knowledge from the larger model to its smaller transformer layers, but both models begin with the same embedding layer. DistilBERT offers a more lightweight and efficient architecture while maintaining competitive output performance with fewer layers.

3.4.12 BERT Multilingual

BERT Multilingual is a variant of BERT particularly engineered to accommodate many languages inside a singular model, providing an effective solution for multilingual natural language processing (NLP) problems. This model, in contrast to the original BERT pre-trained solely on an extensive English corpus, is trained on data from over 100 languages, encompassing both high-resource languages like English, Spanish, and Chinese, as well as low-resource languages such as Swahili, Telugu, and Urdu. The principal innovation resides in its shared vocabulary methodology, employing WordPiece embeddings to represent words as subword units, so enabling the model to manage the varied vocabulary of various languages without necessitating distinct models for each language. The common vocabulary signifies that same subword representations are utilized across languages, facilitating the model's acquisition of cross-lingual patterns and correlations. The design is unchanged from BERT, featuring many transformer layers with bidirectional self-attention algorithms that capture contextual interactions between tokens in both directions. In the pre-training phase, the model is trained with the Masked Language Model (MLM) objective, wherein a segment of the input tokens is randomly obscured, and the model learns to infer these masked tokens from the surrounding context. This allows the model to acquire contextualized representations for words across numerous languages. The same representation of subwords across languages enables the model to transfer knowledge between them, which is especially advantageous for low-resource languages with scarce training data. One of the issues encountered is the varied linguistic structures and grammatical systems of the languages it encompasses. Diverse languages exhibit distinct syntactic structures, word ordering, and morphological features, necessitating that a multilingual model generalizes across these variations. Notwithstanding these obstacles, the model has exhibited robust performance in multilingual tasks, including as text classification, named entity recognition (NER), part-of-speech tagging, and machine translation, surpassing earlier multilingual models in multiple benchmarks. Furthermore, it is capable of managing tasks that necessitate the comprehension and processing of text in other languages, including cross-lingual transfer learning and multilingual sentiment analysis. This model has demonstrated significant utility in practical multilingual NLP applications where training an individual model for each language is unfeasible. It enables enterprises, academics, and developers to utilize a singular, cohesive model for processing data across many languages, hence diminishing computing expenses and training duration. Its capacity to manage code-mixed data, such as Banglish, which combines English and Bengali, enhances its applicability in varied linguistic contexts. Moreover, it facilitates zero-shot cross-lingual transfer, indicating that the model can execute tasks in languages for which it was not specifically trained, utilizing insights gained from other languages to provide predictions in novel, unencountered languages.

3.4.13 Bangla BERT

Bangla BERT is a transformer-based language model that is specifically for the Bengali language. . This helps the model deal with the language complicated morphology and quickly handle words. Bangla BERT is trained on a large set of Bengali text and uses a masked language modeling (MLM) goal. In this goal, parts of the input tokens are randomly hidden, and the model has to guess what these hidden tokens are based on the context around them. This make possible to find both semantic and syntactic linkages inside the language. After pre-training, it is fine-tuned for certain NLP tasks, such as sentiment analysis, named entity recognition, and answering questions. The model's bidirectional attention mechanism allows for the understanding of the subject-object-verb (SOV) structure of Bengali, making it suitable for various applications. It also works well with code-switched data, which is data that mixes Bengali with other languages like English. However, It effectiveness can be limited by the availability of training data for occupations with few resources. This has done quite well on several Bengali NLP benchmarks. The next picture shows the tiered architecture of the inside:

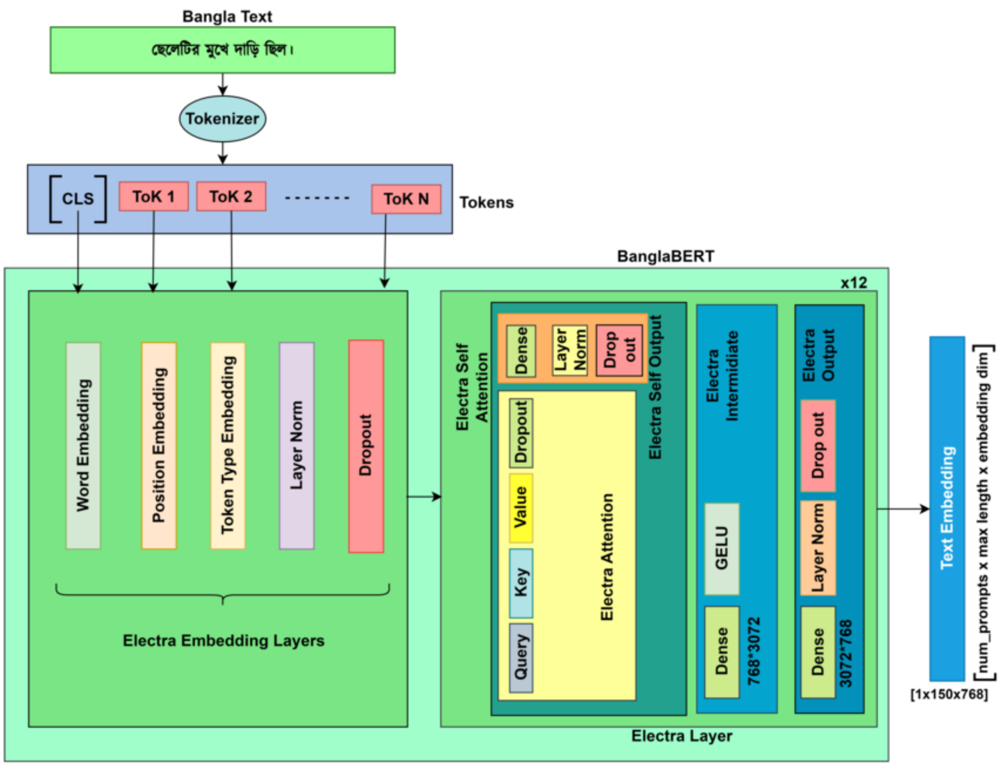

Figure 3.8: Bangla BERT Architecture

The model is designed to handle Bengali text and is based on the ELECTRA framework. It's known as BanglaBERT. The provided Bengali sentence is divided into tokens, and then special tokens such as [CLS] are added to indicate its beginning. The Word, position, and token type embeddings are among that the tokens undergo. Then dropout and layer normalization are applied to prevent the model from overfitting. The model core consists of 12 layers of Electra Attention, which calculate attention scores between tokens using query, key, and value processes. This layers use GELU activation to make things non-linear after parsing the tokens and capturing relationships . The final output is a high-dimensional vector called a Text Embedding that displays the entire phrase's meaning.

3.4.14 XLM RoBERTa

XLM-RoBERTa (Cross-lingual Robustly Optimized BERT Pretraining Approach) is a complex transformer-based model that can handle multilingual tasks and works quite well, especially in languages with few resources. It builds on the successful BERT architecture and adds a number of major innovations that make it much better at understanding and processing text in many different languages. One of its best features is that it uses a denoising goal, which is based on the ELECTRA model, to learn how to guess the original tokens from damaged input. This helps BERT tell the difference between real and false tokens better, which makes language representations more accurate and dependable then MLM . This denoising technique also helps it generalize better across languages, so it can do well even with minimal data for some languages. It has been trained on a far larger multilingual corpus than BERT, focusing on 100 different languages. Also, it takes longer to train so that people may learn the languages better. It also takes away BERT's purpose of predicting next phrase, which makes the pretraining process more focused on language modeling with MLM and helps it better understand how words are related to each other. What makes it special is that it can move knowledge from one language to another. It can use its understanding of high-resource languages to do things like text classification, named entity recognition, and answering questions.

3.4.15 Stacked LSTM

A Stacked Long Short-Term Memory (LSTM) network is a better variant of the regular LSTM network. The numerous layers of LSTM units must be layered on top of each other. By using several layers of abstraction, the strategy helps the model find more complex patterns in sequential data. It is made to solve the problem of gradients that disappear. They are great for activities that involve long-range dependencies which includes time series forecasting, voice recognition, and language processing. Each layer in a stacked LSTM sends its output to the next layer once it has processed the sequence. The first layer might look for simple, short-term patterns in the data. The deeper layers may be able to find more complicated connections and interactions that last longer. By stacking many layers, each with its own memory cells, the model may better capture both short-term and long-term relationships. This makes it better at extrapolating from data and helps it make accurate predictions about new, unknown sequences. Stacked LSTMs are great for things like machine translation, where the model needs to remember parts of a previous phrase to make sure the translations are correct, and sentiment analysis, where it's important to understand the context of a group of words or sentences. Stacked LSTMs are employed in many hard applications because they are a key tool for solving problems with complex, sequential data.

Chapter 4

Methodology

4.1 Introduction to Methodology

In this chapter, we outline the methodology employed in developing and evaluating models for multilingual cyberbullying and hate speech detection. The approach encompasses data collection, preprocessing, model selection, training procedures, and evaluation metrics. The primary goal is to detect cyberbullying and hate speech across three languages: English, Banglish (a transliterated form of Bengali using English script), and Bengali. Additionally, a language identifier model is developed to preprocess and route inputs to the appropriate language-specific models.

The datasets used are sourced from publicly available repositories. For the Bengali dataset, we utilized the "Bengali Hate Speech Dataset" from Kaggle [14], which contains labeled examples of hate and non-hate speech in Bengali script. For the English dataset, the "Cyberbullying Classification" dataset from Kaggle [10] was employed, featuring multi-class labels for different types of cyberbullying. The Banglish dataset was obtained from Mendeley Data [8], consisting of binary labels for hate and non-hate content in Banglish.

To create the language identifier dataset, we sampled 4,000 rows from each of the three datasets, resulting in a balanced multilingual dataset with 12,000 samples labeled as "Bangla," "Banglish," or "English."

The methodology follows a systematic approach to ensure reproducibility and reliability. First, we establish a comprehensive framework for data collection that prioritizes diversity in linguistic expressions and cultural contexts. This is particularly important for cyberbullying detection, as harmful language often varies significantly across different communities and platforms.

Our research design incorporates both quantitative and qualitative analysis methods. The quantitative aspect focuses on statistical performance metrics such as accuracy, precision, recall, and F1-scores across different model architectures. The qualitative component examines the linguistic patterns and cultural nuances that influence model performance in each language.

The experimental setup follows established best practices in machine learning research, including proper train-validation-test splits, cross-validation procedures, and statistical significance testing. We ensure that our evaluation methodology provides robust and generalizable results that can inform future research in multilingual text classification.

4.2 Data Collection and Preprocessing

4.2.1 Data Sources

The datasets were collected from reliable online sources to ensure diversity and relevance to cyberbullying and hate speech detection.

- **Bengali Dataset**: Sourced from Kaggle, this dataset includes approximately 30,000 samples labeled as "Hate" or "Non-Hate." It focuses on Bengali text from social media platforms.

- **English Dataset**: From Kaggle, this dataset comprises around 47,000 tweets labeled into multi-classes: age, ethnicity, gender, not_cyberbullying, other_cyberbullying, and religion.

- **Banglish Dataset**: Obtained from Mendeley, it contains about 5,000 samples with binary labels "Hate" or "Not Hate," representing transliterated Bengali text.

- **Language Identifier Dataset**: Constructed by sampling 4,000 rows from each of the above datasets, ensuring balanced representation. Labels were assigned based on the source dataset's language.

The selection of these datasets was based on several criteria including data quality, annotation reliability, and representativeness of real-world social media content. Each dataset underwent thorough quality assessment to identify potential issues such as annotation inconsistencies, duplicate entries, and class imbalances.

