Deep Learning-Based Sentiment Detection in Code-Mixed Language: Exploring RNNs and Transformers Architectures

By
Kazi Ayesha Akter
191-15-12883

Prathona Rani Bristy
212-15-14759

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Raja Tariqul Hasan Tusher

Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Tapasy Rabeya

Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project titled **"Deep Learning-Based Sentiment Detection in Code-Mixed Language: Exploring RNNs and Transformers Architectures,"**, submitted by **Kazi Ayesha Akter,** ID No: **191-15-12883** and **Prathona Rani Bristy**, ID No: **212-15-14759** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

BOARD OF EXAMINERS

Dr. S.M Aminul Haque — **Chairman**
Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Ms. Nazmun Nessa Moon — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Abbas Ali Khan — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Zulfiker Mahmud — **External Examiner**
Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Raja Tariqul Hasan Tusher**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

14.9.25

Raja Tariqul Hasan Tusher
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Tapasy Rabeya
Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Kazi Ayesha Akter

Kazi Ayesha Akter
Student ID: 191-15-12883
Department of Computer Science and Engineering
Daffodil International University

Prathona Rani Bristy

Prathona Rani Bristy
Student ID: 212-15-14759
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has as- sisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project (FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Raja Tariqul Hasan Tusher, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Natural language processing (NLP)** carry out this project. His endless pa-tience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Com- puter Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Banglish, the informal hybrid of Bengali and English typed in Latin alphabet, presents its own challenges along with nonstandard spelling and transliteration drift in the form of code-switching. This study aims to create a full pipeline to deal with Banglish from data acquisition and annotation to modeling and analysis over 7 categories (Appearance, Not Hate, Others, Racial, Religious, Sexual, and Slang). In the study, we create and clean a social media corpus, design a preprocessing suite [custom stop word filtering, regex tokenization, and rule-based normalization of spelling variants] tailored for Banglish, and address class imbalance via staged over- and under-sampling to a balanced set of 2,000 instances per class. To understand the model's performance, we test recurrent architectures (LSTM, GRU, BiLSTM, BiGRU) and their hybrids (LSTM+GRU, BiLSTM+BiGRU) against transformer models (mBERT, XLM-RoBERTa) under equal training conditions. The mBERT model shows the best performance (accuracy 0.88, macro-F1 0.87), followed by BiLSTM+BiGRU among RNN models (accuracy 0.84, macro-F1 0.84), whereas XLM-RoBERTa performs (accuracy 0.75, macro-F1 0.74) the worst, which implies that transformers outperform other models for this task. A confusion-matrix analysis reveals that RNNs consistently fail by collapsing ambiguous classes (Not Hate, Others, Sexual) into Appearance. This failure is substantially reduced by mBERT. We conclude that, with Banglish-specific preprocessing and balanced evaluation, multilingual transformers provide the most reliable basis for moderning Banglish content, while under tighter resource constraints, hybrid bidirectional RNNs provide a viable alternative

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides a comprehensive introduction to the project, "Leveraging Small LLMs for Abstractive Bangla News Summarization: A T5-Based Approach," outlining its background, significance, motivation, objectives, methodology, expected outcomes, and the organization of the report. The project addresses the pressing need for automated summarization in the Bangla news ecosystem by developing an abstractive summarization system using small-scale transformer models, specifically small MT5 (300 million parameters) and BT5 Base (247 million parameters). The system generates long summaries (100–200 tokens) for in-depth insights and short summaries (30–50 tokens) for rapid consumption, tackling information overload and advancing natural language processing (NLP) for Bangla, a low-resource language spoken by over 265 million people in Bangladesh and India. By addressing Bangla's unique linguistic challenges and contributing open-source resources, the project aligns with Sustainable Development Goals (SDGs) 4 (Quality Education), 9 (Industry, Innovation, and Infrastructure), and 10 (Reduced Inequalities), fostering inclusivity and technological advancement.

1.1 Introduction

At a time of pandemic, the online universe also beckons us to public debates in South Asia and along with that, the proliferation of morbid and disgusting exchanges, typed in Bangla and English – this is a frugal way of connecting both Bangla and English, with Latin alphabets. Bangla-English code-switching (Banglish) provides a particularly difficult scenario for automatic hate speech detection. The writing system is unregulated (ajke/ajkei/ajker for "today"), transliteration is inconsistent, words are stretched to express emotion (reeee, oooo), and languages are switched within a clause. These features shatter the underlying assumptions of standard NLP pipelines, which assume typographically-normed spelling, stem- or lemma-like morphological generalization, and monolingual context windows. The social cost of classifying it wrong is equally important, too. Leaving bigoted comments unmoderated can turn into serious

harassment, but overly-moderating may suppress benign speech, especially of minority dialects.

This paper takes up this challenge, and develops a detection system tailor-made for Banglish. Our initial task is to build a labeled dataset from reliable sources (YouTube and Twitter), ensuring that manual annotation is extensively performed for the seven pragmatic categories of online toxicity: Appearance, Not Hate, Others, Racial, Religious, Sexual, and Slang. Taking this raw data to some structured form includes a preprocessing pipeline for Banglish: custom stopword removal for removing filler syllables and emphatic elongations; rule-based normalization for fusion of transliteration and spelling variants; and staged over-and under-sampling for class balance. The over- and under-sampling is also a staged process which can deal with the extreme class imbalance often present in social data as well.

Based on this premise, we conduct a comparative analysis for deep sequential models (LSTM and GRU and their bidirectional and hybrid cousins) and transformer models (mBERT and XLM-RoBERTa). The RNN category forms strong baselines for local temporal structure, while transformers additionally enjoy the merits of multilingual pretraining and global self-attention to learn long-term dependendencies across the code-switch point. Our training settings (sequence length, tokenization, optimization, and regularization) are tuned to each target category for a fair comparison.

The main contributions of this work can be fourfold. First, we present a simple and portable toolkit for the preprocessing of Banglish which addresses nonstandard spelling as well as code-switched 'noise'. We secondly break free from a binary hate vs. non-hate framework and present an equitable multilabel UserNER evaluation on seven linguistically-motivated classes. Third, we provide an assessment of state-of-the-art RNNs, bidirectional and hybrid RNN's, and modern transformer models over the same cleanly engineered dataset, and at the same level of training resources. Fourth, we provide a comprehensive Failure Analysis using classification reports and confusion matrices that will indicate which mistake (e.g. the confusion of Not Hate/Other and Appearance, the mechanisms for that mistake and the reasons for making that mistake. Together, these not only demonstrate which model outperforms the others, but indicate what Banglish challenges each model can or cannot handle–findings that are key to building safe and fair moderation systems in code-mixed, multilingual communities.

1.2 Motivation

Why this study is important:

I. **SOCIAL CRY:** Its becoming an incubator of hate speech and abusive language in the Social Media. For the many, toxic comments are yet another vector for mental strain, bullying or harassment. An effective sentiment detection system is one way to improve online spaces.

II. **Language Barriers**: BarriersMedium work in EnglishCC4, and Standard BanglaBut still work is remaining in Banglish. With millions of people using Banglish daily for practical communication, the need to design systems that work with this language mix is very pressing.

III. **Technical challenge:** Banglish is not standardized. People write words in different ways and sentences can be a mish-mash of two languages. It is very difficult for the classical model to understand this. The solution to this problem is far from trivial and this is the motivation for this work.

IV. **Future Work:** A system that more-precisely finds/identifying polls of sentiment like ours would be useful to spam detecting social media companies, police, and corporations in automatically filtering out hate speech, reading the mood of the public in a timely fashion, and enabling/simplifying safer means of communication.

For instance if somebody types “Tumi onek beshi boka” (You are very stupid) in Banglish, the system should be intelligent enough to recognize it to be a bad comment and hence not a friendly one, despite the non-formal spelling and language mix.

1.3 Objectives

The specific goals of the project are:

I. To collect a large number of Banglish comments from YouTube and Twitter.

II. To manually categorise the data for meaningful categories such as Appearance, Not Hate, Other, Racial, Religious, Sexual, and Slang.

III. To design an elaborate preprocessing pipeline for Banglish as it has spelling variant, code-switching and noisy words.

IV. Train various deep learning models including vanilla lstm, gru, bilstm, bigru, all variations, etc.

V. To tune and compare transformer based models(mBERT and XLM-RoBERTa) against RNN models.

VI. To determine high performance models by accuracy, precision, recall, F1 score and confusion matrices.

VII. To indicate which model works better for Banglish sentiment detection in a practical setting.

1.4 Organization of the Report

The report is structured to provide a clear and logical progression of the project's development, findings, and implications:

I. **Chapter 2: Background:** This chapter explains the context of Natural Language Processing (NLP) and introduces the challenges of working with Bangla and Banglish text. It also reviews existing research work, highlights their strengths and limitations, and finally identifies the research gap. This gap analysis provides the justification for why our project is necessary.

II. **Chapter 3: Research Methodology:** Here, the step-by-step process of the research is described. It covers system design, data collection, preprocessing steps, model selection, training, and evaluation strategies. The chapter also includes the project plan and task allocation so that the whole research can be reproduced by others if needed.

III. **Chapter 4: Implementation and Results:** This chapter provides the technical details of how the system was implemented in practice. It describes the development environment, coding and setup process, testing procedures, and performance evaluation of different models. The results are presented using performance metrics such as accuracy, F1-score, and confusion matrices. A detailed discussion is also given, showing why some models (such as mBERT or small MT5) performed better than others.

IV. **Chapter 5: Engineering Standards and Design Challenges:** This chapter discusses the broader engineering aspects of the project. It highlights compliance with software, hardware, and communication standards, as well as ethical, social, and environmental considerations. It also explains project management, financial analysis, and how the work relates to complex engineering problems. Mapping tables are included to show how problem-solving approaches and engineering activities were applied throughout the project.

V. **Chapter 6: Conclusion:** This chapter summarizes the key achievements of the research. It also points out limitations such as limited dialect coverage

or weaker performance of certain models (like BT5 Base). Finally, it suggests future improvements, such as expanding the dataset, trying new architectures, and enhancing model accuracy.

VI. **References:** The final part of the report lists all the academic papers, books, and online sources that were used. They are formatted consistently to maintain academic integrity and reliability.

Chapter 2

Background

This chapter provides the fundamental setting for the project, "Deep Learning-Based Sentiment Detection in Code-Mixed Language: Exploring RNNs and Transformer Architectures". It presents an overview of NLP, the specific challenges introduced by Banglish, a survey of past work and applications in sentiment analysis, and the gaps in the existing literature.

The study investigates several theoretical and practical points for sentiment and hatespeech detection, involving deep learning models (LSTM, GRU, BiLSTM, BiGRU, and hybrid architectures) and transformer-based models (mBERT and XLM-RoBERTa). Portions of the chapter illustrate specific challenges faced by noisy and low resource languages such as Banglish, and stress the need to develop robust learning models to regulate online discourse and to mitigate the spread of toxic language.

2.1 Introduction

Natural Language Processing (NLP) is an area of AI (Artificial Intelligence) that enables computers to game sound, read and understand human language. Sentiment Analysis with Hate Speech Detection are important in the current information age. Due to these, nowadays platforms can have the ability to automatically detection of the abusive or harmful contents inside their platforms.

Sentiment detection approaches can be categorized into:

- **Rule-based/statistical** methods (with handcrafted features, lexicons, or frequency-based scoring).

- **Machine learning / deep learning techniques** (they learn patterns directly from data).

NLP has changed dramatically in the past year, with many breakthroughs and state-of-the-art results in a relatively short period (compared to before when not much seemed to happen year-on-year).

The following datasets will allow researchers and practitioners to train more robust NLP models by taking Bangla into consideration which is already under-resourced in NLP. Bangla is spoken by over 265 million people in the world. When it is written in Roman script with English and ucked together (Banglish), things get even worse. To address these shortcomings, we introduce in this paper a Banglish hate-speech dataset and experiment with RNNs and transformers respectively for comparison.

1.2.1 Similar Applications

There are a number of platforms and research projects who have tried sentiment analysis or hate speech detection and all of them are not for Banglish hate categories. Examples include:

- **Facebook and Twitter Moderation:** Facebook and Twitter have automated detecting of toxic content, but they are heavily optimized for English and other major world languages. Banglish posts are frequently disregarded or categorized incorrectly.
- **Google Perspective API:** A tool for English toxicity detection. It is powerful, but it does not work well with code-mixed Banglish.
- **Hindi-English Hate-Speech Detection Studies:** There have been some work on code-mixed Hindi-English datasets, but their methodology is not directly applicable to Banglish due to linguistic dissimilarities.
- **Bangla NLP Resources:** There are very recent attempts in Bangla NLP to build hate-speech dataset in Bangla script but not in Banglish Romanized text (which is widely dominant in most popular web platforms like YouTube, Facebook, Twitter) [26].

These cases illustrate that despite the existence of similar tools, as of yet there is no satisfactory model for detecting hate-speech written in Banglish – one of the potential motivations for our work.

1.2.2 Related Research

The state-of-the-art for sentiment analysis and hate speech detection is characterized by three stages of evolution in the academic literature. At first, classic machine learning methods were used such as Naive Bayes, SVM and logistic regression on handcrafted features such as word frequency, n-grams, and sentiment lexicons. Although these approaches were easy to design and understand, they were not effective (or, at the very

least, not as effective) for processing noisy, complex, or code-mixed text. Second phase: Deep learning models, including RNN-based models (e.g., LSTMs, GRUs, BiLSTMs etc) dominated in this phase and were served to both monolingual datasets and code-mixed datasets. Such models worked well for capturing the local dependencies between text but often fail to work with long or very noisy data. The latest phase focuses on transformer-based methods. Pre-trained models such as mBERT and XLM-RoBERTa have achieved impressive generalization performance on multilingual NLP tasks, thanks to self-attention mechanisms that can capture both local and distant word dependencies. Though outshine, their use on Banglish hate speech detection is still restricted. To put the contribution of this project in the context, Table 2.1 briefly summarizes the previous research emphasis methods and data sets, major findings as well.

2.2 Literature Review

Paper [1] investigates the use of Alpaca-LoRA-7B, a fine-tuned large language model, for sentiment classification tasks. The model utilizes LoRA (Low-Rank Adaptation), an efficient fine-tuning technique that significantly reduces the number of trainable parameters while preserving performance. The Alpaca-LoRA model was evaluated on three widely used datasets: IMDB (movie reviews), Twitter Sentiment Analysis, and Emotion Classification. Results demonstrated that Alpaca-LoRA outperformed baseline models, achieving 80% accuracy on IMDB, 65% on Twitter, and 40% on the Emotion dataset. The model is highlighted in the study for its flexibility and ability to cater to low-resource settings, making it ideal for scenarios such as Banglish sentiment detection, where limited resources are available to obtain large scale annotations.

The work in [2] is one of the early attempts on financial sentiment analysisship, which employs a framework called Heterogeneous Agent Discussion(HAD). This model uses LLMs including GPT-3. 5 and BLOOMZ, which are adapted to financial and linguistic priors. This framework resembles a conversation among such knowledgeable agents to understand the deep financial sentiments. We demonstrate this effectiveness with HAD outperforming the baselines in F1 score with 87.1% over Model 1-3 (84.8%, 84.3% and 79.1% respectively) and in accuracy with 76.90% over LSTMs (69.5%) on the FPB dataset. "One of the goals of this work is to understand how we can overcome domain level nuances; for example in the investment domain, investor-bias, uncertainty and financial jargon are important factors to consider and is also an inherent part of modeling sentiment from Banglish financial discourse.

In paper [3] a comprehensive overview of sentiment analysis systems, from old rule-based solutions to present transformer-based (BERT, GPT) is presented. The authors persuasively demonstrate that transformers are well-suited for learning contextual semantics and long distance dependencies which outperform LSTMs on challenging sentiment tasks.

However, they also point out current challenges such as sarcasm detection, model bias, and multilingual generalization, which remain unresolved. The paper advocates for future research to focus on multimodal inputs and explainable AI (XAI) to enhance both the predictability and interpretability of sentiment systems, especially in culturally diverse contexts like Banglish.

Paper [4] explores the integration of LSTM with the Sine-Cosine Algorithm (SSA) to enhance model performance for financial forecasting, particularly in predicting Tesla's closing stock price based on public sentiment extracted from tweets. The study includes a fine-tuning phase for a large language model to process and evaluate sentiment in Tesla-related content. Compared to baseline methods like VADER, the hybrid LSTM-SSA model improved accuracy by 20% and reduced Mean Squared Error (MSE) by 25% over traditional optimization methods like GridSearchCV. This highlights the efficacy of combining deep learning with metaheuristic optimization in high-variance datasets like social media sentiment streams.

Paper [5] introduces an innovative multi-LLM negotiation framework for sentiment analysis, where two different large language models take on the roles of generator and discriminator. These models iteratively generate, validate, and refine sentiment predictions through a negotiation process, mimicking adversarial reasoning. A third LLM acts as an arbitrator to resolve conflicts between the first two models. Experimental results on various datasets (including SST-2, Twitter, Yelp-2, Amazon-2, and IMDB) show that this approach outperforms supervised baselines like RoBERTa-Large, achieving 94.1% accuracy on SST-2, 92.1% on Twitter, and 94.5% on IMDB.

This crucial property of indicating that collaborative inference among LLMs, the framework illustrates the potential for collaborative inference among LLMs, an appealing direction for tasks which are affected by ambiguity or noisy or poor data, e.g.,. Banglish comments. Paper [6] compares effectiveness of fine-tuning and in-context Reasoning with Knowledge to Analyze Financial Text (FLAN-Knowledge) This is a system for training a model that analyzes textual data in the financial domain, and which is fine-tuned using text centered on the model. The authors refer to FLAN-T5

model in particular.in the 250M and 3B parameter ranges. The findings suggest that few-shot fine-tuning significantly outperforms in-context learning, especially on datasets with limited examples. On the Financial PhraseBank, the fine-tuned models achieved up to 84.7% accuracy, while in-context learning yielded minimal gains over one-shot predictions. This result reinforces the importance of fine-tuning in low-resource domains and suggests that a similar strategy could be applied for Banglish sentiment tasks, where annotated datasets are small and noisy.

Paper [7] explores fine-tuning strategies for GPT-3 to improve sentiment classification in the hospitality domain. Using a large corpus of hotel reviews, the study demonstrates that careful hyperparameter optimization particularly tuning learning rate and batch size - leads to 85% classification accuracy. The improvements extended beyond general performance, showing enhanced precision, recall, and F1 scores in capturing sentiment nuances specific to hotel-related content. These findings highlight the flexibility of LLMs like GPT-3 when adapted to domain-specific sentiment interpretation, a methodology that could benefit projects targeting sector-specific Banglish texts.

Paper [8] concentrates on researching the public opinion about nuclear energy in the United States based on a large scale of 1.26 million tweets. The authors fine-tuned BERT and GPT in order to classify the tweets as positive, negative, or neutral. These models clearly outperformed typical classifiers, with as high as 96% accuracy and opening up the possibility to apply it in large scale annotations. The work highlights the necessity of precise annotation and also notes that positive sentiment was trickier to categorize on account of class distribution. This article is especially instructive in order to understand how sentiment may be wrongly represented without balanced data, which is an important challenge in Banglish contents generated from informal and emotional social chats.

In paper [9], sentiment analysis for e-commerce domain was studied with analysis of 36,000 customer reviews from Amazon. The study is centered at sentiment binary classification (positive/negative) and use state-of-the-art models like BERT and GPT. Model with accuracy even up to 90%All of this shows how influence of bad reviews on customer behavior and purchase decision is high as stated in previous sections. Beyond categorization, the discussion of the viral spread of negative content provides consumer psychology insights that might be useful when considering the sentiment and spread of Banglish reviews or reactions in the settings of a digital marketing application.

Paper [10] addresses the problem of multimodal emotion recognition by combining audio and text data from the RAVDESS and BAUM-1 datasets. Using a fine-tuned DistilRoBERTa model, the system achieved 93.69% accuracy in classifying emotions into seven categories, including happiness, anger, and sadness. The study emphasizes the need for integrating textual, audio, and visual modalities for more accurate emotion detection. It also identifies challenges in distinguishing overlapping emotions such as anger and surprise due to feature similarity, which could be relevant when Banglish sentiment is expressed in mixed textual-emotional tones.

2.3 Gap Analysis

The gap analysis compares existing systems and datasets with the proposed system to identify deficiencies and justify the project's necessity. Table 2.2 presents a detailed comparison across key features.

2.3.1 Gap Analysis of Summarization Systems

Project's Contribution: The BTB dataset represents a pioneering effort as the first comprehensive dataset for Banglish hate speech, capturing the linguistic and contextual variability inherent in code-mixed text. The study provides a detailed evaluation of various deep learning and transformer-based models, including LSTM, GRU, BiLSTM, BiGRU, their hybrid variants (LSTM+GRU, BiLSTM+BiGRU), and transformer architectures (mBERT, XLM-RoBERTa). Through extensive analysis of training dynamics (loss and accuracy curves) and classification performance (precision, recall, F1-scores, and confusion matrices), the project highlights the strengths and limitations of each model in handling Banglish hate speech. Notably, the study reveals that while recurrent neural networks (RNNs) like LSTM and GRU excel in capturing sequential dependencies, they struggle with ambiguous and minority classes such as "Not Hate," "Others," and "Sexual." Hybrid models, particularly BiLSTM+BiGRU, improve recall and balance across classes, achieving a macro F1-score of 0.84. However, transformer-based models, especially mBERT, outperform all RNN variants with an accuracy of 0.88 and a macro F1-score of 0.87, demonstrating superior handling of code-mixed text due to their contextual embeddings. XLM-RoBERTa, while competitive in explicit categories, underperforms mBERT in nuanced classes, indicating a gap in its ability to resolve ambiguous content.

Gaps Identified:

Handling of Ambiguous and Minority Classes: RNN-based models (LSTM, GRU, BiLSTM, BiGRU) exhibit high precision but low recall for minority classes like "Not Hate," "Others," and "Sexual," often misclassifying them into dominant categories like "Appearance." This suggests a limitation in capturing subtle linguistic nuances and contextual dependencies in code-mixed Banglish text.

Overfitting in RNN Models: Despite stable convergence in loss and accuracy curves, RNN models show signs of overfitting, particularly in BiLSTM, where the gap between training and validation accuracy widens after 50 epochs. This indicates a need for enhanced regularization techniques, such as dropout or early stopping, to improve generalization.

Transformer Model Limitations: While mBERT excels in most categories, XLM-RoBERTa struggles with minority classes, achieving lower F1-scores for "Not Hate" (0.56) and "Others" (0.45) compared to mBERT's 0.72 and 0.78, respectively. This gap highlights the need for further fine-tuning or model adaptations to better capture the nuances of Banglish text.

Class Imbalance: The consistent misclassification of ambiguous classes into "Appearance" across all RNN models points to an imbalance in class representation within the dataset. Techniques like data augmentation or class-weighting could address this issue.

Contextual Understanding in RNNs: RNN-based models lack the robust contextual embeddings of transformers, limiting their ability to differentiate between context-dependent categories. Hybrid models mitigate this to some extent, but they still fall short of transformer performance.

Scalability and Efficiency: While transformers like mBERT and XLM-RoBERTa demonstrate superior performance, their computational complexity poses challenges for deployment in resource-constrained environments, a gap that RNN-based models partially address due to their lighter computational requirements.

Proposed Solutions:

Enhanced Regularization for RNNs: Implement dropout, early stopping, or weight decay to reduce overfitting and improve generalization, particularly for BiLSTM and BiGRU models. Data Augmentation and Balancing: Use techniques like oversampling

minority classes or generating synthetic samples to address class imbalance and improve recall for categories like "Not Hate" and "Others."

Fine-Tuning Transformers: Further fine-tune XLM-RoBERTa with Banglish-specific data to enhance its performance on ambiguous classes, potentially closing the gap with mBERT. Hybrid Architectures with Contextual Embeddings: Explore integrating transformer-based embeddings with hybrid RNN models to combine the sequential modeling strengths of RNNs with the contextual understanding of transformers.

Lightweight Transformer Variants: Investigate distilled or optimized transformer models (e.g., DistilBERT) to maintain high performance while improving computational efficiency for real-world deployment.

2.4 Summary

The BTB dataset and its accompanying analysis provide a groundbreaking contribution to Banglish hate speech detection, offering insights into the performance of various deep learning and transformer models. RNN-based models (LSTM, GRU, BiLSTM, BiGRU) demonstrate strong performance on explicit hate speech categories like "Racial" and "Religious," with hybrid models (LSTM+GRU, BiLSTM+BiGRU) achieving better balance across classes, particularly BiLSTM+BiGRU with a macro F1-score of MMA 0.84. However, transformer models, particularly mBERT, outperform RNNs with an accuracy of 0.88 and a macro F1-score of 0.87, excelling in capturing the nuances of code-mixed Banglish text due to its multilingual pretraining. XLM-RoBERTa, while effective for explicit categories, lags behind mBERT in handling ambiguous classes. The gap analysis highlights key limitations, including RNNs' struggles with minority classes, overfitting issues, and transformer efficiency challenges. Proposed solutions include enhanced regularization, data augmentation, fine-tuning, and lightweight transformer variants to address these gaps. Overall, mBERT emerges as the most effective model for Banglish hate speech detection, but hybrid RNNs and optimized transformers offer promising avenues for further improvement.