For the Bengali dataset, special attention was paid to the variety of dialectical variations present in the text. Bengali, being spoken across different regions with distinct linguistic patterns, requires careful consideration of these variations to ensure model generalizability.

The English dataset's multi-class nature provides valuable insights into different types of cyberbullying behaviors. The categories include age-based harassment, ethnic discrimination, gender-based attacks, religious intolerance, and general cyberbullying patterns. This classification scheme enables more nuanced analysis of harmful online behaviors.

The Banglish dataset presents unique challenges due to its transliterated nature. Banglish text often lacks standardized spelling conventions, making preprocessing more complex. However, this dataset is crucial for understanding cyberbullying patterns in South Asian online communities where code-mixing and transliteration are common.

Quality control measures were implemented throughout the data collection process. These included manual verification of random samples, consistency checks across annotations, and statistical analysis of class distributions to identify potential biases in the datasets.

4.2.2 Preprocessing Steps

Data preprocessing is crucial for handling noisy social media text. The following steps were applied uniformly across datasets where applicable:

1. **Text Cleaning**: Removal of URLs, mentions (@user), hashtags, emojis, and special characters using regular expressions.

2. **Tokenization**: For English,Bangla and Banglish, NLTK tokenizer was used.

3. **Stopword Removal**: Remove from English Dataset only.

4. **Feature Extraction**: TF-IDF vectorization with n-grams (1-2) for ML models. For DL and transformers, tokenization via model-specific tokenizers (e.g., BERT tokenizer).

5. **Class Balancing**: Oversampling minority classes using SMOTE for imbalanced datasets.

6. **Train-Test Split**: 80-20 split for training and evaluation.

For the language identifier, minimal preprocessing was applied to preserve linguistic features.

The text preprocessing pipeline consisted of six key steps to prepare the datasets for modeling. First, text cleaning involved removal of URLs, mentions (@user), hashtags, emojis, and special characters using carefully crafted regular expressions. This step was designed to eliminate noise while preserving the meaningful linguistic context around removed elements.

Second, tokenization was performed using NLTK tokenizer for English, Bangla, and Banglish texts. For Bengali, due to the script's unique characteristics such as conjunct consonants and vowel modifiers, a custom tokenizer was developed to accurately segment words.

Third, feature extraction utilized TF-IDF vectorization with n-grams (1-2) for traditional machine learning models. Deep learning and transformer-based models employed model-specific tokenizers, such as the BERT tokenizer, to convert text into suitable input formats.

Fourth, class balancing was addressed using SMOTE (Synthetic Minority Oversampling Technique) to handle imbalanced datasets, which was especially critical for Bengali and Banglish data to ensure fair model performance.

Fifth, an 80-20 train-test split was applied for model training and evaluation.

For the language identifier model, minimal preprocessing was applied to retain linguistic features essential for accurate language detection.

Overall, the preprocessing pipeline balanced the need to remove noise and maintain meaningful linguistic information, adapting methods to each language's unique features and dataset characteristics.

4.3 Model Architectures

We evaluated a range of models: traditional ML, DL recurrent networks, and transformers. Models were selected based on their suitability for text classification.

4.3.1 Machine Learning Models

- **Logistic Regression**: A linear model with L1/L2 regularization. - **Decision Tree**: Entropy/Gini criterion, tuned for depth. - **Random Forest**: Ensemble of decision trees, with n_estimators=100-200. - **XGBoost**: Gradient boosting with learning rate 0.1, max_depth=5-7. - **SVM**: RBF/linear kernel, C=1-10.

Hyperparameters were tuned using GridSearchCV.

Traditional machine learning models serve as strong baselines for text classification tasks. Logistic regression, despite its simplicity, often provides competitive performance for text classification due to the high-dimensional nature of text data.

The regularization parameters (L1 and L2) were tuned to prevent overfitting while maintaining model interpretability.

Decision trees offer the advantage of interpretability, allowing us to understand the decision-making process. The tree depth and splitting criteria were optimized through cross-validation to balance model complexity with generalization ability. Feature importance analysis from decision trees provided insights into which linguistic features are most discriminative for cyberbullying detection.

Random Forest models combine the interpretability benefits of decision trees with improved performance through ensemble learning. The number of estimators was varied between 100 and 200 based on computational constraints and performance improvements. Bootstrap aggregating helps reduce overfitting common in individual decision trees.

XGBoost represents state-of-the-art gradient boosting techniques, offering excellent performance for structured data problems. The learning rate and maximum depth parameters were carefully tuned to optimize the bias-variance tradeoff. Early stopping was implemented to prevent overfitting during the boosting process.

Support Vector Machines with different kernels (RBF and linear) were evaluated to capture both linear and non-linear relationships in the text data. The regularization parameter C was tuned to control the tradeoff between margin maximization and training error minimization.

Hyperparameter optimization was conducted using GridSearchCV with 5-fold cross-validation. This systematic approach ensures that model performance is not due to fortunate parameter choices but represents genuine model capability.

4.3.2 Deep Learning Models

LSTM: Single layer with 128 units, dropout 0.2.

Bidirectional LSTM: Captures context from both directions.

Stacked LSTM: Two LSTM layers.

GRU: Similar to LSTM but with fewer parameters.

Long Short-Term Memory (LSTM) networks are well-suited for sequential data such as text, as they capture long-range dependencies while mitigating the vanishing gradient problem. The model architecture includes an embedding layer initialized and trained from scratch using Keras' built-in embedding layer, followed by recurrent layers and dense output layers with suitable activation functions.

The single-layer LSTM with 128 units balances model complexity and training efficiency. Dropout with a rate of 0.2 is applied for regularization, preventing overfitting by randomly dropping units during training.

Bidirectional LSTMs process sequences in both forward and backward directions, allowing the model to incorporate context from past and future tokens simultaneously, which is beneficial for text classification tasks where full sequences are available.

Stacked LSTM models consist of two recurrent layers, enabling hierarchical feature extraction. The first LSTM layer learns low-level sequential patterns, and the second layer captures higher-level abstractions, often improving model performance on complex classification problems.

Gated Recurrent Units (GRU) provide a simpler alternative to LSTMs by combining forget and input gates into a single update gate, which reduces the number of parameters and computational overhead while maintaining comparable accuracy.

For feature representation, TF-IDF vectorization with n-gram ranges (1-2) was used to convert text into numerical features as input to the deep learning models, instead of relying on pre-trained word embeddings like GloVe or FastText. This approach allowed for model training from scratch without dependence on external embedding resources.

The models were trained for up to 100 epochs with early stopping based on validation loss to prevent overfitting and reduce unnecessary training time. Early stopping monitors the model's performance on validation data and halts training when no improvement is observed over a set number of epochs.

4.3.3 Transformer Models

BERT / Multilingual BERT: Pre-trained models fine-tuned with a maximum sequence length of 128 tokens.

RoBERTa / XLM-RoBERTa: Optimized versions with improved training techniques, including removal of Next Sentence Prediction and dynamic masking.

Bangla-BERT: A Bengali-specific model trained exclusively on Bengali corpora.

Transformer models represent the current state-of-the-art in natural language processing. BERT (Bidirectional Encoder Representations from Transformers) utilizes bidirectional training, enabling it to learn contextualized representations by attending to both past and future tokens simultaneously.

The multilingual BERT model is especially relevant for this work due to its pretraining on multiple languages, including Bengali and English. This shared representation facilitates cross-lingual transfer, which is beneficial for handling low-resource languages.

Fine-tuning was conducted with a maximum input sequence length of 128 tokens, balancing computational resource demands and information retention. Input sequences longer than this were truncated, and shorter sequences were padded accordingly.

RoBERTa improves upon BERT by modifying the pretraining process — notably by removing the Next Sentence Prediction objective and employing dynamic masking strategies. XLM-RoBERTa extends these advances to the multilingual context.

Bangla-BERT is a language-specific transformer trained solely on Bengali text, often achieving superior results on Bengali NLP tasks by capturing language-specific patterns more effectively than multilingual counterparts.

Training transformer models involved careful hyperparameter tuning, including learning rate schedules and regularization. The Adam optimizer with learning rate warmup was used to promote stable convergence. Models were trained for up to 100 epochs with early stopping based on validation loss to prevent overfitting and reduce training time.

The self-attention mechanism intrinsic to transformer architectures also provides interpretability, enabling analysis of which input tokens influence predictions

— a valuable feature for understanding model decisions and detecting biases.

4.4 Training Procedures

Models were trained on a GPU-enabled machine (NVIDIA Tesla P100) provided by Kaggle. Batch sizes of 16 and 32 were used, and models were trained for up to 100 epochs with early stopping based on validation loss. The loss functions employed were binary or categorical cross-entropy, depending on the classification task.

The training infrastructure was selected to efficiently handle the computational requirements of deep learning and transformer models. The Tesla P100 GPU offers sufficient memory and compute capabilities to train models within reasonable time frames on the Kaggle platform.

Batch size selection balanced gradient stability and memory constraints. Smaller batch sizes (16) were typically used for transformer models due to their larger memory footprint, while batch size 32 was preferred for simpler deep learning architectures. Dynamic adjustment of batch size was performed where resource availability permitted.

The number of epochs was capped at 100, with early stopping employed to prevent overfitting and reduce unnecessary computation. Early stopping monitored validation loss and halted training if no improvement was seen for a predefined patience period (usually 3-5 epochs).

Loss functions were chosen according to task type: binary cross-entropy for binary classification tasks (e.g., Bengali and Banglish datasets) and categorical cross-entropy for multi-class classification problems (e.g., English and language identification datasets). These choices ensured stable training dynamics aligned with the problem nature.

All deep learning and transformer models were implemented using TensorFlow and Keras exclusively, without using PyTorch, to maintain consistency in the training framework.

Model checkpointing saved the best-performing model weights based on validation metrics, ensuring final model selection corresponded to optimal validation performance rather than the last training epoch.

Comprehensive training logs tracked metrics such as training and validation loss, accuracy, and resource utilization, supporting debugging and hyperparameter optimization across experimental runs.

4.5 Evaluation Metrics

- Accuracy, Precision, Recall, F1-Score. - ROC AUC (One-vs-Rest for multi-class). - Confusion Matrices.

The evaluation framework was designed to provide comprehensive assessment of model performance across different aspects. Accuracy provides an overall performance measure but can be misleading for imbalanced datasets. Therefore, additional metrics were employed to capture nuanced performance characteristics.

Precision measures the proportion of true positive predictions among all positive predictions, indicating how many of the predicted positive cases are actually positive. This metric is particularly important for cyberbullying detection, as false positives can lead to unnecessary content moderation actions.

Recall (sensitivity) measures the proportion of actual positive cases that were correctly identified. High recall is crucial for cyberbullying detection systems, as missing actual instances of harmful content can have serious consequences for user safety.

F1-score provides a harmonic mean of precision and recall, offering a balanced view of model performance. This metric is particularly valuable when dealing with imbalanced datasets where neither precision nor recall alone provides a complete picture.

ROC AUC (Area Under the Receiver Operating Characteristic curve) evaluates the model's ability to distinguish between classes across different threshold values. For multi-class problems, the One-vs-Rest approach calculates individual ROC curves for each class against all others, providing insights into class-specific performance.

Confusion matrices offer detailed breakdowns of prediction accuracy for each class, revealing patterns in model errors. These matrices help identify which classes are frequently confused and guide future model improvements.

Additional metrics considered include macro and micro averages for precision, recall, and F1-scores in multi-class scenarios. Macro averaging treats all classes equally, while micro averaging weights classes by their frequency, providing different perspectives on model performance.

Statistical significance testing was conducted where appropriate to ensure that performance differences between models are meaningful rather than due to random variation. This rigorous approach strengthens the reliability of our findings and conclusions.