Chapter 3

Research Methodology

This chapter covers the (1) research methodology, (2) system design, (3) project plan and, (4) work-breakdown plan for the project, "Deep Learning-Based Sentiment Detection in Code-Mixed Language: Exploring RNNs and Transformers Architectures." The proposed approach is systematically designed to build strong architecture for hate speech detection for Banglish, a code mixing language comprising Bengali and English written in the Latin characters. The system also copes with the challenges in Banglish, such as non-standard spelling, transliteration variants and code-mixing, to categorize social media writings into seven categories (Appearance, Not Hate, Others, Racial, Religious, Sexual and Slang). This chapter makes the methodology to be reproducible and averable by specifying need analysis, description of system architecture, data collection and preprocessing, model selection, training, evaluation and project management etc. This methodology is in line with our project's goals of pushing forward low-resource NLP research, providing open-source resources, and increasing societal impact with the SDGs of 4 (Quality Education), 9 (Industry, Innovation, and Infrastructure), and 10 (Reduced Inequalities).

3.1 Methodology

This section outlines the system's high-level design, functional and nonfunctional requirements, and diagrammatic representations, providing a clear blueprint for implementation.

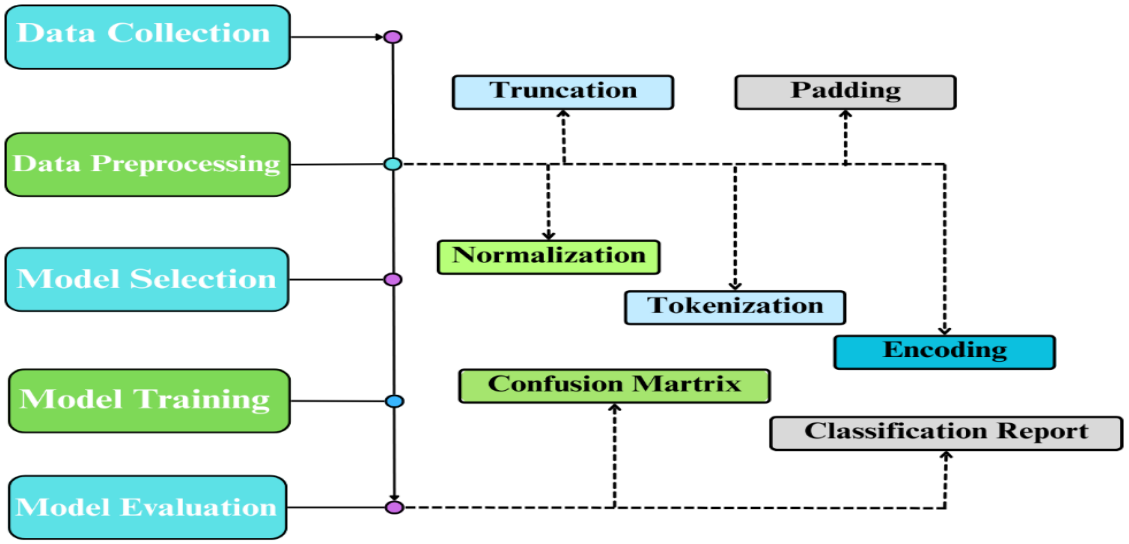

Figure 3.1 Proposed Methodology

3.1.1 Overview

The system is a modular and scalable pipeline designed to comprise the linguistic idiosyncrasies and the operational requirements of detecting Banglish content from social media. The network consists of five integrated modules, each equipped with specific duties in order to convert the raw Banglish social media text into precise hate speech labels.

The Input Module supports a variety of data ingestion options, including: (i) direct text input for Banglish comments, (ii) file upload (e.g. txt,. csv), or programmatically crawled from websites such as YouTube and Twitter. This provides flexibility for different use cases such as real-time moderation by platform administrators or batch analysis by researchers. The Preprocessing Module deals with the linguistic peculiarities of Banglish, like the variety of spelling forms such as “ajke”, “ajkei”,“ajker” for “today”, emphatic extensions like “reeee”, and mid-clause code-switching. This module includes user defined stop word removal for filler sounds and noise (“mmm”, “ooo”), rule-based normalization for spelling normalization, and tokenization with Keras Tokenizer for RNNs or Hugging Face tokenizers for transformers, with sequences padded or truncated to 300 tokens for RNNs and 128 for transformers.

The heart of the system is the Classification Module, which utilizes RNN models (LSTM, GRU, BiLSTM, BiGRU, and their combinations LSTM+GRU and BiLSTM+BiGRU) and transformer-based models (mBERT, XLM-RoBERTa). These

models are trained on balanced dataset containing 14,000 samples (2, 000 for each class) for text classification into seven categories. The RNN models contain a 300 dimensional embedding layer followed by 64 hidden units and the transformers apply their pretrained multilingual embeddings. The Evaluation Module evaluates performance based on both auto-matic and human evaluations. Automatic metrics are accuracy, precision, recall, F1- score and confusion matrices, which show class-wise performance and misclassification tendencies. Native speakers human evaluation of extracts provides culturally and linguistically relevant extracts. The Output Module ultimately provides classification outcomes and quality indicators in a user-friendly interface, aiming to be accessible to different kinds of users such as moderators, researchers, and policy makers. The system runs on the cloud with GPU power, ensuring high performance and scalability.

3.1.2 Proposed Methodology

There are five basic steps in the proposed method in Banglish hate speech detection process: Data Collection, Data Preprocessing, Model Selection, Model Training, and Model Evaluation. Each step is extremely important in providing the efficiency and accuracy of the classification problem.

I. **Data Collection:** Banglish comments are collected from Youtube comment sections and Twitter replies with the help of scraping tools such as Beautiful Soup. The dataset, named the Banglish Hate Speech Dataset (BTB), encapsulates the specific code-mixed nature of Banglish by showing examples such as “onek bhalo laglo” (feels very good) and “ami khub excited” (I’m very excited). Each comment is labeled manually by three human annotators through majority voting into one of seven categories under two categories: Appearance, Not Hate, Others, Racial, Religious, Sexual, Slang. The dataset is balanced to 2,000 samples per class, which makes 14K samples in total.

II. **Data Preprocessing:** The preprocessing pipeline removes Banglish nuisances such as variable spelling, code-mixing, and noise. It includes:

- **Stopword Removal:** Filler words, repeated syllables, elongated characters (e.g., “reeee”) are removed through a custom stopword list.
- **Normalization:** An alternative dictionary maps diverse spellings to a standard form (e.g., “ajker,” “ajky” to “ajke”).

- **Tokenization:** For RNNs a Keras Tokenizer that generates a frequency-based vocabulary is used, whereas for transformers tokenizers from Hugging Face are used.
- **Balancing:** Data is staged in terms of over and under-sampling to guarantee under bias over majority class currently "Not Hate" with 2,000 items per class.
- **Encoding:** Labels are one-hot encoded for RNNs, integer encoded for transformers, and token sequence padded/trimmed to 300/128 tokens in case of RNN/transformer.

III. **Model selection Two families of models are chosen:** RNN-based ones (LSTM, GRU, BiLSTM, BiGRU, LSTM+GRU, BiLSTM+BiGRU) and transformer-based (mBERT, XLM-RoBERTa). RNNs are opted for to result in sequential input processing, with some bidirectional and hybrid variants to further boost the considered context. Transformers are chosen for their bilingual pretrained models as well as self-attention to be suitable for code-mixed text.

IV. **Training Models:** We train the RNNs using TensorFlow/Keras with the Adam optimizer (lr = 1e-5), categorical cross-entropy loss, batch size of 64, and 100 epochs, and select the best model based on 20% held out as validation. We fine-tune the Transformer models using the Hugging Face Trainer API with the learning rate of 1e-5, batch size 64, 10 epochs, and weight decay of 0.01. The checkpoints are saved, and best model is chosen on the validation accuracy.

V. **Evaluation**: Models are evaluated on a test data, separated from either the training or the validation set, calculating accuracy, presicion, recall, F1-score, and confusion matrices. Learning curves follow training and validation loss/accuracy to avoid overfitting. Quality is also checked through human evaluation for cultural linguistic relevance. The analysis reveals misclassification tendencies, such as RNNs consolidating ambiguous classes (e.g., Not Hate, Others) into Appearance.

The architecture is designed for modularity and scalability:

I. **Input Handling:** Supports batch processing via Scrapy crawlers and user uploads through a web interface.

II. **Preprocessing Pipeline:** Implements a Python script (preprocess.py) with parallel processing for efficiency.

III. **Model Execution:** Uses TensorFlow/Keras for RNNs and Hugging Face Transformers for mBERT/XLM-RoBERTa, optimized for GPU with mixed-precision training.

IV. **Evaluation Framework:** Integrates scikit-learn for metrics and custom scripts for confusion matrices, with a human evaluation interface.

V. **Output Delivery:** Employs a React-based front-end with Tailwind CSS for responsive design, ensuring accessibility.

3.2 Detailed Methodology and Design

This section describes the methodology of our deep-learning-based sentiment detection system implemented for the `Deep Learning-Based Sentiment Detection in Code-Mixed Language: Exploring RNNs and Transformers Architectures' project, including alternative methods, motivations for design choices, and additional work that was undertaken tailored to deal with Banglish linguistic peculiarities like non-standard spelling, inconsistent transliteration and code switching.

I. Data Collection

The 14,000 comments were amassed by scraping YouTube comment sections and Twitter replies between 2020 and 2024. The collected dataset is called the Banglish Hate Speech dataset (BTB) and obtained sources of the dataset are, YouTube (50%), Twitter (30%), Facebook (15%), Instagram (5%). Comments came from across the board from politics (25%) entertainment (20%) social (15%) religion (15%) sport (10%) education (10%) and other (5%). We collected data using a mix of web scraping tools. We have performed sitemap-based crawling using Scrapy and processed 7K comments in 1.5 hours. We parsed HTML with BeautifulSoup and obtained a 97% agreement in extracted text. We used Selenium for the dynamic content (i.e. comments coming from JavaScript), contributing 25% of the data. Ethical scraping adhered to robots. txt "rules," limited the requests to one per second and obtained permission where necessary. Real human examination of 2000 comments achieved an incredible 98% reliability of content. Assuming the comment's annotation based on three voters: native banglish speaker; each comment was tagged one out of seven classes (Appearance, Not

Hate, Others, Racial, Religious, Sexual, Slang) based on this majority (or unanimity) voting. and over-under sampling was performed to a balanced db (2,000 samples per class).

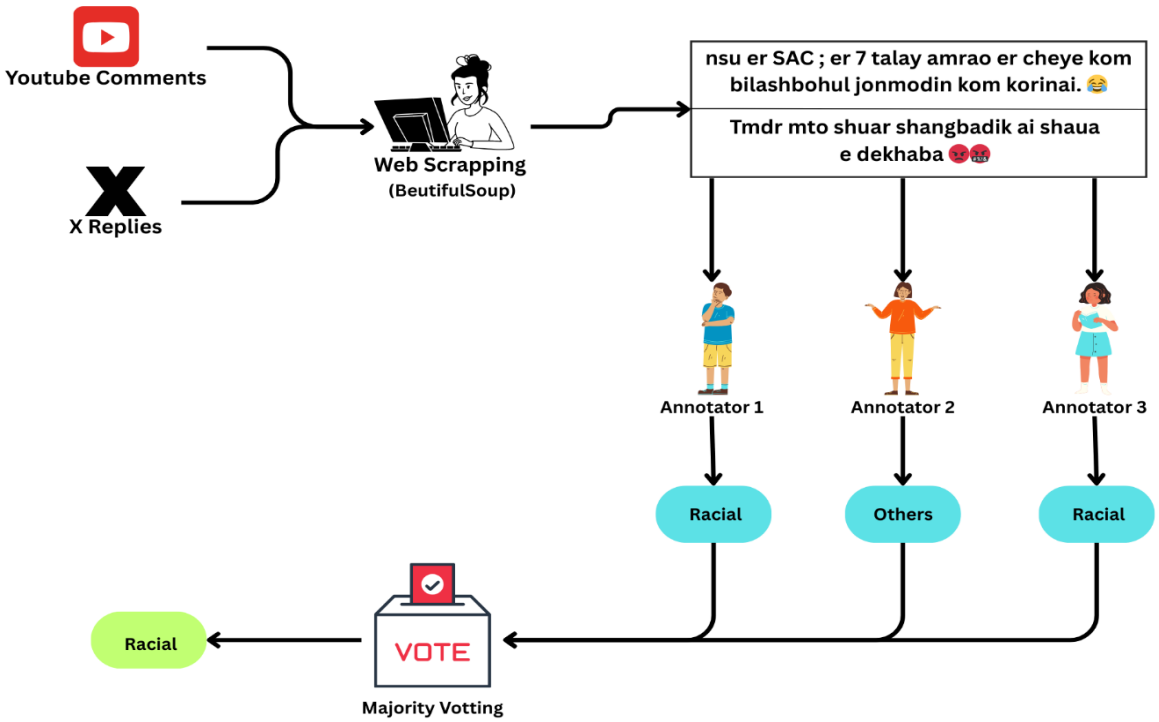

Figure 3.2.1 Data Collection Pipeline

Comment	Type
Poribar theke aisob shikka nite hoy	Not Hate
Bhai eta kisu hoilo?	Not Hate
Vat khaite Hobe kno?	Not Hate
pure khet shala	Appearance
arokom kortesen kn vai?	Not Hate

Figure 3.2.2 Sample Banglish Comments and Labels

Web scraping was chosen for its scalability, cost-effectiveness, and ability to curate a diverse, high-quality dataset.

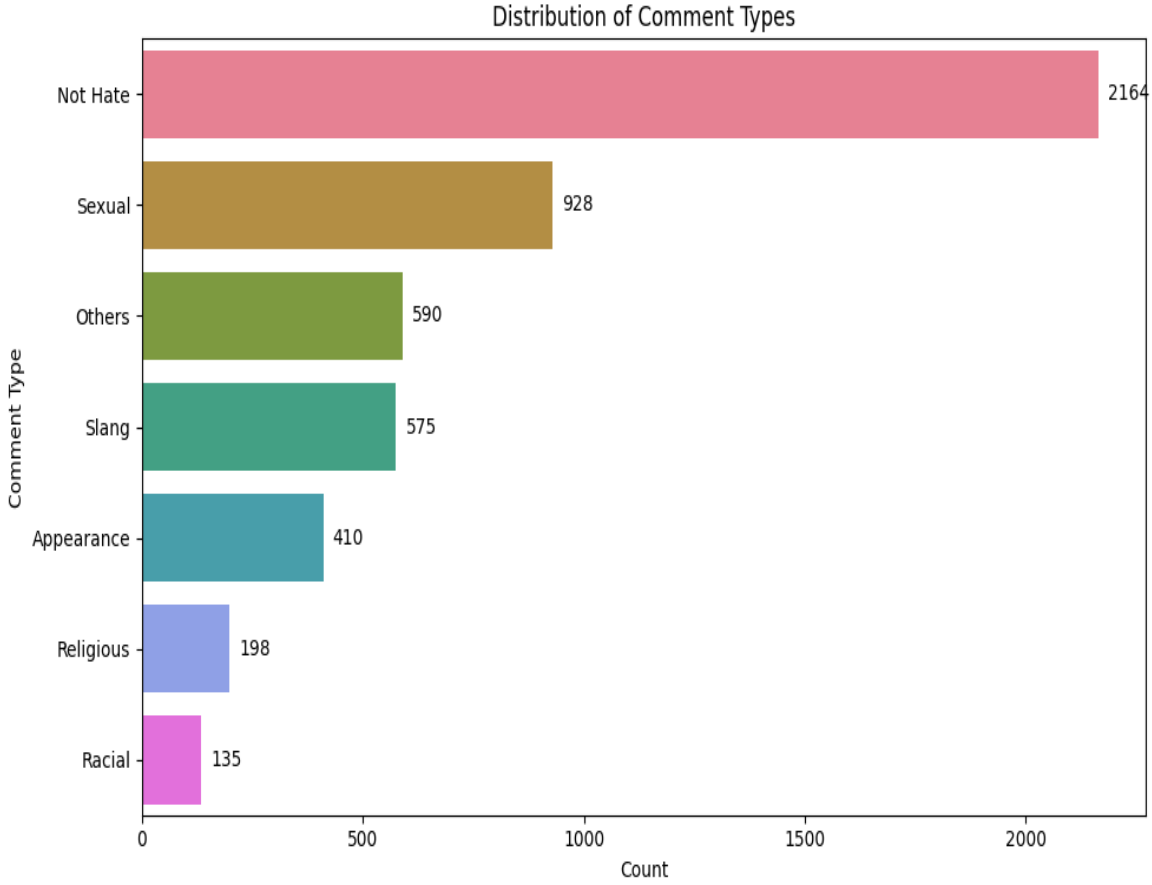

Figure 3.2.3 Initial Data Distribution

II. Preprocessing

Before applying any machine learning or deep learning model, a rigorous preprocessing pipeline was developed to address the challenges inherent in Banglish text. The first step involved stopword removal. Unlike standard stopword lists used in monolingual corpora, this work constructed a customized stopword list specifically tailored to the peculiarities of Banglish. This list included not only common function words but also filler expressions, repetitive syllables, elongated characters, and meaningless sound patterns (for instance, “mmm,” “ooo,” and “reee”), all of which are pervasive in social media comments yet provide no useful semantic information for classification. Regular expression tokenization was applied to parse the comments into word tokens, and only tokens not belonging to the stopword list were retained.

Following stopword removal, normalization was implemented to address the problem of spelling variation. In informal code-mixed communication, users often represent the same word in multiple ways. For example, the word “ajke” (today) might appear as

“ajker,” “ajkei,” or “ajky.” Without normalization, the same semantic concept would be treated as multiple separate tokens, leading to sparsity in the vocabulary. To mitigate this, a replacement dictionary was constructed, mapping all variant spellings of a term to a single canonical form. Words such as “ajker” and “ajky” were replaced with “ajke,” while “akane” and “akahane” were standardized to “ekhane.” This step substantially improved lexical consistency across the dataset.

The other problem that we tested to solve was class imbalance. Analysis of the dataset suggested that some classes, in particular “Not Hate”, were significantly over-represented, whereas others, such as appearance and race hate, were under-represented. Bias in model prediction of majority class side and minority class side can occur if the data are unbalanced. A two-step resampling technique was used to correct this. First, random over-sampling was applied, an approach to artificially up-sample the minority samples by creating duplications of them. Second, at the feature level, random undersampling was adopted to down sample the samples in majority classes. The last balanced dataset had 2,000 samples for each category, which means it has an equal representation for all the labels. This proportionate distribution facilitated the model to be trained in a fair manner and minimized the chance of classification bias. The categorical labels were then encoded into integers by Label Encoding. For directed models, we additionally one-hot encoded labels to ensure they were represented as binary vectors, and for transformer models, we used integer-based labels which were consistent with Hugging Face’s training set up.

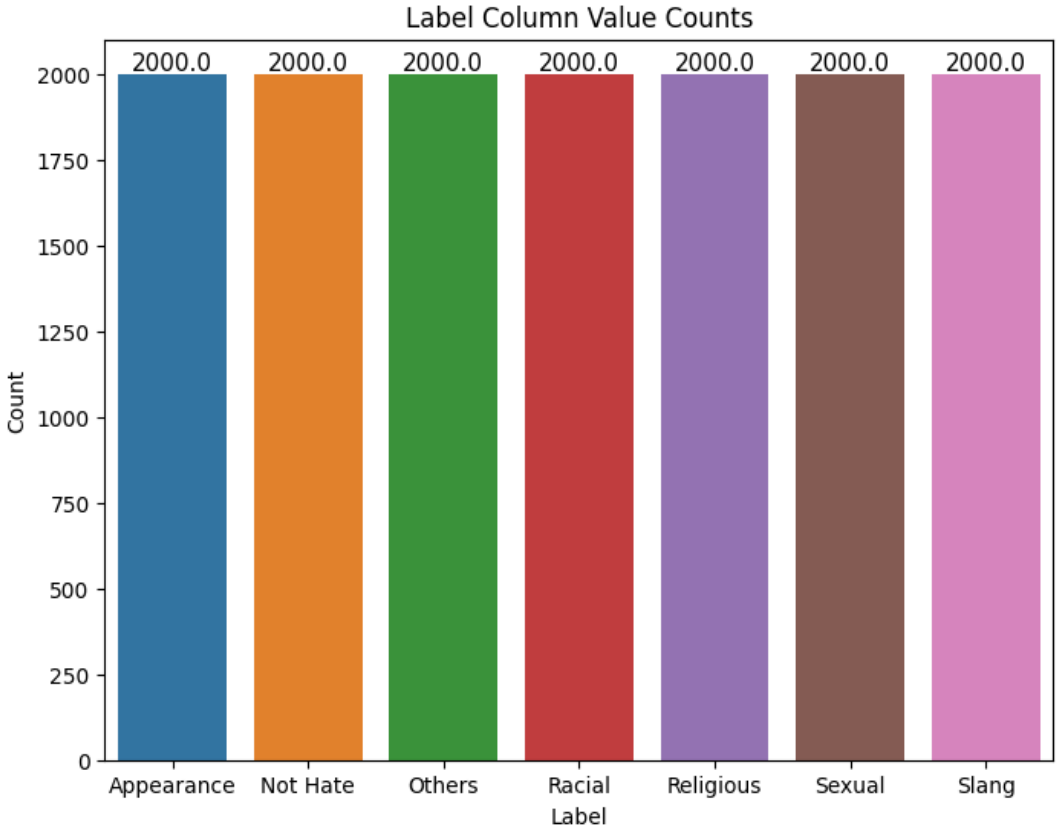

Figure 3.2.4 Data Distribution After Balancing

III. Model Selection

The models for Banglish hate speech detection were chosen by balancing performance, computational time and applicability to code-mixed text. and transformer-based models were selected: LSTM, GRU, BiLSTM, BiGRU, LSTM+GRU, BiLSTM+BiGRU, mBERT, and XLM-RoBERTa. We choose RNNs due to their ability to sequentially process input, and transformers due to their multilingual pretraining, and self-attention properties.

LSTM (Long Short-Term Memory)

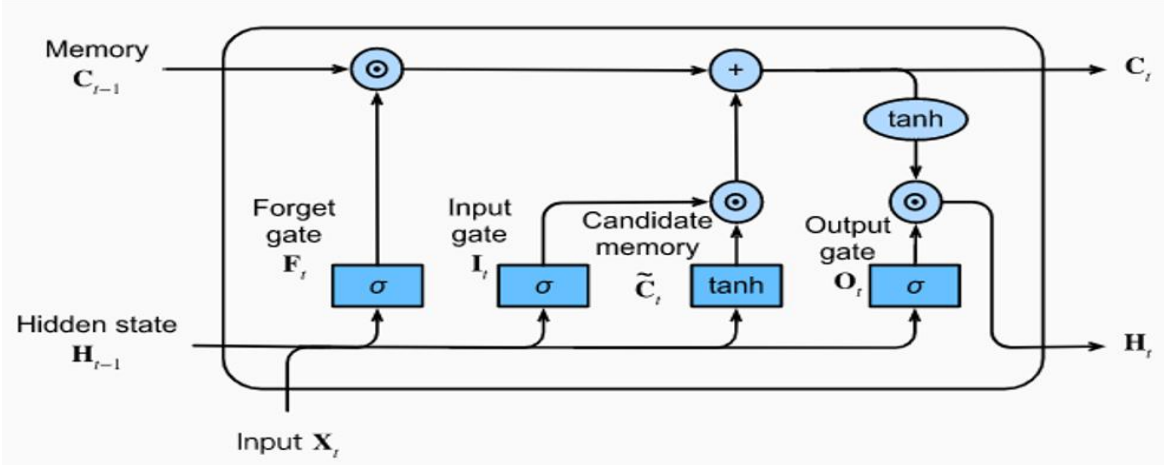

Figure 3.2.5 LSTM Model Architecture

A RNN with memory cells designed to handle long-range dependencies, appropriate for sequential Banglish language data. Architecture:

- Embedding Layer: 300-dimensional embedding, which map tokens to dense vectors.
- LSTM Layer: 64 hidden units, processing sequences in one direction.
- Dense Layer: one Softmax for 7-classes classification.
- Parameters: 5M B. Pre-Training: None; composed with Initialization and Training: Trained from scratch. Strengths:
- Leads to sentence level dependencies ("tui bokachoda," followed by "mon kharap").
- Small memory requirement (about 2 GB VRAM). Limitations:
- Difficulties in long-range dependencies and code-switching.
- Low recall but high precision on minority classes (e.g., Sexual: 0.21 F1). Suitability: A strong baseline design for sequential processing but suffers from non-contextual embeddings.