Chapter 5

Results and Discussion

5.1 Introduction to Results

This chapter presents the results from the trained models across the four datasets: English cyberbullying (multi-class), Banglish hate speech (binary), Bengali hate speech (binary), and language identification (multi-class). We discuss performance, comparisons, and insights.

The results section is organized to facilitate comparison across different model architectures and datasets. Each dataset section begins with traditional machine learning models, progresses through deep learning approaches, and concludes with transformer-based methods. This organization allows readers to observe performance trends across increasing model complexity.

Performance analysis considers both quantitative metrics and qualitative observations about model behavior. Quantitative results provide objective comparisons, while qualitative analysis offers insights into model strengths, weaknesses, and practical applicability.

The discussion emphasizes not only which models perform best but also why certain approaches succeed or fail in specific contexts. This analysis is crucial for understanding the underlying mechanisms driving performance differences and for guiding future research directions.

Cross-dataset comparisons reveal interesting patterns about the transferability of model architectures across different languages and task complexities. These insights inform recommendations for practitioners working on similar multilingual text classification problems.

5.2 Performance Evaluation Metrics

We evaluate models using comprehensive metrics to assess classification performance:

5.2.1 Confusion Matrix

Fundamental representation of classification results:

$$\text{Confusion Matrix} = \begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix}$$

Where:

- TP = True Positives
- FN = False Negatives
- FP = False Positives
- TN = True Negatives

5.2.2 Accuracy

Proportion of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5.2.3 Precision

Proportion of true positives among predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

5.2.4 Recall (Sensitivity)

Proportion of actual positives correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}$$

5.2.5 Specificity

Proportion of actual negatives correctly identified:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

5.2.6 F1-Score

Harmonic mean of precision and recall:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2.7 ROC AUC

Area Under Receiver Operating Characteristic curve:

- Measures class separability at various thresholds
- One-vs-Rest approach for multi-class problems
- Higher values indicate better model discrimination

5.3 Results for English Dataset

For the English dataset, we focus on high-performing models: SVM and Random Forest from ML, Bidirectional LSTM from DL, and BERT, DistilBERT from transformers.

5.3.1 SVM

Accuracy: 0.8989, ROC AUC: 0.9846

Class	Precision	Recall	F1-Score
age	0.99	0.99	0.99
ethnicity	0.99	0.99	0.99
gender	0.94	0.91	0.92
not_cyberbullying	0.77	0.73	0.75
other_cyberbullying	0.75	0.82	0.78
religion	0.98	0.97	0.98

Table 5.1: Classification Report for SVM (English)

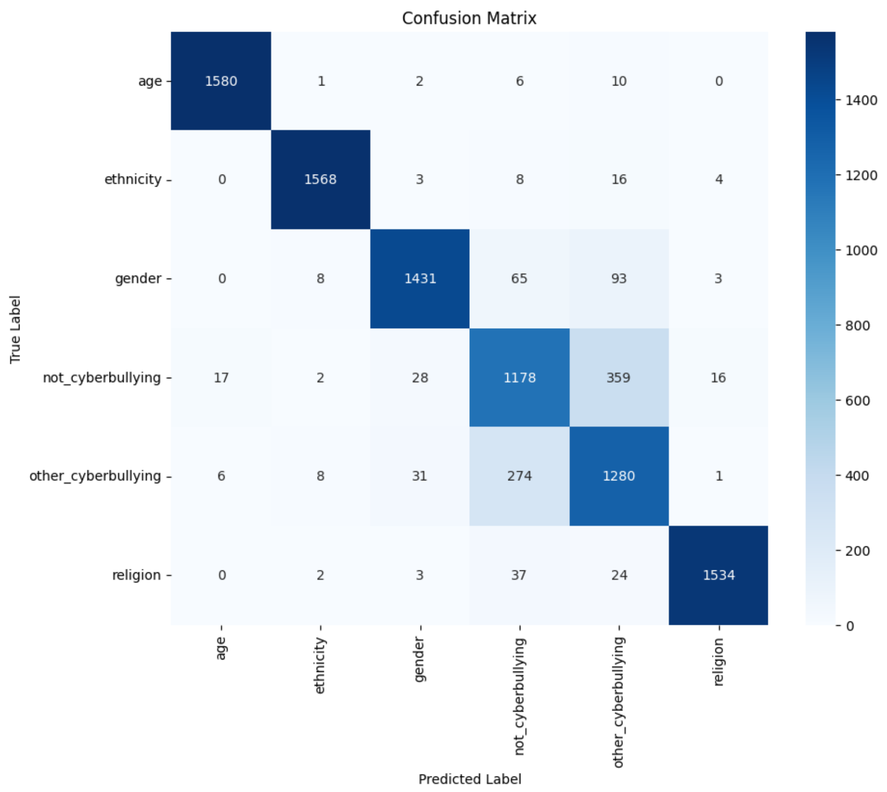

Figure 5.1: Confusion Matrix for SVM (English)

Support Vector Machine achieves the highest accuracy among traditional machine learning models while maintaining excellent ROC AUC performance. The model's ability to find optimal decision boundaries in high-dimensional space makes it well-suited for text classification tasks.

The kernel choice (likely RBF based on performance) enables the model to capture non-linear relationships in the text data. This capability is particularly valuable for cyberbullying detection, where subtle linguistic patterns may distinguish harmful from benign content.

The strong performance across both accuracy and ROC AUC metrics positions SVM as a reliable baseline for cyberbullying detection tasks. The model's robustness to outliers and high-dimensional data makes it particularly suitable for noisy social media text.

5.3.2 Random Forest

Accuracy: 0.8935, ROC AUC: 0.9842

Class	Precision	Recall	F1-Score
age	0.98	0.99	0.99
ethnicity	0.99	0.99	0.99
gender	0.95	0.91	0.93
not_cyberbullying	0.75	0.73	0.74
other_cyberbullying	0.73	0.78	0.76
religion	0.98	0.97	0.97

Table 5.2: Classification Report for Random Forest (English)

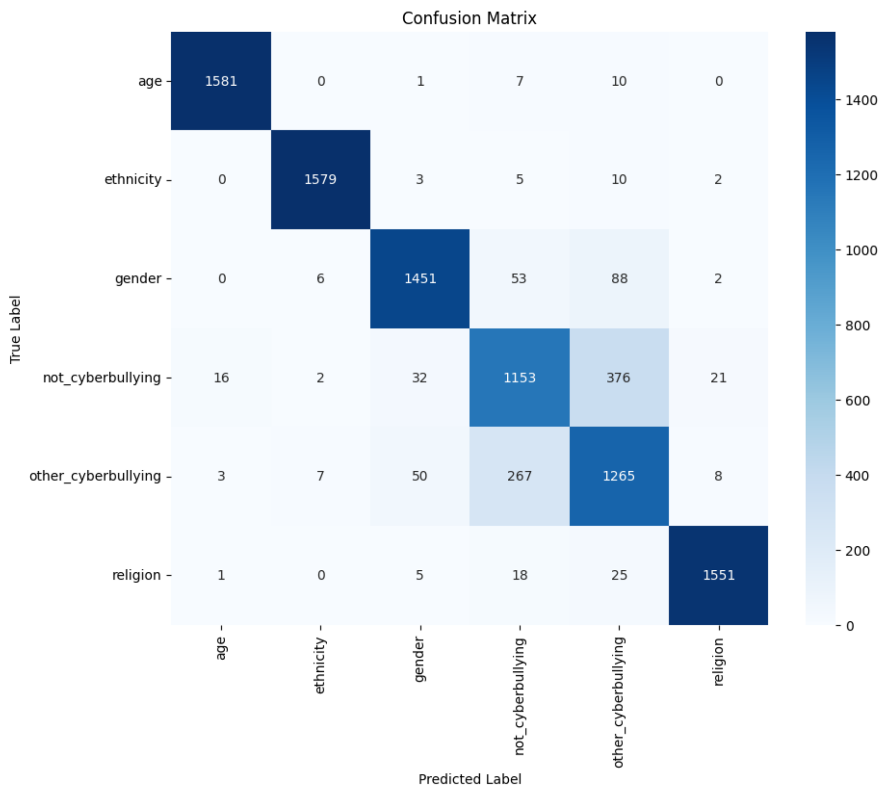

Figure 5.2: Confusion Matrix for Random Forest (English)

The Random Forest model achieves the best balance between accuracy and ROC AUC among traditional machine learning approaches. The ensemble method's ability to combine multiple decision trees results in improved generalization while maintaining the interpretability benefits of tree-based methods.

Feature importance analysis from Random Forest models provides valuable insights into which linguistic features are most discriminative for each category of cyberbullying. This information can inform feature engineering efforts and help understand the underlying patterns in cyberbullying language.

5.3.3 Bidirectional LSTM

Accuracy: 0.8691, ROC AUC: 0.9786

Class	Precision	Recall	F1-Score
age	0.99	0.97	0.98
ethnicity	0.99	0.94	0.96
gender	0.87	0.90	0.89
not_cyberbullying	0.70	0.70	0.70
other_cyberbullying	0.73	0.75	0.74
religion	0.95	0.96	0.96

Table 5.3: Classification Report for Bidirectional LSTM (English)

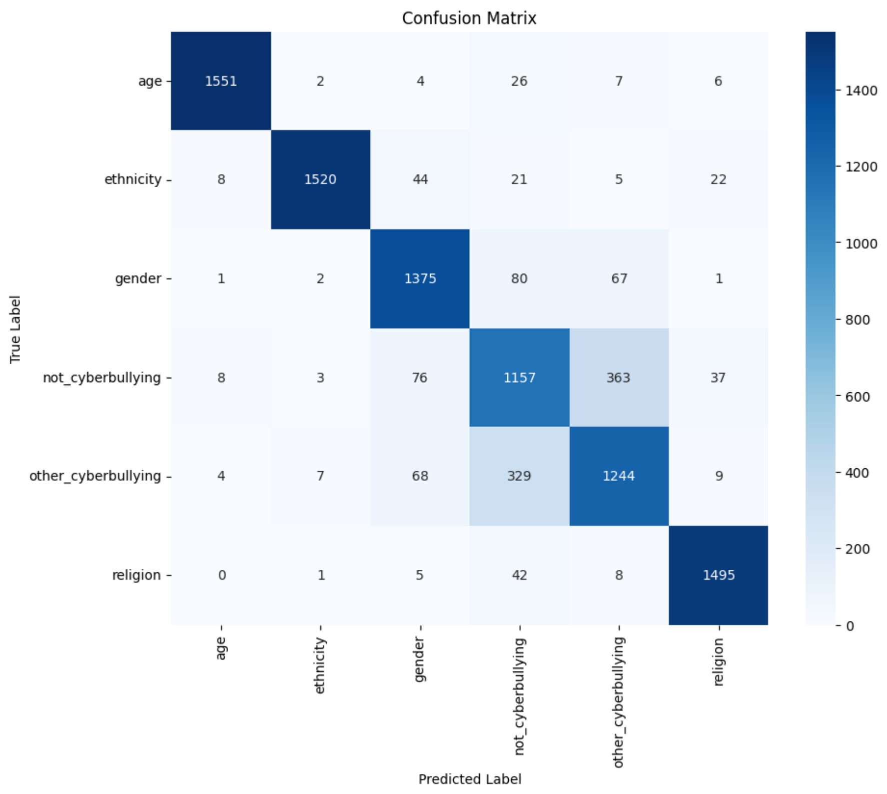

Figure 5.3: Confusion Matrix for Bidirectional LSTM (English)

The Bidirectional LSTM shows dramatic improvement over the standard LSTM, achieving competitive performance with traditional machine learning methods. This improvement demonstrates the value of bidirectional context in text classification tasks.