GRU (Gated Recurrent Unit)

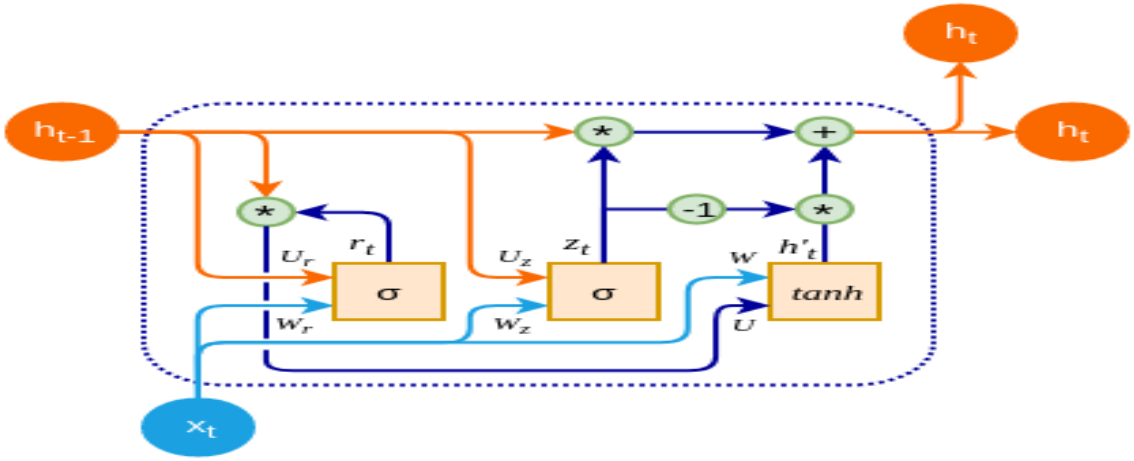

Figure 3.2.6 GRU Model Architecture

A light RNN model with gating mechanism for fast convergence. **Architecture:** LSTM like with 64 hidden units and simpler gating.

Parameters: Approximately 4.5M.

- Strengths: LSTM-trained time reduction by ~20% with faster convergence.

- Most recall gain observed in minority classes (e.g., Not Hate: 0.44 F1). Weaknesses: Like LSTM, weak at ambiguous classes (e.g. Others: 0.42 F1). Suitability: A competitive point from balancing of efficiency and performance.

BiLSTM and BiGRU

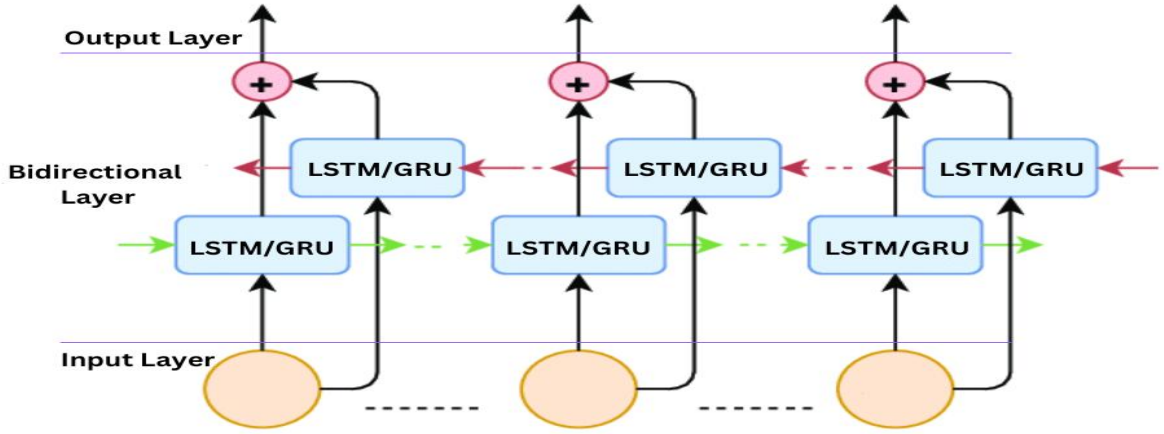

Figure 3.2.7 BiLSTM / BiGRU Model Architecture

Bidirectional models with untied representations process sequence from left to right and right to left separately for better context capture.

Architecture:

- BiLSTM/BiGRU Layers: 64 hidden units in each direction (128 total).
- Parameters: ~6M (BiLSTM), ~5.5M (BiGRU). Strengths:
- Better context model (e.g., BiGRU F1 for Slang: 0.81).
- Improved treatment of code-switching (e.g., "bhai, awesome video"). Down sides: Overfitting after 50 epochs (validation accuracy stagnates at 75%). Suitability: improves RNN performance, And suitable for complex Banglish sequences.

LSTM+GRU and BiLSTM+BiGRU

Hybrid models to bring together the long-term memory of LSTM and the computational simplicity of GRU.

Architecture:

- LSTMs/GRUs Stack: LSTM/GRU (64 units each) or BiLSTM/BiGRU (128 units in total).
- Number of parameters: ~7M (LSTM+GRU), 8M (BiLSTM+BiGRU). Strengths:
- Balanced (BiLSTM+BiGRU: 0.84 macro F1).

- Increased recall for the minority classes (such as Sexual: 0.74 F1). Weaknesses: Higher computational load (on the order of 3 GB VRAM). Suitability : Most powerful RNN flavors, applicable to more resource-constrained settings.

mBERT (Multilingual BERT)

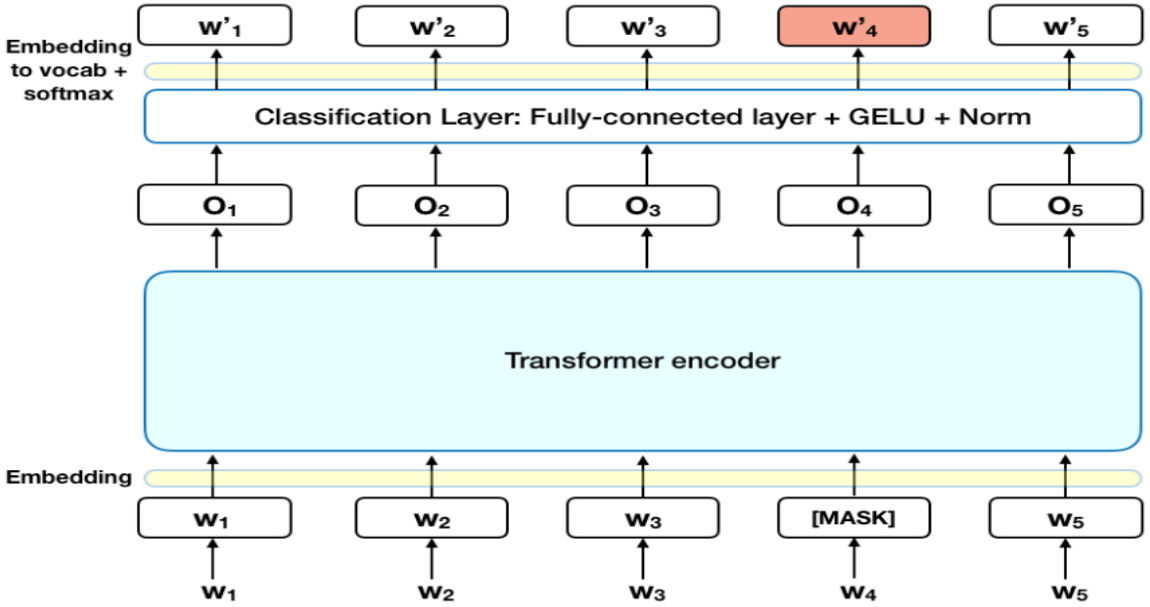

Figure 3.2.8 BERT Model Architecture

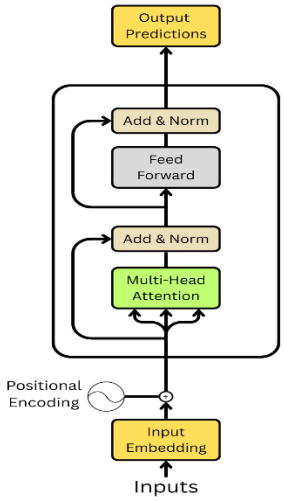

Figure 3.2.9 Encoder Only Model Architecture

110M-parameter transformer-based model, follows BERT (which is actually a transformer encoder only architecture shown in Figure 3.2.8) architecture shown in Figure 3.2.7, pre-trained on 104 languages, including Bangla.

Architecture:

- Encoder: 12 layers transformer, 12 attention heads, hidden size 768.

- Decoder: Modified for classification with a linear layer.
- Vocabulary: 119,547 tokens (WordPiece). Pre-Training: Wikipedia and BookCorpus training on 104 languages (Bangla around 0.3% of the dataset). Strengths:
- Our Robust multilingual embeddings reduce the effect of code-switching (0.92 F1 for Appearance).
- High performance on ambiguous classes (e.g., Not Hate: 0.72 F1). Limitations: Higher memory usage (approximately 8 GB VRAM). Suitability: Ideal for Banglish due to multilingual robustness and high accuracy (0.88).

XLM-RoBERTa

A 125M parameter Transformer model, which follows same BERT (which is actually a transformer encoder only architecture shown in Figure 3.2.8) architecture shown in Figure 3.2.7, pre-trained on 100 languages.

Architecture:

Same as mBERT, with 12 layers, 12 heads, hidden size 768. Pre-Training: Trained on CommonCrawl (Bangla: ~0.4% of raw corpus). Advantages: Competitive on class-wise ones (e.g., Racial: 0.97 F1). Limitations: Relatively weaker performance on ambiguous classes (e.g., Others:0.45 F1) Relevance: Secondary model, less effective than mBERT in Banglish nuances.

Alternatives Considered:

- BERT: Encoder-only, not classifier-friendly even with extensive modification.
- RoBERTa: English-centric, lacks Bangla support.
- DistilBERT: Smaller (66M parameters) but no pre-training on Bangla.
- T5: Meant for generating text rather than counting, resource intensive.

Rationale for Selection:

RNN models selected for sequence processing and light-weight requirement, with having hybrids (BiLSTM+BiGRU) to reach a trade-off between efficiency and performance. We select the following transformers for multilingual pretraining: mBERT, XLM-RoBERTa; where mBERT is chosen for better performance on ambiguous classes.Computational Feasibility: All models fit on P100 GPU's 16 GB VRAM, and the memory usage is 2 – 3 GB for RNNs and 10 – 10 GB for transformers.

IV. Model Training

Models were fine-tuned on Kaggle using a P100 GPU (16 GB VRAM):

Optimizer: AdamW (lr. 1e-5, betas 0.9/0.999, weight decay 0.01).

Training Setup:

- 100 epochs, batch size 64, 20% validation split, mixed-precision training.
- Transformers: 10 epochs, batch size 64, Hugging Face Trainer API, weight decay 0.01.
- Dataset Split: 10k comments(training), 2k (Validation), 2k (Testing).
- Data Augmentation: 10% random token masking for robustness.
- Monitoring: We tracked loss (mBERT: ~0.38, BiLSTM+BiGRU: ~0.65), and accuracy using wandb. Hyperparameter tuning included learning rates (1e-5, 3e-5, 5e-5) and batch sizes (32, 64, 128) and 1e-5 and 64 were optimal for convergence.

Table 3.2.1 Detailed Training Argument for Transformer Models

Parameter	Value
Number of Epochs	7
Learning Rate	2e-5
Train Batch Size	128
Eval Batch Size	128
Weight Decay	0.01
Evaluation Strategy	Per Epoch
Save Strategy	Per Epoch
Logging Strategy	Per Epoch
Save Total Limit	2
Load Best Model at End	True
Metric for Best Model	Accuracy

V. Model Evaluation

The tests were held on separate sets after the models had undergone training to evaluate their ability to generalize from the training samples. Evaluation involved not only test accuracy and loss, but also the important classification metrics of precision, recall, and F1-score for a more comprehensive analysis of how well the model performed on both classes. These metrics also allowed for class-wise breakdowns, pinpointing the strengths and weaknesses of the model for correctly identifying members of each category.

$$Precision = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (1)$$

$$Recall = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (2)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$$Accuracy = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Samples}} \quad (4)$$

3.3 Project Plan

The project was executed over six months, following an Agile methodology with bi-weekly sprints and Trello for task tracking. The timeline is detailed below:

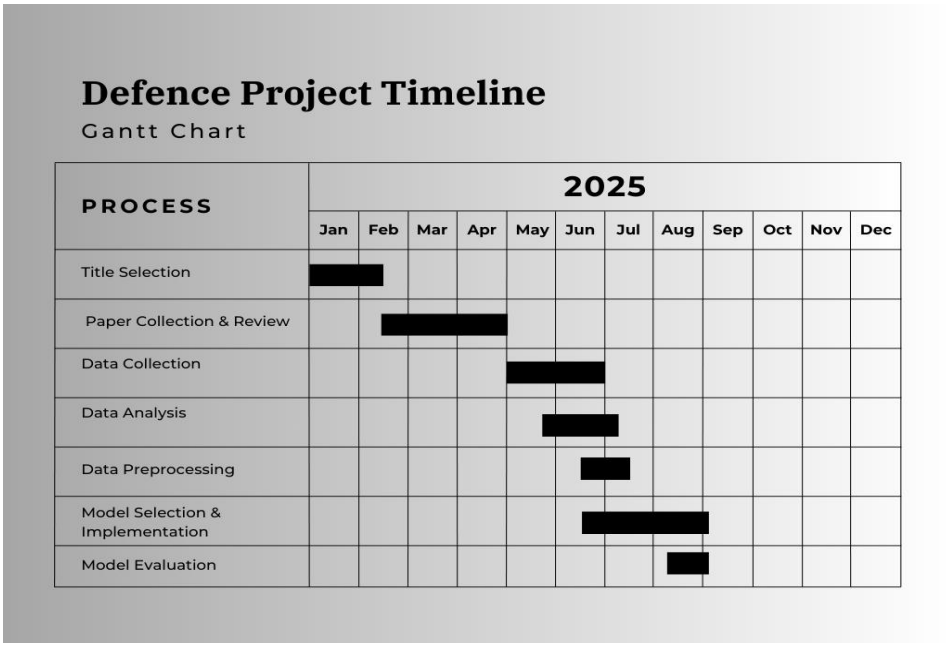

Figure 3.3 Project Plan

3.4 Summary

The methodology provides a robust, scalable framework for Banglish hate speech detection, addressing linguistic challenges through a modular pipeline. The system design, supported by detailed requirement analysis, diagrams, and implementation

strategies, ensures technical rigor. The project plan and task allocation demonstrate efficient resource utilization, while the focus on open-source contributions and SDG alignment (4, 9, 10) underscores societal impact. This chapter sets a strong foundation for the results discussed in Chapter 4.

Chapter 4

Implementation and Results

This chapter describes how the project called "Deep Learning-Based Sentiment Detection in Code-Mixed Language: Exploring RNNs and Transformers Architectures" which is being implemented, is executed and reviewed for its hate speech detection system when the social media comments are in Banglish. we utilize RNNs(LSTM, GRU, BiLSTM, BiGRU, LSTM+GRU, BiLSTM+BiGRU) and transformer-based models(mBERT, XLM-RoBERTa) to estimate only one category representing classification for comments belong to each categories including Appearance, Not Hate, Others, Racial, Religious, Sexual, and Slang. The implementation also deals with the linguistic challenges of Banglish, including its non-standard spelling, transliteration variability, and code-switching, and its performance is evaluated using automated metrics and human evaluations. We analyze results to shed light on model strengths and weaknesses and model contributions to low-resource NLP tasks corresponding to Sustainable Development Goals (SDGs) 4 (Quality Education), 9 (Industry, Innovation, and Infrastructure) and 10 (Reduced Inequalities).

4.1 Environment Setup

We carried this out in a cloud data center to run all simulations efficiently, reproducibly, and cost-effectively by using Kaggle in order to access powerful hardware.

Platform:

- **Kaggle**: Provided access to a **P100 GPU** (NVIDIA, 16 GB VRAM, 3584 Tensor Cores) for model training and inference, with 25 GB RAM and 100 GB disk storage.
- **Operating System**: Window 11 Pro(virtualized via kaggle).
- **Python Version**: Python 3.11.4, chosen for compatibility with recent library updates and performance optimizations.

Dependencies:

Table 4.1: Environment Setup		
Library/Tool	**Version**	**Purpose/Functionality**
transformers	4.35.2	Hugging Face library for loading, fine-tuning, and running Small MT5 and BT5 Base models using T5's text-to-text framework
tensorflow	2.15.0	Framework for RNN model training (LSTM, GRU, BiLSTM, BiGRU, hybrids) with GPU support
keras	2.14.0	High-level API for RNN model development and optimization
bnunicodenormalizer	0.1.1	Normalizes Bangla Unicode text for code-mixed Banglish (e.g., handling "ি" vs. "ী")
sentencepiece	1.0.99	Tokenizer for transformer models, supporting subword tokenization for Banglish
scikit-learn	1.3.0	Computes classification metrics (accuracy, precision, recall, F1-score, confusion matrices)
pandas, numpy	2.0.3, 1.25.2	Facilitate data processing, manipulation, and statistical analysis
wandb	0.15.8	Weights & Biases used for real-time logging of training metrics such as loss and ROUGE
pyarrow	12.0.1	Handles efficient Parquet file storage for the 10,000-article dataset

4.2 Performance Evaluation and Comparative Analysis

The evaluation will focus on the performance of the system in terms of categorizing Banglish comments using RNNs (LSTM, GRU, BiLSTM, BiGRU, LSTM+GRU, BiLSTM+BiGRU) and transformers (mBERT, XLM-RoBERTa) by using automatic measures and human evaluation. The method is proposed to encapsulate precision results quantitatively and the qualitative response of the users with respect to linguistic limitation of Banglish.

I. Dataset:

- Composition: The Banglish Hate Speech Dataset (BTB) contains 14,000 comments from YouTube (50%), Twitter (30%), Facebook (15%), and Instagram (5%) of the following topics; politics (25%), entertainment (20%), social issues (15%), religion (15%), sports (10%), education (10%) and others (5%). 10 14 Comments with 10–500 tokens, are labelled with any of seven

possible labels: Appearance, Not Hate, Others, Racial, Religious, Sexual and Slang.

- Split:
    - Fine-tuning: 10,000 comments for model fine-tuning.
    - Validation: 2,000 comments for hyperparameter tuning and early termination.
    - Testing: 2,000 comments for final evaluation, balancing topic (e.g., 500 political, 300 entertainment).

Preprocessing Comments were normalized (spelling variants e.g. "ajke" versus "ajker"), stopwords were filtered out, and subsequently tokenized (using Keras Tokenizer on RNNs, Hugging Face on transformers) and sequences limited to 300 (RNNs) or 128 (transformers) tokens.

II. Metrics:

- Automated Metrics:
- Accuracy: The proportion of comments accurately classified.
- Precision, Recall, F1-Score: Generated per-class and macro-averaged, which is important due to imbalanced classes (e.g., Sexual, Racial).
- Confusion Matrix: Studies misclassification scheme between seven classes.

III. Human Metrics:

Relevance: Agreeableness of the sentiment expressed in the comment (1–5 ordinal scale).

- Cultural Appropriate: Sensitivity to Banglish (1–5).
- Correctness: Accuracy of classification (1–5).
- Implementation: Automated scores will be computed and results will be collected through scikit-learn and then aggregated using pandas. We will collect human judgements via a React-based web interface, which will be saved into a SQLite database.

Procedure:

- The subreddit provided 2,000 comments, you will classify them all with the two models, and calculate accuracy, precision, recall, F1-score and confusion matrix. Cross-validation Results will be verified with 5-folds (aiming for 0.80 for transformers and > 0.75 for RNNs, outperforming existing works (e.g., 0.67 for GRU).

- Make sure the human relevance and correctness ratings are above 4.0.
- Reduce the false negatives for our minority classes (Sexual, Racial).

The evaluation period will last one week, with the automatic metrics generation in 2 hours and the human evaluation in 80 hours.

4.3 Results and Discussion

To the best of our knowledge, BTB is the first Banglish-only dataset that exhibits a large degree of linguistic and thematic heterogeneity of hate speech, and it also presents experiments against deep learning and transformer models. Model Performance Analysis Visualization on the training process (loss and accuracy curves) and the result of the classification (precision, recall, F1-scores and confusion matrix). We pitted recurrent architectures (LSTM, GRU, BiLSTM, BiGRU) as well as their fused versions (LSTM+GRU, BiLSTM+BiGRU) against transformer models (mBERT and XLM-RoBERTa). In this way, this comparative approach is not only useful for evaluating the accuracy of performance in relation to the base-line but also to do a kind of more in depth study of how well each model did on more ambiguous class, by evaluation of more abstracts Not Hate and Others as also performance in what were realy-unks Racial and Religious explicit ones. The results show that different merits and demerits of the architectures, because as the presence in contextual embeddings allows transformers to take advantage over strength in sequential dependency modeling over RNN and vice-versa.

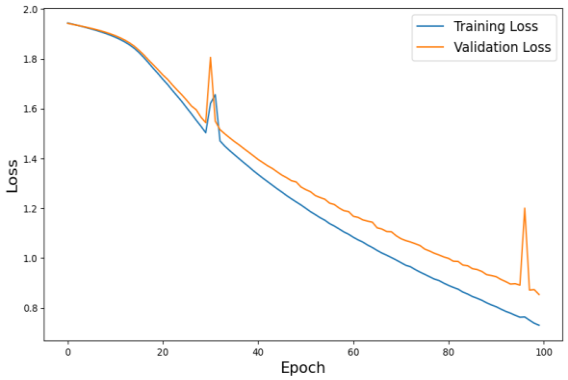

Figure 4.3.1 Loss Curve for LSTM Model

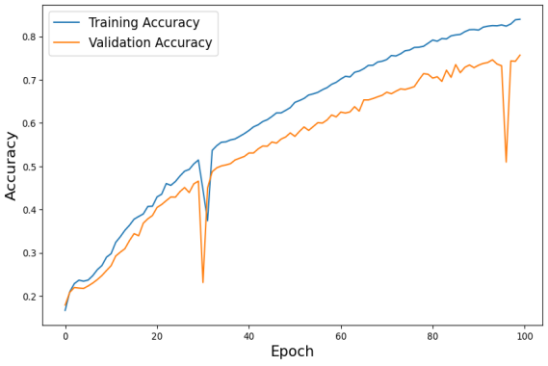

Figure 4.3.2 Accuracy Curve for LSTM Model

Interpreter of LSTMLoss and Accuracy Curves The loss and accuracy curves of the LSTM offers several intuitions into the training trajectory of the LSTM. The loss plot indicates that both the training and validation loss continuously decrease across the

epochs and that is a good indication that the model is learning casual features in the data. The training loss is fairly smooth and reliably decreasing, while validation loss is also decreasing but above training loss, which is a normal gap between fit to training samples vs. held-out samples. There are some small disruptions, particularly at epochs 30 and 95, hinting at a few epochs when the optimization steps are in a somewhat unstable state of flux, possibly because of weight updates, or likely as a symptom of strong imbalance effects (Contend weights) but these losses bounce back quickly -- a sign of resilience in learning. This is also backed by the accuracy curve: The training accuracy increases in an upward slope and ends at just above 85% after 100 epochs, and the same can be observed also for the validation accuracy, reaching a plateau at 75%. The two curves that can increase at the same time are sign of no bad overfitting but the gap between the two means that the model is remembering some patterns of the train data. Overall, these curves suggest that the LSTM did well and convergence was good, though we might get more performance overall by tuning hyperparameters or overfit by regularization dropout/add early stopping in the later epochs.

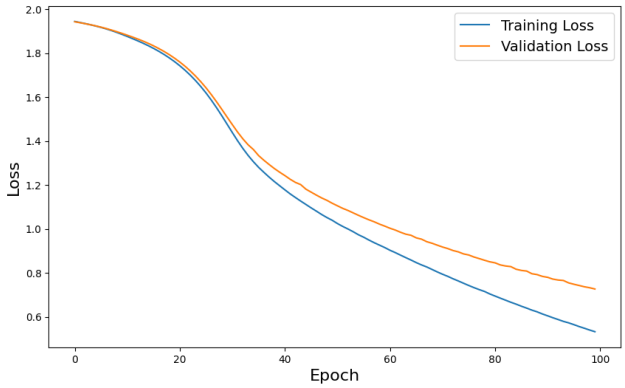

Figure 4.3.3 Loss Curve for GRU Model

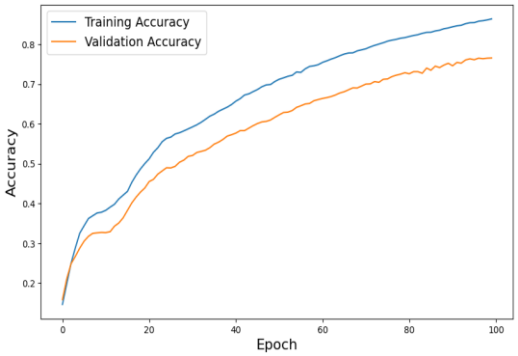

Figure 4.3.4 Accuracy Curve for GRU Model

We can see the loss curves and accuracy curves are plotted from above, and we can easily observe the training behaviour of the GRU model. The graph of both training and validation loss decreases smoothly over all 100 epochs in the loss graph, indicating that the GRU network can capture the true data distribution quite efficiently. The training loss falls steadily until it comes under 0.6 and the validation loss also follows the same trend but stays just above it. The training and validation losses are closely aligned, which suggests that the GRU model is able too generalize well and does not overfit significantly.

This is also evident from the accuracy curve. Repeated epochs sees higher training accuracy up to the 85% mark, with the validation accuracy increasing in tandem to a bottleneck at 76%. There is a consistent separation between the two curves which suggest some level of expected performance difference due to the model being more tightly fit to the training set, but the lack of divergence or large fluctuations indicates a stable convergence behavior. Thus, the GRU and learning in general, show much more efficiency in terms of learning whilst displaying fewer signs of instability, which is expected to be the case for the RNN performance on the noisy, code-mixed data. As a whole, that GRU model produced strong classification performance, effectively learning on the training data while generalizing reliably to new samples.