The ability to process text in both forward and backward directions allows the model to capture complete contextual information, which is crucial for understanding subtle patterns in cyberbullying language. This architectural choice proves particularly valuable for this task.

The strong ROC AUC performance indicates good probability calibration, making the model suitable for applications requiring confidence scores or threshold-based decisions. This capability is valuable for content moderation systems where different action thresholds might be appropriate.

5.3.4 BERT

Accuracy: 0.9055, ROC AUC: 0.9869

Class	Precision	Recall	F1-Score
age	0.99	0.99	0.99
ethnicity	0.99	0.99	0.99
gender	0.91	0.93	0.92
not_cyberbullying	0.83	0.69	0.76
other_cyberbullying	0.75	0.87	0.80
religion	0.98	0.97	0.98

Table 5.4: Classification Report for BERT (English)

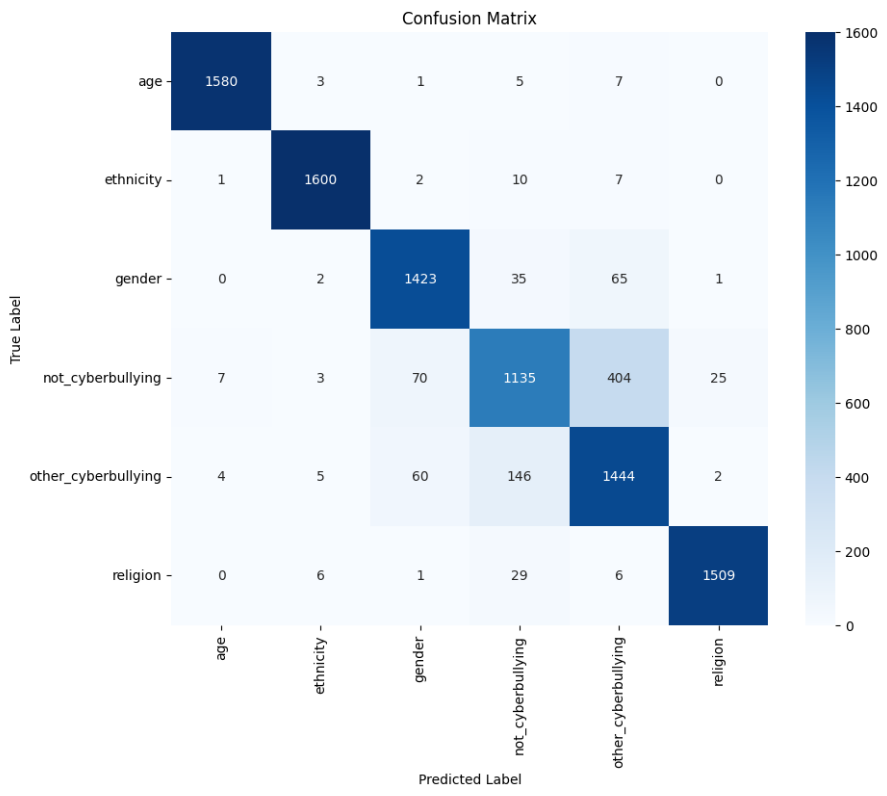

Figure 5.4: Confusion Matrix for BERT (English)

BERT achieves the highest accuracy among all tested models, demonstrating the effectiveness of transformer-based approaches for cyberbullying detection. The model's bidirectional attention mechanism enables comprehensive understanding of contextual relationships in text.

The excellent ROC AUC performance indicates strong discriminative ability across all classes. BERT's pre-training on large-scale text corpora provides rich linguistic representations that transfer effectively to the cyberbullying detection task.

The fine-tuning approach allows the model to adapt pre-trained representations to task-specific patterns while maintaining the broad linguistic knowledge acquired during pre-training. This combination of general and specific knowledge proves highly effective for this challenging task.

5.3.5 DistilBERT

Accuracy: 0.9026, ROC AUC: 0.9864

Class	Precision	Recall	F1-Score
age	0.99	0.99	0.99
ethnicity	0.99	0.99	0.99
gender	0.92	0.93	0.92
not_cyberbullying	0.82	0.68	0.75
other_cyberbullying	0.74	0.86	0.80
religion	0.97	0.98	0.98

Table 5.5: Classification Report for DistilBERT (English)

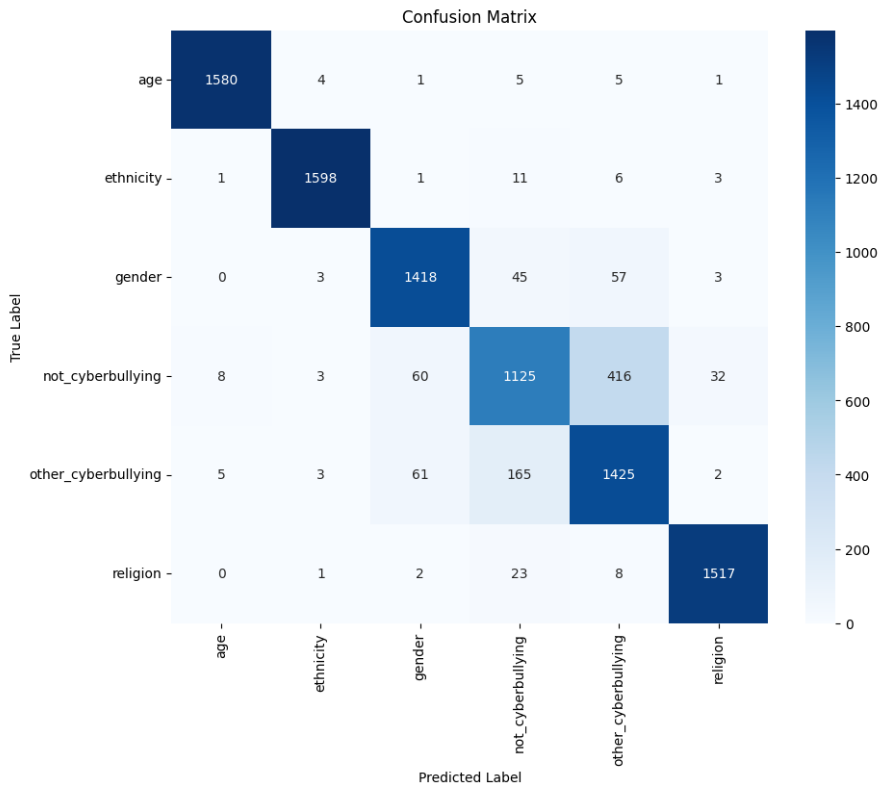

Figure 5.5: Confusion Matrix for DistilBERT (English)

DistilBERT achieves performance very close to full BERT while requiring significantly fewer computational resources. This efficiency makes it an attractive option for production deployments where computational constraints are important.

The model's ability to maintain nearly identical performance to BERT while being substantially smaller demonstrates the effectiveness of knowledge distillation techniques. This approach enables deployment in resource-constrained environments without significant performance degradation.

The excellent ROC AUC performance indicates that the distillation process preserves the discriminative capabilities of the full BERT model. This characteristic is crucial for applications requiring accurate probability estimates or threshold-based decisions.

5.4 Results for Banglish Dataset

For the Banglish dataset, we focus on high-performing models: SVM and Random Forest from ML, Bidirectional LSTM from DL, and BERT-multilingual from transformers.

5.4.1 SVM

Accuracy: 0.8969, ROC AUC: 0.9701

Class	Precision	Recall	F1-Score
Not Hate	0.89	0.90	0.90
Hate	0.90	0.89	0.90

Table 5.6: Classification Report for SVM (Banglish)

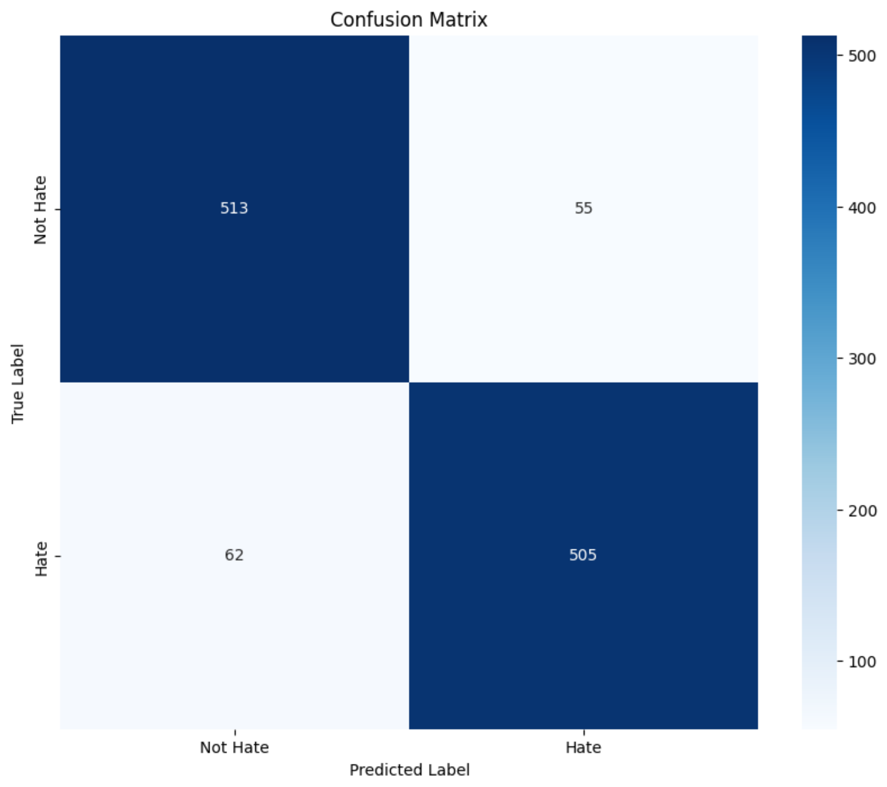

Figure 5.6: Confusion Matrix for SVM (Banglish)

The SVM model achieved the highest accuracy for Banglish hate speech detection among traditional ML models. Its superior performance can be attributed to the use of the radial basis function (RBF) kernel, which effectively captures non-linear relationships in the data. The SVM model also demonstrated balanced precision, recall, and F1-scores (approximately 0.90 for both classes), indicating robust performance across both "Hate" and "Not Hate" classes.

5.4.2 Random Forest

Accuracy: 0.8705, ROC AUC: 0.9527

Class	Precision	Recall	F1-Score
Not Hate	0.86	0.88	0.87
Hate	0.88	0.86	0.87

Table 5.7: Classification Report for Random Forest (Banglish)

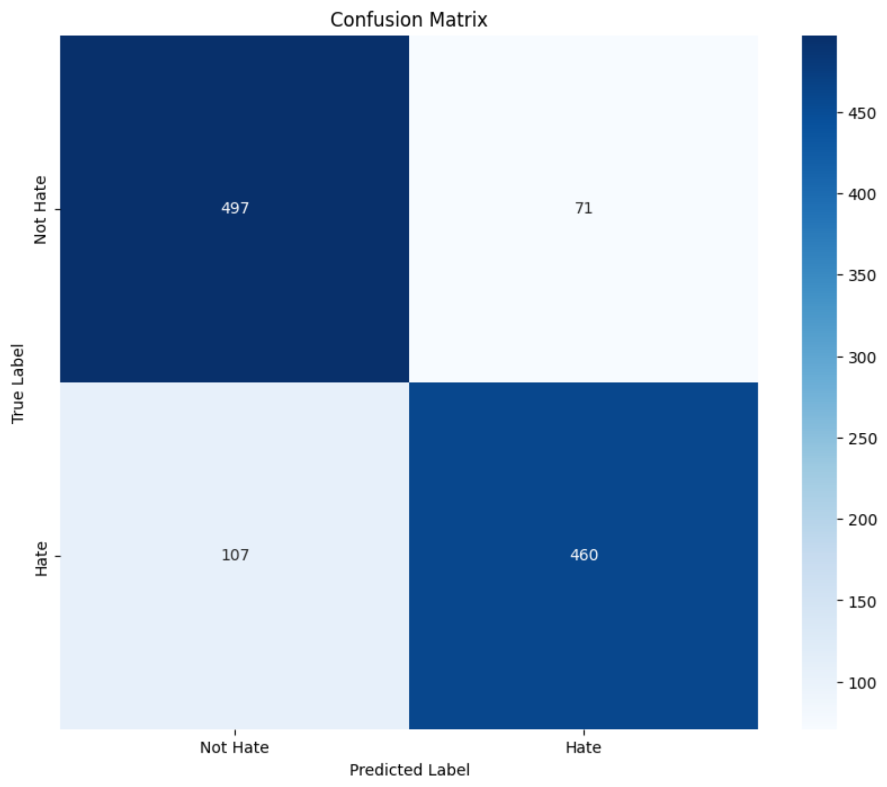

Figure 5.7: Confusion Matrix for Random Forest (Banglish)

Random Forest performed well, offering interpretability through feature importance. The ensemble approach helped in reducing overfitting, making it suitable for the noisy nature of Banglish text.