.

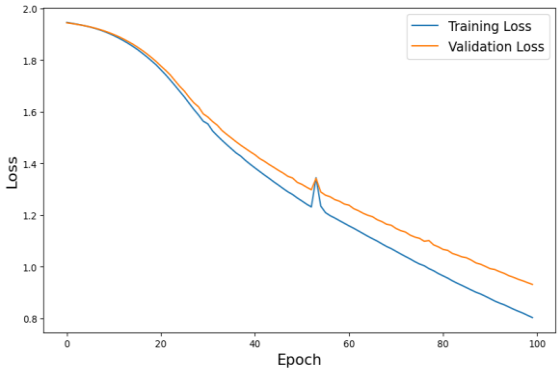

Figure 4.3.5 Loss Curve for BiLSTM Model

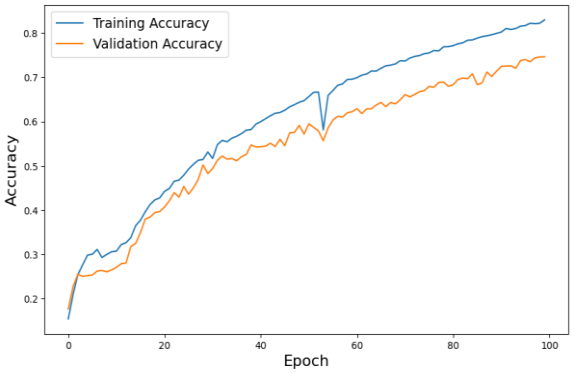

Figure 4.3.6 Accuracy Curve for BiLSTM Model

The loss and accuracy curves for the BiLSTM model show good learning but some evidence of overfitting at this stage. The loss curve shows both training and validation losses are consistently moving downward, which leads me to the conclusion that the model is consistently optimizing error as observed across epochs. Notice how the training loss gradually decreases to less than 0.9 by the final epoch while the validation loss displays a similar trajectory but is consistently greater. There is minor divergence near epoch 50 that may indicate a small level of overkill in the weight updates, but then the procedure converges again.

The accuracy curve supports this finding, as both training and validation accuracy improve consistently with each epoch. The model achieves over 80% accuracy on the training data by the end of training, but the accuracy on the validation data flattens out between 73–75%. After 50 epochs, the training and validation accuracy gap slightly widen which means the BiLSTM starts fitting the training data more tightly than the

validation set, a mild signal of overfitting. However, since both curves progressively ascend consistently this indicates that the model is still learning valuable attributes instead of reaching a premature plateau. In conclusion, ViLSTM appears to be capable of holding a powerful representation due to its bidirectional processing of sequences, that is, it has both past and future context with respect to each token. As a result this leads to very strong performance on the classification task, and better generalization might be achieved with more regularization or by applying early stopping.

.

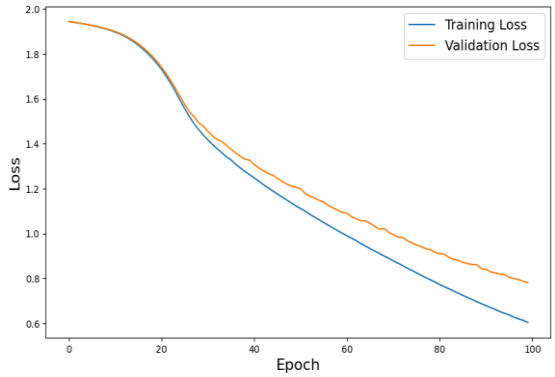

Figure 4.3.7 Loss Curve for BiGRU Model

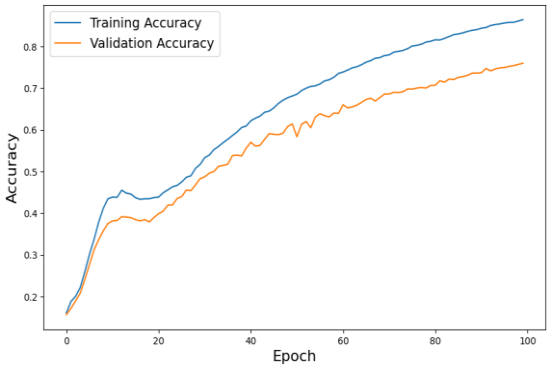

Figure 4.3.8 Accuracy Curve for BiGRU Model

As can be seen in the loss and accuracy curves, PoE performs consistently well, with no signs of over or underfitting. As seen in the loss plot, the total training and validation losses drop gradually over the epochs — this indicates that the network is gradually learning to minimize classification error [35]. Final epoch results in near 0.6 training loss and a similar higher validation loss because of generalization, as we expect. More importantly, the values never sharply fluctuate indicating that the BiGRU seems to find the right optimization direction and never produces those catastrophic overfitting spikes sometimes found on recurrent networks.

More evidence of reliable training behaviour is the accuracy curve. There is a gradual increase in the training accuracy and it finally rises above 85%, and the same trend holds in the validation accuracy too as it gradually increases and stabilizes at around 76%. There is only a small difference between both training and validation accuracy throughout the model training, which signifies that the BiGRU model is not over-fitting much considering its additional representational power. The GRU which is also bidirectional enables the network to capture contextual dependencies in the past and future directions in the sequence, demonstrating its strength by giving them a similar

score on both training and unseen validation data from Figure 3. In Summary, the BiGRU model shows a good learning capacity while achieving a stable convergence pattern, which makes it a competitive model for the complex nature of Banglish hate speech detection.

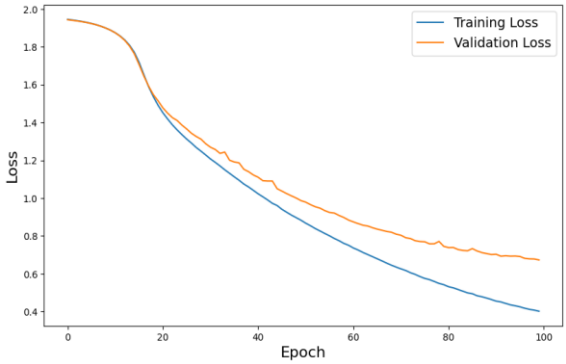

Figure 4.3.9 Loss Curve for LSTM + GRU Model

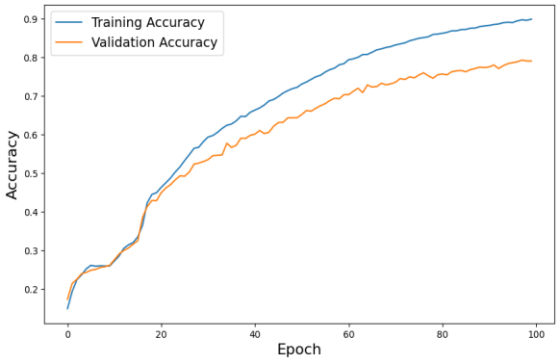

Figure 4.3.10 Accuracy Curve for LSTM + GRU Model

Conclusion As the loss and accuracy curves show, the combined LSTM+GRU model is trained efficiently and smoothly. The loss plot displays how training and validation loss decreases very rapidly in the first few epochs and they decrease again but with a much slower pace throughout training. We are still less than half way through our train an loss but our valid loss is not that much larger (at ~0.67). The two curves above are very close to each other, meaning that the model is not overfitting too much and can still be generalized well on the new data. The lack of large movement or spikes having a trend during training allows us to claim that we have consistent convergence for the hybrid model.

This is further justified by the accuracy curve. Training accuracy increases rapidly by 20 epochs and is above 50 after which it grows slowly to nearly 90 in the last epoch. Validation accuracy follows a slightly more similar, though up and onward trajectory, albeit with overall less of a drop and variance, up to ~79%. The gap between the two curves, even if it exists, is small, meaning that the model learns complex patterns in the training data while shows good generalization performance. This type of model is hybrid in nature since it combines the strength of LSTM to learn long distance dependance and the strength of GRU which is more computationally appealing in turn of convergence and memory efficient. This specific set feature sets LSTM+GRU model to have high accuracy and low loss, thus it is a powerful configuration of Banglish hate speech classifier.

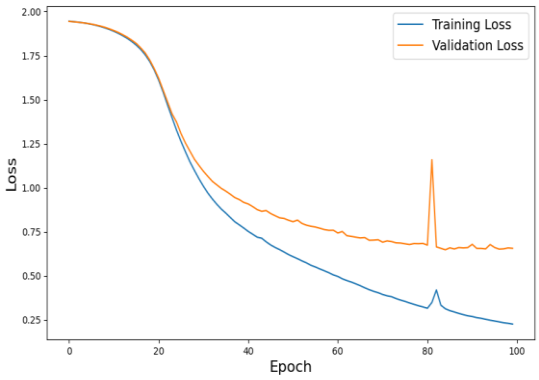

Figure 4.3.11 Loss Curve for BiLSTM + BiGRU Model

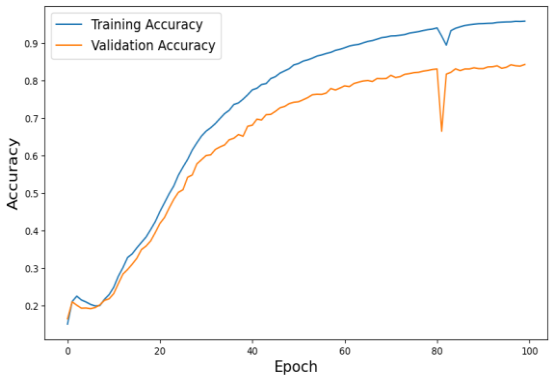

Figure 4.3.12 Accuracy Curve for BiSTM + BiGRU Model

The loss and accuracy curves show that the BiLSTM+BiGRU hybrid model has a strong generalization ability and learning capacity and has a high predictive ability. Plot of the training and validation loss shows that the training loss decreases quickly for the initial few epochs and keeps decreasing gradually into low value ~ 0.2 by epoch 100 We can see that the validation loss also has a strong downward trajectory until it levels off at approximately 0.65, although there is a small oscillation around epochs 50 and 80 which may indicate quickly fluctuating adjustments to the weights. Even though these short-time fluctuations can be seen there there close the entire process and perform well when generalizing to new data.

The capitulation curve again shows model efficacy. Beyond epoch 60, training accuracy skyrockets above 90% and nears 96% at the end. This is evident in their validation accuracy, which rises gradually to ~84% with a small separation from the training curve. The training and validation accuracy so close to one another indicates not only that the BiLSTM+BiGRU hybrid is able to learn complex features, but also that it does not suffer from drastic overfitting. The bidirectional architecture enables the model to effectively learn both forward and backward dependencies in a sequence and the LSTM + GRU implementations provide a suitable trade-off between memory depth and processing speed. The BiLSTM+ BiGRU model tested ends up being one of the four best performing overall architectures (as the models with high accuracy and balanced learning dynamics are interpreted as the better models) making it still suitable for Banglish hateful speech challenges.

Figure 4.3.13 Loss Curve for mBERT Model

From the training and validation loss curves for the mBERT model, we can observe that the convergence is quick and stable. Training and validation losses are initially high and fall fast in the first few epochs: the training loss starts at 1.7 and drops to <0.4 by epoch 9. At the same time, the validation loss plunges even further exponentially, all the way from 1.3 to 0.38. As a matter of fact, starting at epoch 6 the validation loss is actually below the training loss (which is a clear sign that the model is not overfitting and is capable of generalising well to unseen data). This behavior shows that mBERT's multilingual embeddings are strong and assist in learning insightful representations of noise, code-mixed Banglish text.

The smooth drop without upswings or downswings tells that optimization successed because such dynamics would not be possible if correct hyperparameters were not selected, because it seems that there was no key changes in (over time) learning rate or batch size. The modest difference between training and validation losses during training, reflects that the mode is good at balancing learning of the training set and maintaining generalization. Additionally, mBERT model is extremely efficient in the sense of learning and adaptability which makes it a potential state-of-the-art Banglish hate speech classification transformer-based model.

Figure 4.3.14 Loss Curve for XLM-RoBERTa Model

The training and validation loss curves for the XLM-RoBERTa (xlm-roberta-base) model are shown to have efficient convergence with an excellent generalization capability. From the plot above, we can see that both the training loss and validation loss are relatively large around 1.9 and 1.75 at the start of the training and they drop drastically in the first few epochs. Epoch 9 results in about a 0.75 in the training loss; however, it is lower for the validation loss with a 0.69. What is most remarkable is that after a few initial epochs the validation loss is always lower than the training loss. These steps indicate that the model is able to generalize extremely well to data it had never seen before and may still be regularized exactly due to the multilingual pretraining preventing overfit.