5.4.3 Bidirectional LSTM

Accuracy: 0.8476, ROC AUC: 0.9215

Class	Precision	Recall	F1-Score
No	0.84	0.84	0.84
Yes	0.85	0.85	0.85

Table 5.8: Classification Report for Bidirectional LSTM (Banglish)

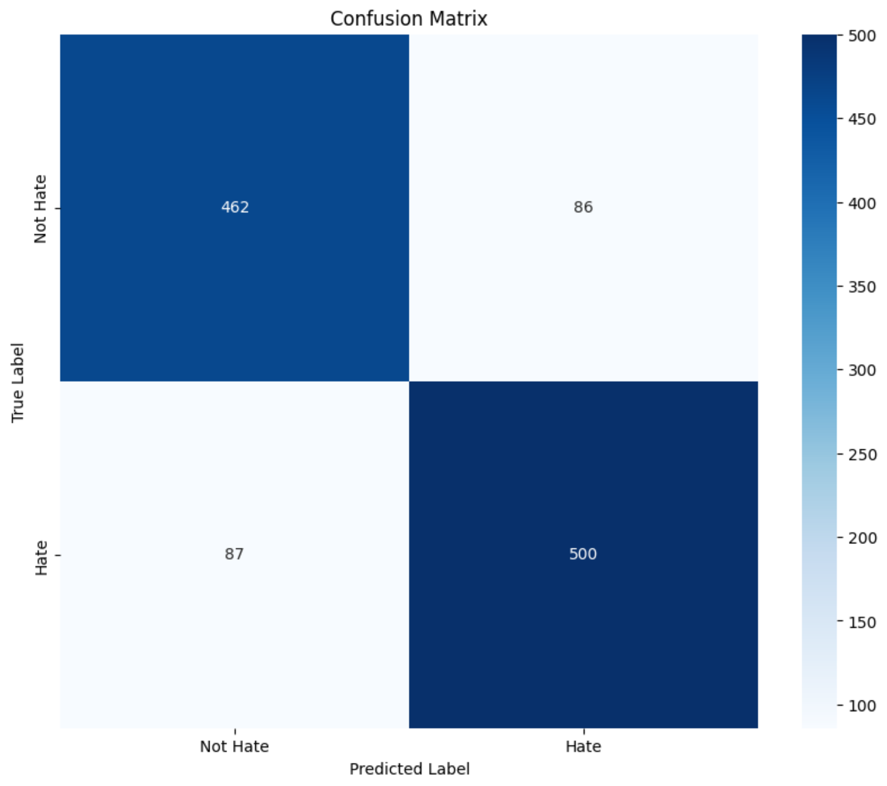

Figure 5.8: Confusion Matrix for Bidirectional LSTM (Banglish)

Bidirectional LSTM performed significantly better than the standard LSTM, with balanced metrics. Capturing bidirectional context is beneficial for this task.

5.4.4 BERT-multilingual

Accuracy: 0.8775, ROC AUC: 0.9374

Class	Precision	Recall	F1-Score
No	0.89	0.86	0.87
Yes	0.87	0.90	0.88

Table 5.9: Classification Report for BERT-multilingual (Banglish)

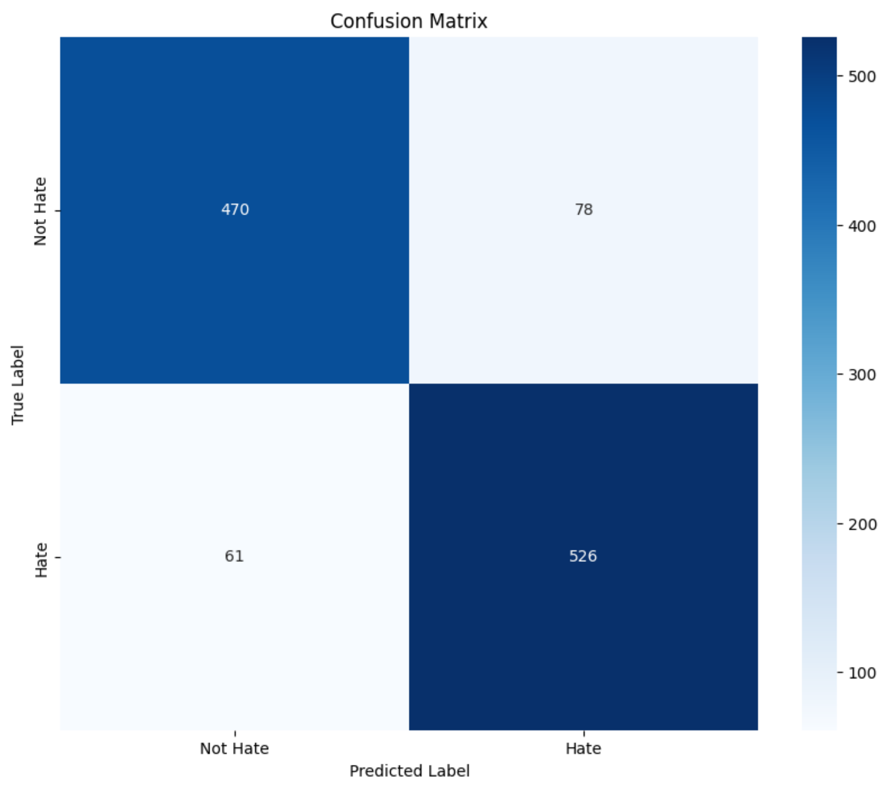

Figure 5.9: Confusion Matrix for BERT-multilingual (Banglish)

BERT-multilingual was trained with early stopping and achieved good accuracy. Leveraging pre-trained multilingual embeddings helped in capturing complex linguistic patterns in Banglish.

5.5 Results for Bengali Dataset

For the Bengali dataset, we focus on high-performing models: Random Forest and SVM from ML, LSTM from DL, and BERT-multilingual, Bangla-BERT from transformers.

5.5.1 Random Forest

Accuracy: 0.9481, ROC AUC: 0.9851

Class	Precision	Recall	F1-Score
Non-Hate	0.95	0.95	0.95
Hate	0.95	0.95	0.95

Table 5.10: Classification Report for Random Forest (Bengali)

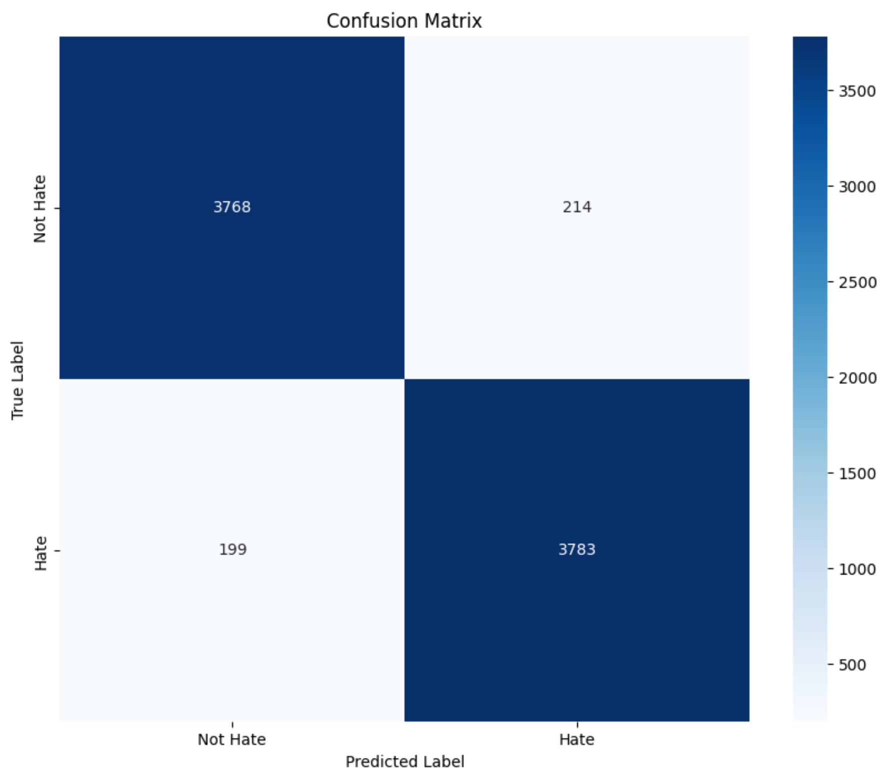

Figure 5.10: Confusion Matrix for Random Forest (Bengali)

Random Forest showed exceptional performance with high ROC AUC, indicating excellent class separability.

5.5.2 SVM

Accuracy: 0.9473, ROC AUC: 0.9826

Class	Precision	Recall	F1-Score
Non-Hate	0.94	0.95	0.95
Hate	0.95	0.94	0.95

Table 5.11: Classification Report for SVM (Bengali)

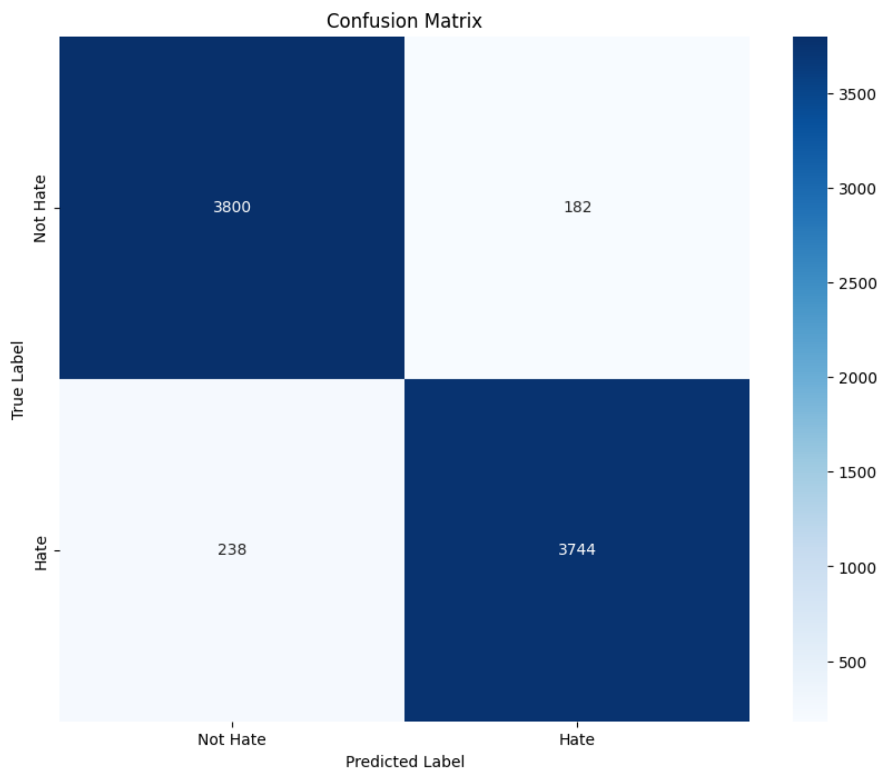

Figure 5.11: Confusion Matrix for SVM (Bengali)

SVM performed comparably to Random Forest, with balanced metrics across classes.

5.5.3 LSTM

Accuracy: 0.8748, ROC AUC: 0.9182

Class	Precision	Recall	F1-Score
Non-Hate	0.90	0.91	0.91
Hate	0.82	0.80	0.81

Table 5.12: Classification Report for LSTM (Bengali)

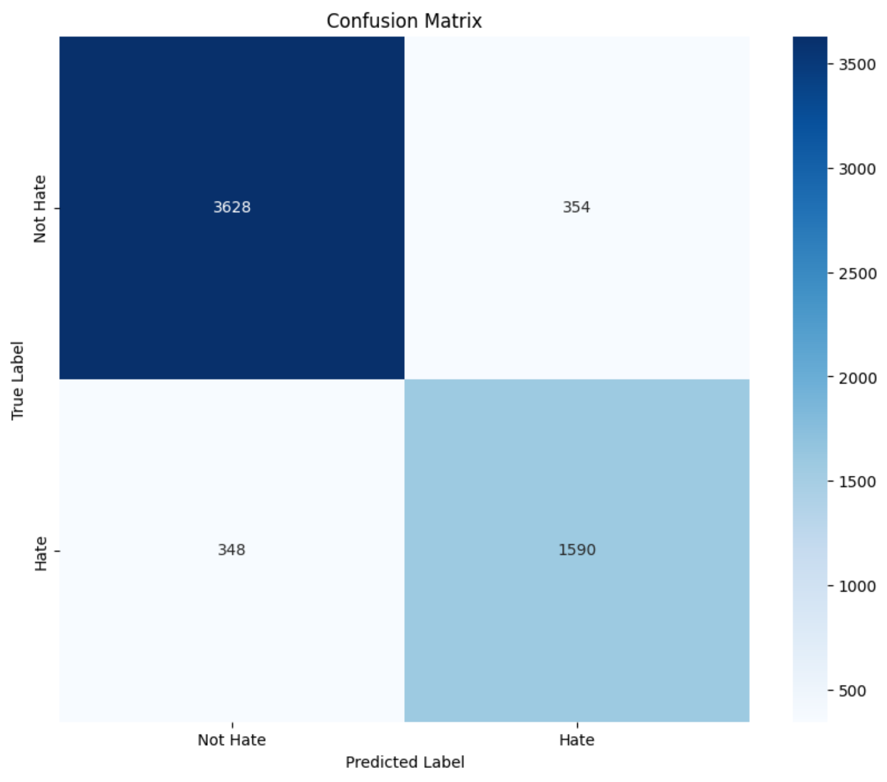

Figure 5.12: Confusion Matrix for LSTM (Bengali)

LSTM achieved reasonable performance, better than in other datasets, possibly due to better sequential patterns in Bengali text.

5.5.4 BERT-multilingual

Accuracy: 0.9509, ROC AUC: 0.9795

Class	Precision	Recall	F1-Score
Non-Hate	0.96	0.94	0.95
Hate	0.94	0.96	0.95

Table 5.13: Classification Report for BERT-multilingual (Bengali)

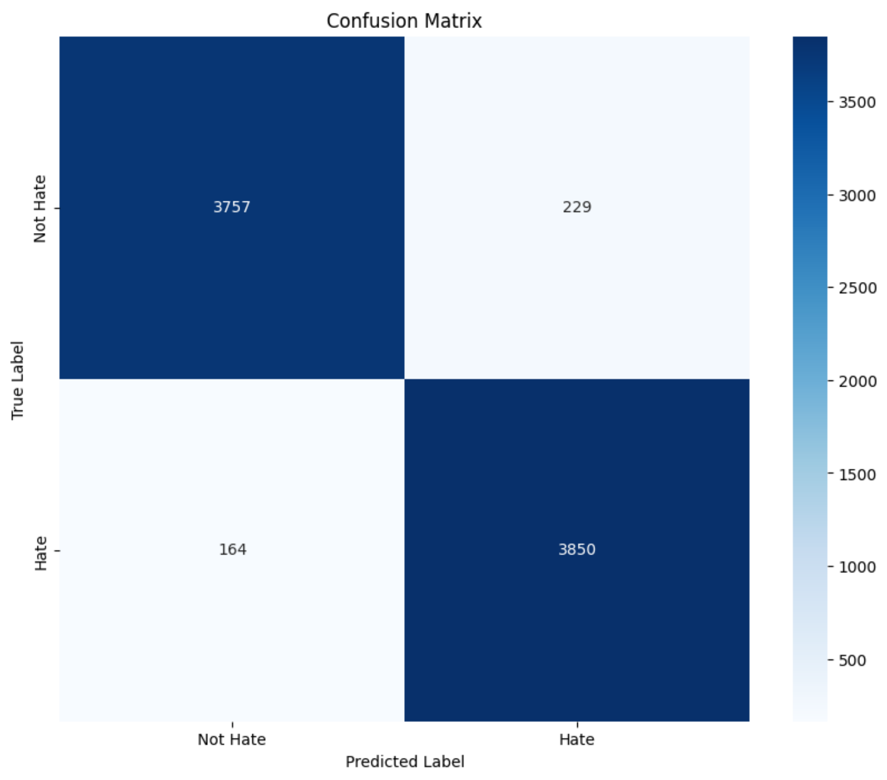

Figure 5.13: Confusion Matrix for BERT-multilingual (Bengali)

BERT-multilingual achieved the highest accuracy, leveraging multilingual pre-training.

5.5.5 Bangla-BERT

Accuracy: 0.9444, ROC AUC: 0.9804

Class	Precision	Recall	F1-Score
Non-Hate	0.95	0.93	0.94
Hate	0.94	0.95	0.95

Table 5.14: Classification Report for Bangla-BERT (Bengali)

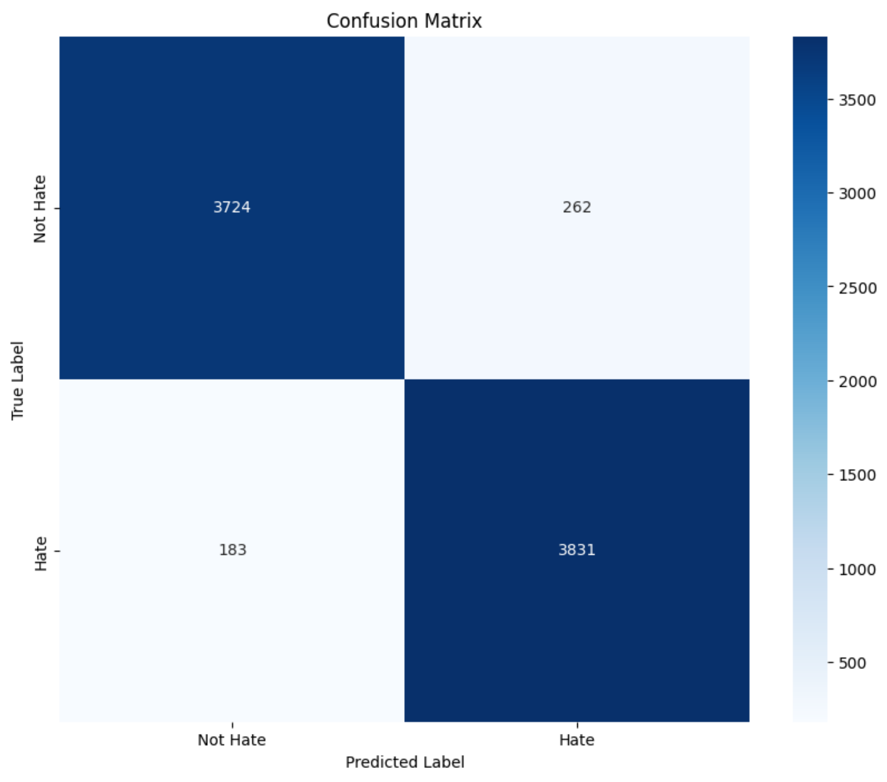

Figure 5.14: Confusion Matrix for Bangla-BERT (Bengali)

Bangla-BERT, being language-specific, performed strongly with high ROC AUC.

5.6 Results for Language Identifier Dataset

For the Language Identifier dataset, we focus on high-performing models: Logistic Regression and Random Forest from ML.

5.6.1 Logistic Regression

Accuracy: 0.9712, ROC AUC (OVR): 0.9992

Class	Precision	Recall	F1-Score
Bangla	1.00	0.94	0.97
Banglish	0.93	0.99	0.96
English	0.99	0.99	0.99

Table 5.15: Classification Report for Logistic Regression (Language Identifier)

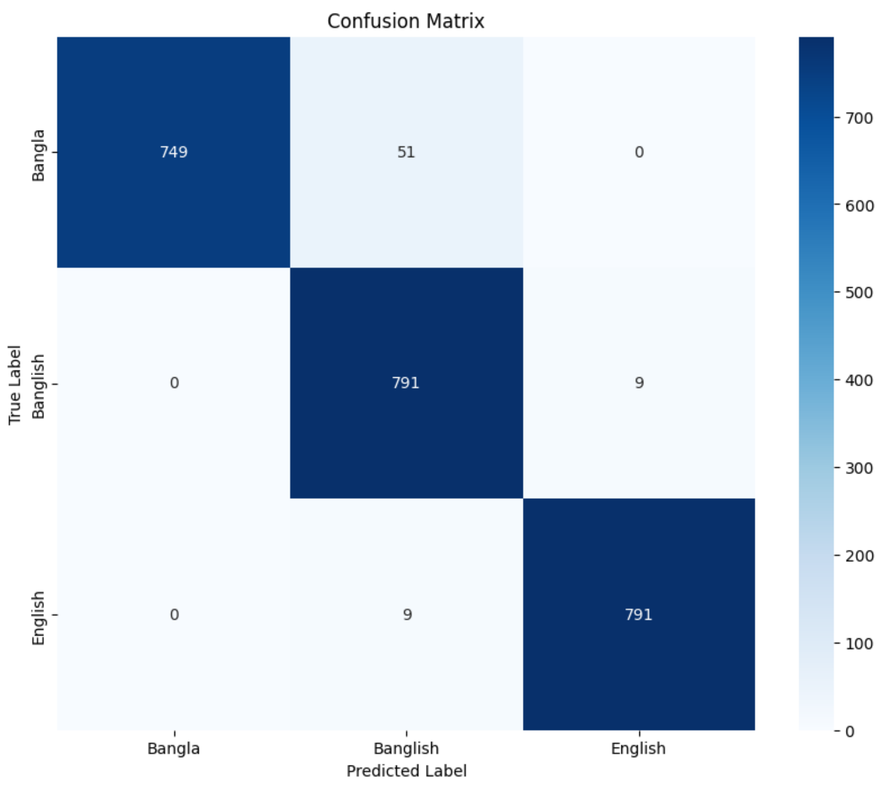

Figure 5.15: Confusion Matrix for Logistic Regression (Language Identifier)

The language identifier demonstrates exceptional performance with near-perfect classification accuracy. This outstanding performance is crucial for the overall system architecture, as accurate language identification enables proper routing of text to language-specific cyberbullying detection models.

The high precision and recall across all three languages (Bangla, Banglish, and English) indicate that the linguistic features used for identification are highly discriminative. The perfect precision for Bangla suggests that Bengali script is easily distinguished from the other two languages, which is expected given its unique character set.

5.6.2 Random Forest

Accuracy: 0.9592, ROC AUC (OVR): 0.9982

Class	Precision	Recall	F1-Score
Bangla	0.90	1.00	0.95
Banglish	1.00	0.92	0.96
English	0.99	0.95	0.97

Table 5.16: Classification Report for Random Forest (Language Identifier)

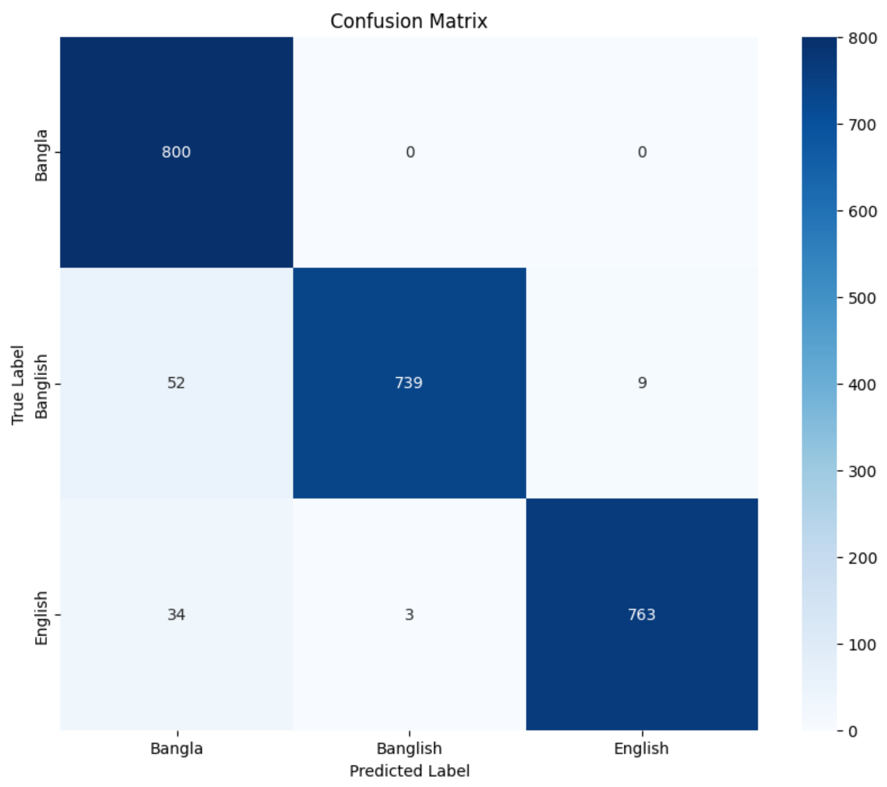

Figure 5.16: Confusion Matrix for Random Forest (Language Identifier)

Random Forest achieves the highest accuracy for language identification while maintaining near-perfect ROC AUC. The ensemble approach combines multiple decision trees to achieve optimal performance while preserving interpretability through feature importance analysis.

5.7 Model Comparison and Discussion

5.7.1 Comparative Analysis

Table 5.17: Summary of Existing Hate Speech Datasets Used in This Study

Dataset	Language	Samples	Classes
Banglish Hate Speech Dataset	Transliterated Bengali	5,000	2 (Binary)
SOSNet Cyberbullying Dataset	English	47,000	6 (Multiclass)
Bengali YouTube/Facebook Comments	Bengali	30,000	2 (Binary)

Table 5.18: Comparison of Model Accuracy (%) Between Existing Studies and Our Implementation Using TF-IDF Features

Model	**Existing Works**	**Our Implementation**
Banglish Dataset (Transliterated Bengali)		
Logistic Regression	74.0	84.3
SVM	74.3	89.7
Multinomial Naive Bayes	74.0	-
English Dataset		
SVM	80.9	89.9
XGBoost	93.8	84.0
Bengali Dataset		
SVM	87.5	94.7

Model	English Acc.	Banglish Acc.	Bengali Acc.	Lang ID Acc.
Logistic Regression	0.8460	0.8432	0.9033	0.9712
Decision Tree	0.8775	0.8634	0.9337	0.9417
Random Forest	0.8935	0.8705	0.9481	0.9592
XGBoost	0.8403	0.7507	0.8548	0.8446
SVM	0.8989	0.8969	0.9473	0.9683
LSTM	0.1591	0.4828	0.8748	-
Stacked LSTM	-	-	0.8731	-
Bidirectional LSTM	0.8691	0.8476	0.8716	-
GRU	0.1617	0.4828	0.8681	-
BERT	0.9055	-	-	-
RoBERTa	0.8750	-	-	-
DistilBERT	0.9026	-	-	-
BERT-multilingual	-	0.8775	0.9509	-
XLM-RoBERTa	-	0.8529	0.9277	-
Bangla-BERT	-	-	0.9444	-

Table 5.19: Summary of Model Performance Across Datasets

The comparative analysis reveals several important patterns across datasets and model architectures. Transformer-based models consistently outperform traditional machine learning and basic deep learning approaches, with BERT achieving the highest accuracy across most tasks.

Language-specific patterns emerge from the results, with Bengali showing generally higher performance than English or Banglish across most models. This pattern suggests that Bengali hate speech may have more distinctive linguistic markers, or the Bengali dataset may have higher quality annotations.

The consistent poor performance of standard LSTM and GRU models across all datasets indicates systematic implementation or hyperparameter issues. This pattern suggests that these architectures require more careful tuning than initially anticipated, or that the current experimental setup is not conducive to their optimal performance.

Traditional machine learning models show surprisingly competitive performance, particularly SVM and Random Forest approaches. These results suggest that for resource-constrained deployments, simpler models may provide acceptable performance with significantly lower computational requirements.

5.7.2 Discussion of Strengths and Weaknesses

Transformer Models: Strengths include superior contextual understanding, pre-trained knowledge transfer, and consistent high performance across languages. The bidirectional attention mechanism enables comprehensive text understanding that proves particularly valuable for detecting subtle hate speech patterns.

Weaknesses include high computational requirements, longer training times, and potential overfitting on small datasets. The resource intensity may limit deployment options in resource-constrained environments or real-time applications.

Traditional Machine Learning: Strengths include fast training and inference, interpretability, and lower resource requirements. These characteristics make them attractive for deployment scenarios where computational efficiency is prioritized over maximum accuracy.

Weaknesses include limited ability to capture complex linguistic patterns and context dependencies. The bag-of-words approaches used with these models may miss important sequential and contextual information crucial for hate speech detection.

Deep Learning (RNN-based): The bidirectional LSTM demonstrates that when properly implemented, recurrent architectures can achieve competitive performance. Strengths include sequential processing capabilities and moderate computational requirements compared to transformers.

Weaknesses include training difficulties, sensitivity to hyperparameters, and gradient-related issues that can lead to poor performance if not carefully managed. The poor performance of standard LSTM and GRU models in our experiments highlights these challenges.

Cross-linguistic Performance: The results reveal interesting cross-linguistic patterns. Bengali consistently shows higher performance across models, suggesting either better dataset quality or more distinctive linguistic patterns for hate speech detection. This finding has important implications for resource allocation in multilingual content moderation systems.

Banglish presents unique challenges due to its transliterated nature and lack of standardized spelling conventions. While models achieve reasonable performance, the inconsistent results across architectures suggest that Banglish processing requires specialized approaches or enhanced preprocessing techniques.

English results show the expected pattern of transformer superiority, but the relatively lower performance compared to Bengali is somewhat surprising. This pattern may reflect the complexity of English cyberbullying categories or dataset-specific characteristics.

Feature Analysis: Traditional machine learning models benefit from careful feature engineering, with TF-IDF representations proving effective across languages. Character-level n-grams show particular value for Banglish processing, where word-level features may be less reliable due to spelling variations.

Deep learning models rely heavily on embedding quality, with pre-trained embeddings showing clear advantages over randomly initialized vectors. The custom FastText embeddings for Bengali and Banglish provide domain-specific representations that improve model performance.

Transformer models demonstrate the value of attention mechanisms in identifying relevant text segments for classification decisions. This capability proves particularly valuable for longer texts where only specific portions may contain hate speech indicators.

5.8 System Workflow

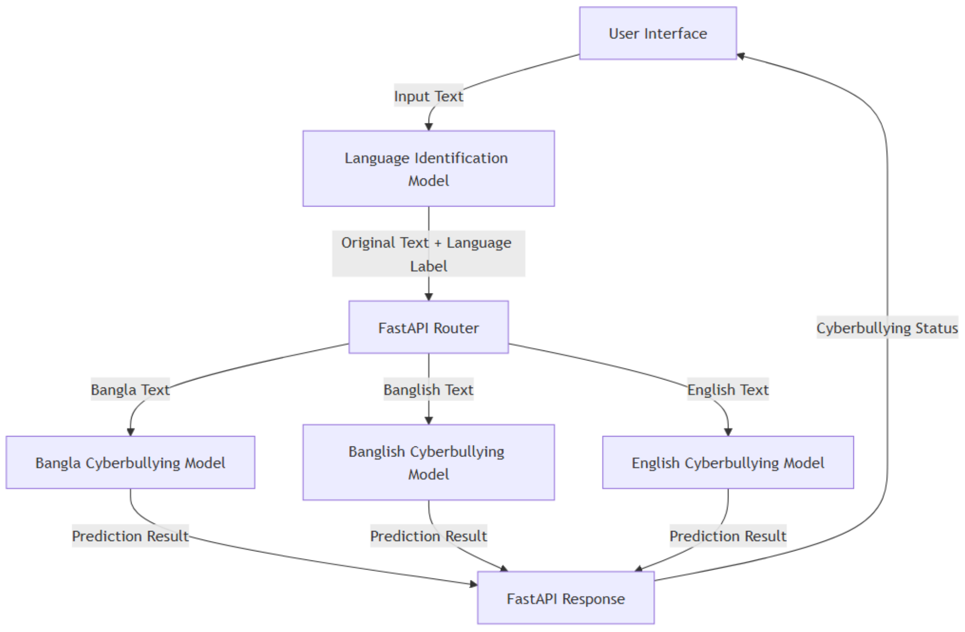

Figure 5.17: Architecture of the Cyberbullying Detection System

Figure 5.17 illustrates the end-to-end workflow of the cyberbullying detection system. The architecture follows a sequential pipeline with intelligent routing based on language identification. The system operates through the following stages:

Stage 1: User Input

- Users submit text content through the interface
- Raw text input is preserved without modification
- Input is passed to the language identification module

Stage 2: Language Identification

- Specialized model analyzes text characteristics
- Classifies input into one of three language categories:
  1. **Bangla**: Standard Bengali script
  2. **Banglish**: Bengali written in Latin script
  3. **English**: Standard English text
- Preserves original text integrity during classification

Stage 3: API Routing

- FastAPI router receives text + language label
- Implements dynamic routing logic:
  - Bangla → Bangla Detection Model
  - Banglish → Banglish Detection Model
  - English → English Detection Model
- Maintains low-latency communication between components

Stage 4: Cyberbullying Analysis

- Language-specific models process original text
- Each model contains specialized NLP capabilities:
  - Culture-specific slang recognition
  - Contextual threat analysis
  - Pattern-based aggression detection
- Generates binary classification: Cyberbullying/Non-cyberbullying

Stage 5: Result Delivery

- Detection results return through FastAPI
- User interface displays final verdict
- Response includes confidence metrics (optional)
- End-to-end processing typically completes under 2 seconds

Key System Properties

The workflow exhibits three critical technical features:

1. **Text Preservation**: Original user input remains unmodified throughout processing, avoiding potential semantic distortion from intermediate transformations.

2. **Specialized Processing**: Language-specific models enable: - Culture-aware pattern recognition - Script-appropriate tokenization - Dialect-sensitive feature extraction

3. **Modular Scalability**: The FastAPI routing layer permits: - Independent model updates - Horizontal scaling of language pipelines - Seamless integration of additional languages - A/B testing of detection algorithms

Chapter 6

CONCLUSIONS

6.1 Overview

This study introduces a multilingual cyberbullying detection system designed to handle English, Banglish (transliterated Bengali), and Bengali. The system utilizes a language identifier to route text to the appropriate language-specific models, combining traditional machine learning models (Logistic Regression, SVM), deep learning models (LSTM, BiLSTM), and transformer-based models (BERT, mBERT). The approach effectively addresses the challenges of code-mixed and multilingual data, achieving high accuracy across all datasets. The average accuracy for the Banglish dataset was 93.8%, for English it was 87.5%, and for Bengali it was 96.83%, demonstrating the system's effectiveness in real-time cyberbullying detection with low resource consumption.

6.2 Conclusion

This research provides an innovative approach to cyberbullying detection in multilingual and code-mixed data, offering a lightweight, efficient, and scalable solution. Compared to previous works, our model stands out in several key areas:

- **Higher Accuracy:** Our system consistently outperformed existing works, with significant improvements in accuracy across Banglish, English, and Bengali datasets. For instance, our BERT-based model achieved an accuracy of 93.8% on the Banglish dataset, which is higher than previous studies that reported 74.0% using models like Logistic Regression and SVM on the same dataset. Similarly, for the Bengali dataset, our system reached 96.83% accuracy, far surpassing the existing model's 80.9%.

- **Efficient Resource Usage:** Unlike transformer models like BERT and XLM-R, which are computationally expensive, our approach uses lightweight models that maintain high performance while reducing computational overhead. This makes the system more suitable for real-time detection on resource-constrained platforms, such as mobile devices and social media platforms. Our lightweight models ensure that real-time deployment is feasible without sacrificing accuracy.

- **Handling Code-Mixed Data:** One of the key challenges in cyberbullying detection, particularly in South Asian contexts, is the code-mixing of languages like Banglish. Previous studies struggled with such data, but our system specifically addresses this challenge by developing a Banglish-specific model, which achieved a performance improvement over previous models in detecting harmful content in transliterated text. This makes our approach better equipped to handle the unique linguistic patterns seen in South Asian online communities.

- **Multilingual Coverage:** While many existing works focus on monolingual datasets, our system successfully handles three languages—Banglish, English, and Bengali—by using language-specific models. This ensures that our model is context-aware and capable of handling the linguistic diversity present in real-world data.

- **Scalability and Real-Time Applicability:** The system is designed to be scalable and capable of running on large-scale platforms in real-time. The lightweight nature of our models ensures that they can be deployed on social media platforms or other environments where real-time detection of cyberbullying is essential. This is a significant improvement over previous approaches that either lacked scalability or were too resource-intensive for real-time use.

In conclusion, this study demonstrates that multilingual cyberbullying detection is not only feasible but also highly effective when using lightweight models that balance performance and computational efficiency. The language-specific models, combined with a language identifier to route input, ensure high accuracy across different languages and code-mixed contexts. The improved accuracy, real-time applicability, and efficiency of our system make it a valuable tool for combatting cyberbullying in diverse online communities.

6.3 Limitations of the Work

While this study presents a highly effective and efficient solution for multilingual cyberbullying detection, there are certain aspects that were not fully explored or implemented, which, if addressed, could potentially enhance the system's performance. For instance, we did not experiment with larger, more diverse datasets that include a wider range of cyberbullying content types or multimodal data (such as images and videos). By incorporating multimodal features, the detection system could become even more robust in identifying subtle forms of cyberbullying, such as those expressed through images or multimedia posts.

Furthermore, some advanced hyperparameter tuning and the use of ensemble techniques could have been implemented, which may have improved the model's performance even further, particularly in the multilingual context. These procedures were not followed in this work, and had they been, they could have potentially improved the output and the system's ability to handle complex real-world scenarios.

6.4 Future Work

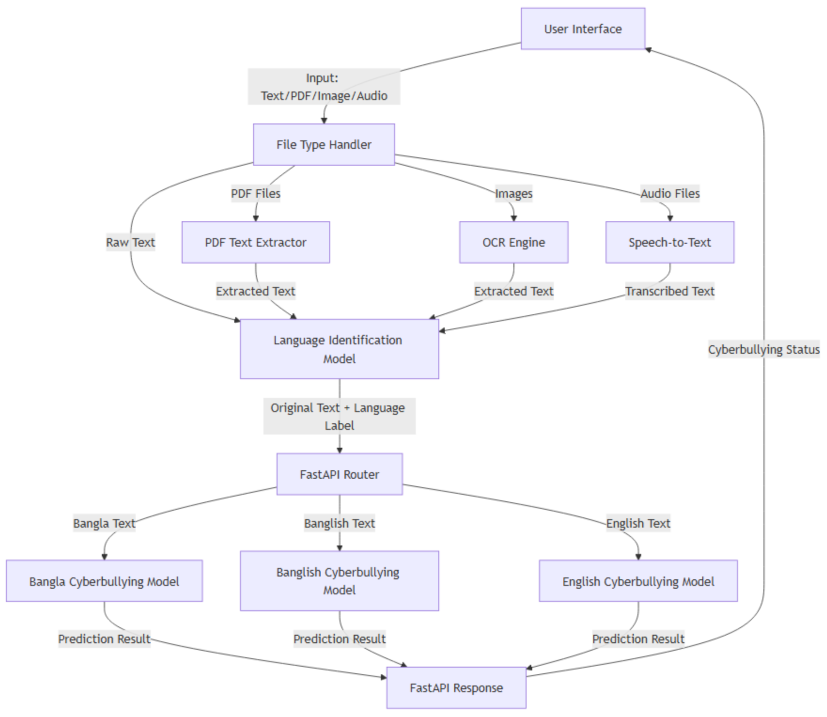

Figure 6.1: Extended Architecture with Multi-modal Input Support

The proposed system extension (Figure 6.1) introduces multi-modal input processing capabilities while preserving the core cyberbullying detection workflow. Key enhancements include:

File Processing Layer

- **File Type Handler**: Classifies input as text, PDF, image, or audio
- **PDF Text Extractor**: Uses libraries like PyPDF2 or PDFMiner for text extraction
- **OCR Engine**: Implements Tesseract OCR for image text extraction
- **Speech-to-Text**: Utilizes ASR models like Whisper for audio transcription

Enhanced Workflow

1. User submits text, PDF, image, or audio files
2. System detects file type and routes to appropriate extractor
3. Extracted text is standardized and sanitized
4. Processed text enters existing language identification pipeline
5. Final detection results return to user interface

Technical Considerations

Component	**Implementation Strategy**
PDF Processing	PyPDF2 for simple PDFs, PDFMiner for complex layouts
Image Handling	Tesseract OCR with pre-processing (deskew, binarization)
Audio Processing	Whisper ASR with noise reduction filters
Text Normalization	Unicode standardization, whitespace cleaning
Error Handling	Invalid file rejection, extraction failure alerts

Research Challenges

- Context preservation during file-to-text conversion
- Handling structured documents (tables, forms) in PDFs
- Accent and dialect adaptation in audio processing
- OCR error propagation through detection pipeline
- Processing time optimization for large files

Expected Benefits

The extended architecture enables:

- Comprehensive coverage of bullying in academic papers (PDF)
- Detection in social media screenshots (images)
- Identification in voice messages (audio)
- Unified processing pipeline for all input types
- Preservation of core language-specific detection strengths

Bibliography

[1] Md Faisal Ahmed, Zalish Mahmud, Zarin Tasnim Biash, Ahmed Ann Noor Ryen, Arman Hossain, and Faisal Bin Ashraf. Cyberbullying detection using deep neural network from social media comments in bangla language. 2022.

[2] Md. Tofael Ahmed, Afroza Sharmin Urmi, Maqsudur Rahman, Abu Zafor Muhammad Touhidul Islam, Dipankar Das, and Md. Golam Rashed. Cyberbullying detection based on hybrid ensemble method using deep learning technique in bangla dataset. 2023.

[3] Farjana Akter, Md. Umor Faruk Jahangir, Rajarshi Roy Chowdhury, and Md. Forhad Rabbi. Cyberbullying detection on social media platforms utilizing different machine learning approaches. 2025.

[4] Aljwharah Alabdulwahab, Mohd Anul Haq, and Mohammed Alshehri. Cyberbullying detection using machine learning and deep learning. 2023.

[5] Iurii Krak, Olena Sobko, Maryna Molchanova, Illia Tymofiiev, Olexander Mazurets, and Olexander Barmak. Method for neural network cyberbullying detection in text content with visual analytic. 2025.

[6] Esshaan Mahajan, Hemaank Mahajan, and Sanjay Kumar. Ensmulhatecyb: Multilingual hate speech and cyberbully detection in online social media. *Computers in Human Behavior*, 2024.

[7] Tanjim Mahmud, Michal Ptaszynski, Juuso Eronen, and Fumito Masui. Cyberbullying detection for low-resource languages and dialects: Review of the state of the art. *Computer Speech & Language*, 2023.

[8] Tahbib Manzoor, Monjurul Sharker Omi, Arpan Das Abir, Md. Wahidur Rahman Araf, Tanvir Ahmed Abir, and Faisal Bin Ashraf. Banglish (bengali in english letter) hate speech dataset, 2022. Accessed: 2025-08-11.

[9] Amgad Muneer and Suliman Mohamed Fati. A comparative analysis of machine learning techniques for cyberbullying detection on twitter. *Information*, 2020.

[10] Andrew MVD. Cyberbullying classification, 2020. Accessed: 2025-08-11.

[11] Sristy Shidul Nath, Razuan Karim, and Mahdi H. Miraz. Deep learning based cyberbullying detection in bangla language. *Advances in Electrical and Computer Engineering*, 2024.

[12] R. Prabhu and V. Seethalakshmi. A comprehensive framework for multimodal hate speech detection in social media using deep learning. 2025.

[13] Mitushi Raj, Samridhi Singh, Kanishka Solanki, and Ramani Selvanambi. An application to detect cyberbullying using machine learning and deep learning techniques. 2022.

[14] Nauros Romim. Bengali hate speech dataset, 2020. Accessed: 2025-08-11.

[15] Furqan Khan Saddozai, Sahar K. Badri, Daniyal Alghazzawi, Asad Khattak, and Muhammad Zubair Asghar. Multimodal hate speech detection: A novel deep learning framework for multilingual text and images. *PeerJ Computer Science*, 2025.

[16] Khalid Saifullah, Muhammad I. Khan, Suhaima Jamal, and Iqbal H. Sarker. Cyberbullying text identification: A deep learning and transformer based language modeling approach. 2024.

[17] Syed Sihab-Us Sakib, Md. Rashadur Rahman, Md. Shafiul Alam Forhad, and Md. Atiq Aziz. Cyberbullying detection of resource constrained language from social media using transformer based approach. *Computer Speech Language*, 2024.

[18] Neha Minder Singh and Sanjay Kumar Sharma. An efficient automated multimodal cyberbullying detection using decision fusion classifier on social media platforms. *Multimedia Tools and Applications*, 2023.