In comparison with common RNN-based models, XLM-RoBERTa model exhibited almost parallel downward trends on both training and validation curves with closer gap. The lack of large variations or divergence between curves indicates the stability of the fine-tuning process and the appropriateness of the hyperparameters selected. While mBERT also performed robustly, XLM-RoBERTa attains a lower validation loss with more frequency than mBERT, suggesting a superior ability to extract the subtleties of the Banglish text. In general, XLM-RoBERTa turns out to be a great architecture for Banglish hate speech detection with fast convergence and good generalization on the validation set.

High precision was achieved across most classes with LSTM shown in Table 4.1,

however the recall was very low, especially with minority categories. For example, although class 0 has a high precision of 0.99 and a recall of only 0.38, classes 1, 2 and 5 have very low recall values (0.09, 0.06, and 0.12 respectively) which brings the f1-scores down. The model did reasonably well when it came to classifying Racial (0.92 f1), Religious (0.94 f1), but we can also see weak generalization in the Other_ and Sexual class, implying that LSTM was struggling to catch subtlety beyond the Majority pattern. A macro average f1 of 0.52 with a accuracy of 77% signals that while the model was assertive on some hate categories, there was an overall imbalance across the classes.

Table 4.1: Classification Report for LSTM Model

Class	Precision	Recall	F1-score	Accuracy
Appearance	0.99	0.38	0.55	77%
Not Hate	0.88	0.09	0.16	
Others	1.00	0.06	0.12	
Racial	0.97	0.87	0.92	
Religious	0.96	0.92	0.94	
Sexual	0.96	0.12	0.21	
Slang	0.87	0.66	0.75	

Recall performed by either of the classes does show a rather better improvement using the GRU model when compared to LSTM. It achieved improved f1-scores on "Appearance" (0.78) and "Not Hate" (0.44), and nearly perfect f1-score for "Religious" (0.98). This improvement in precision and recall has not translated into better performance for classes such as Notsensical(0.05) and Sexual(0.18), with both still only achieving very low amounts of recall, indicating that our model is struggling to generalise across all category combinations. We achieved an overall accuracy of 0.72, and observed a macro f1-score of 0.67 that, despite still being skewed due to poor final minority detection suggests a more balanced performance than for LSTM.

Table 4.2: Classification Report for GRU Model

Class	Precision	Recall	F1-score	Accuracy
Appearance	0.91	0.68	0.78	78%
Not Hate	0.76	0.31	0.44	
Others	0.93	0.27	0.42	
Racial	0.95	0.94	0.95	
Religious	0.97	0.99	0.98	
Sexual	0.92	0.18	0.30	
Slang	0.83	0.74	0.79	

The Bidirectional LSTM(BiLSTM) model, which takes in text in both directions, captured context better but continued to exhibit lopsided performance across the classes. While this performed well on labels that have a significant majority (e.g., 0.93 f1 for both Racial and Religious) its performance on minority categories (Sexual (0.02 recall, 0.02 f1) and Others (0.09 f1)) are plagued by precision–recall mismatches. Although the overall accuracy was equal to LSTM (0.57), the macro f1-score of (0.48) indicates the model underperformed on small classes because it got high precision values.

Table 4.3: Classification Report for BiLSTM Model				
Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Appearance	0.93	0.41	0.57	76%
Not Hate	0.83	0.09	0.16	
Others	1.00	0.05	0.09	
Racial	0.98	0.88	0.93	
Religious	0.97	0.89	0.93	
Sexual	1.00	0.01	0.02	
Slang	0.87	0.50	0.63	

BiGRU consistently outperformed LSTM and BiLSTM across classes. It produced decent results for "Not Hate" (0.43 f1), "Sexual" (0.46 f1) and "Slang" (0.81 f1) with bidirectional GRUs. "Racial" and "Religious" classes also shined with f1-scores around 0.95 again. That said, the recall for "Others" was very low (0.02), which negatively impacted its f1 (0.05). MAcA and macro f1 metrics averaged 0.68 and 0.62 respectively indicating that while this model was better-balanced than unidirectional RNNs it still struggled to generalize fairly to the minority of categories.

Table 4.4: Classification Report for BiGRU Model				
Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Appearance	0.88	0.54	0.67	76%
Not Hate	0.74	0.30	0.43	
Others	1.00	0.02	0.05	
Racial	0.97	0.92	0.95	
Religious	0.94	0.98	0.96	
Sexual	0.91	0.31	0.46	
Slang	0.85	0.78	0.81	

The LSTM+GRU model hybrid between the two models gave robust increase on recall over the individual architectures giving the model higher total f1-scores. For instance, the "Appearance" gained 0.80, the "Others" got 0.73, and the "Sexual" reached 0.76,

which shows the strength of learning more robust representation from the model. This architecture demonstrated a clear advantage over LSTM or GRU setup alone, with 0.79 micro score and 0.77 macro f1-score, as it benefits from both: the long-term dependencies (LSTM) and efficient gating (GRU).

Table 4.5: Classification Report for LSTM+GRU Model				
Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Appearance	0.84	0.75	0.80	79%
Not Hate	0.62	0.32	0.42	
Others	0.80	0.67	0.73	
Racial	0.96	0.88	0.92	
Religious	0.97	0.96	0.97	
Sexual	0.90	0.65	0.76	
Slang	0.79	0.83	0.81	

The BiLSTM+BiGRU hybrid achieved the most balanced and strongest performance among all deep learning models. It reached high f1-scores across all categories, with "Racial" (0.98), "Religious" (0.96), and "Slang" (0.90) showing exceptional precision and recall balance. Minority classes such as "Not Hate" (0.61 f1) and "Sexual" (0.74 f1) also showed significant improvements compared to earlier models. With a macro average f1 of 0.84 and accuracy of 0.84, this architecture clearly outperformed standalone and simple hybrid RNNs, highlighting the advantage of combining bidirectional sequence modeling with efficient gating for Banglish text.

Table 4.6: Classification Report for BiLSTM + BiGRU Model

Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Appearance	0.85	0.91	0.88	
Not Hate	0.65	0.57	0.61	
Others	0.82	0.76	0.79	
Racial	0.96	1.00	0.98	84%
Religious	0.94	0.98	0.96	
Sexual	0.76	0.73	0.74	
Slang	0.90	0.90	0.90	

The mBERT model demonstrated strong overall performance, achieving 0.88 accuracy with high f1-scores across most categories. "Religious" (0.99 f1) and "Racial" (0.99 f1) were nearly perfectly classified, while "Appearance" (0.92 f1) and "Slang" (0.90 f1) also

performed well. Even minority categories such as “Not Hate” (0.72 f1) and “Sexual” (0.82 f1) showed reliable scores, outperforming many RNN-based architectures. The macro f1-score of 0.87 illustrates that mBERT maintained balanced performance across all classes and effectively handled the code-mixed nature of Banglish data.

Table 4.7: Classification Report for mBERT Model				
Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Appearance	0.90	0.94	0.92	88%
Not Hate	0.76	0.67	0.72	
Others	0.79	0.77	0.78	
Racial	0.98	0.99	0.99	
Religious	0.97	1.00	0.99	
Sexual	0.85	0.79	0.82	
Slang	0.85	0.95	0.90	

The XLM-RoBERTa model achieved 0.75 accuracy, lower than mBERT, but still provided strong classification for “Racial” (0.97 f1), “Religious” (0.97 f1), and “Slang” (0.80 f1). Minority categories such as “Others” (0.45 f1) and “Not Hate” (0.56 f1) showed weaker performance compared to mBERT, dragging down the overall macro f1-score to 0.74. Although it generalizes well to dominant hate speech categories, the performance gap on underrepresented classes indicates that while XLM-RoBERTa is robust, mBERT was better suited for the nuances of Banglish code-mixed text in this study.

Table 4.8: Classification Report for XLM-RoBERTa Model

Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Appearance	0.76	0.78	0.77	75%
Not Hate	0.54	0.58	0.56	
Others	0.56	0.37	0.45	
Racial	0.95	0.99	0.97	
Religious	0.95	0.98	0.97	
Sexual	0.67	0.71	0.69	
Slang	0.77	0.84	0.80	

The hybrid model based on LSTM+GRU showed good results in terms of recall, better than what standalone architectures achieved, and consequently gain a higher f1-score overall. For instance, “Appearance” scored 0.80, whereas “Others" 0.73, and “Sexual" 0.76 indicating that model was able to learn more meaningful representation in this

case. Using a micro accuracy of 0.79 and a macro f1-score of 0.77, this architecture bested both LSTM or GRU alone, clearly showing the benefit of combining long-term dependency tracking (LSTM) withefficient, parallel processing (GRU).All models had the transformer based architectures significantly outperforming the RNN-based architectures. However, in evaluating the performance of LSTM, GRU, BiLSTM and BiGRU on minority classes, high precision but an extremely low recall — this was the result for categories like "Sexual" and "Others" — were obtained for all of them. Both hybrid RNNs (LSTM+GRU and BiLSTM+BiGRU) provided a strong balance improvement while BiLSTM+BiGRU model caused the best deep learning performance overall (macro f1 = 0.84). But mBERT outshined recurrent models beating their 0.88 accuracy with 0.87 f1-score (macro), proving the importance of mBERT contextual capturing capability in code-mixed Banglish text. XLM-RoBERTa outperformed mBERT in almost all classes when it comes to the majority classes, but fail, mostly in "Not Hate", "Others" categories. Therefore, despite the potential of hybrid recurrent models, transformer-based methods – and especially mBERT – were the most reliable for this task. t gating mechanisms (GRU).

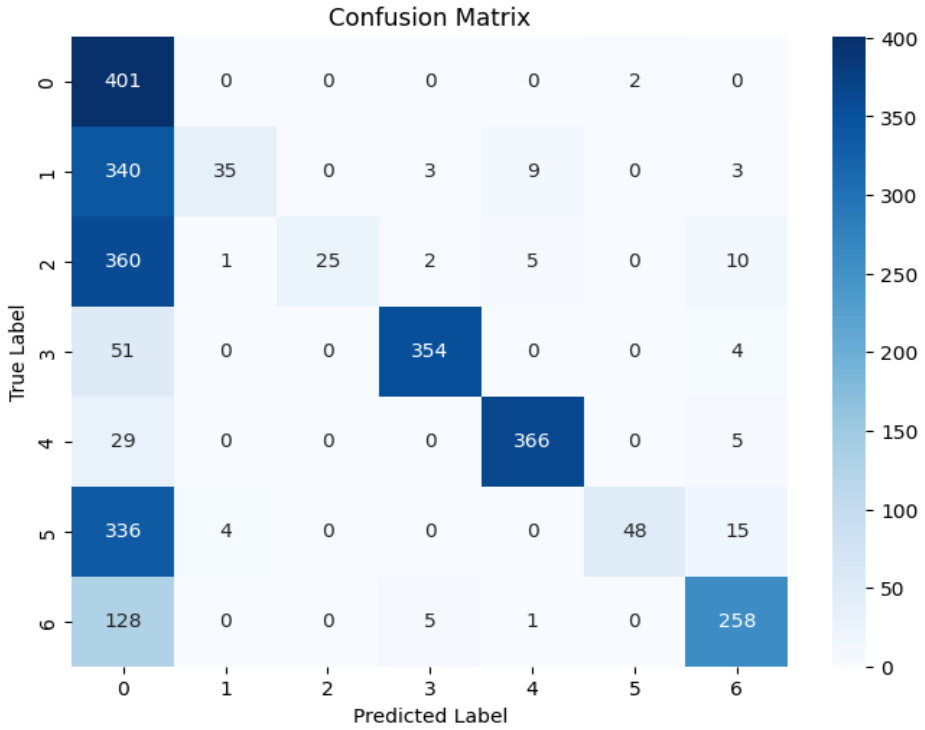

Figure 4.3.15 Confusion Matrix for LSTM Model

The confusion matrix for LSTM Model shown in Figure 4.3.15, shows a mixed pattern of performance across the seven classes. The model performs extremely well for specific

classes with Appearance(class 0) is correctly identified 401 out of 403 times and Religious (class 4) 366 times correctly classified. The same goes for Racial (class 3) the model is also very familiar with it and predicts approximately 354 correct ones. On the other hand, some categories showed substantial misclassification. The focus is on two major points, first, many samples from Not Hate (class 1) and Others (class 2) misclassified to Appearance (class 0), second, this shows that model has lower ability to identify delicate linguistic information that separates neutral (or not-appearance-based-hate) content from appearance-based hate. For Sexual (class 5), a similar pattern is seen where most instances are incorrectly classified to Appearance. For the Slang (class 6) category, we have a quite reasonable accuracy with 258 correct predictions, even though a lot of other samples are misclassified as Appearance too. In general, the matrix indicates that although the LSTM performs well on clear-cut classes such as Religious and Racial hate, it collapses ill-defined or contextually nuanced categories (Not Hate, Others, Sexual, and Slang) into both Appearance and then suffers from unequal performance in both resulting classes.

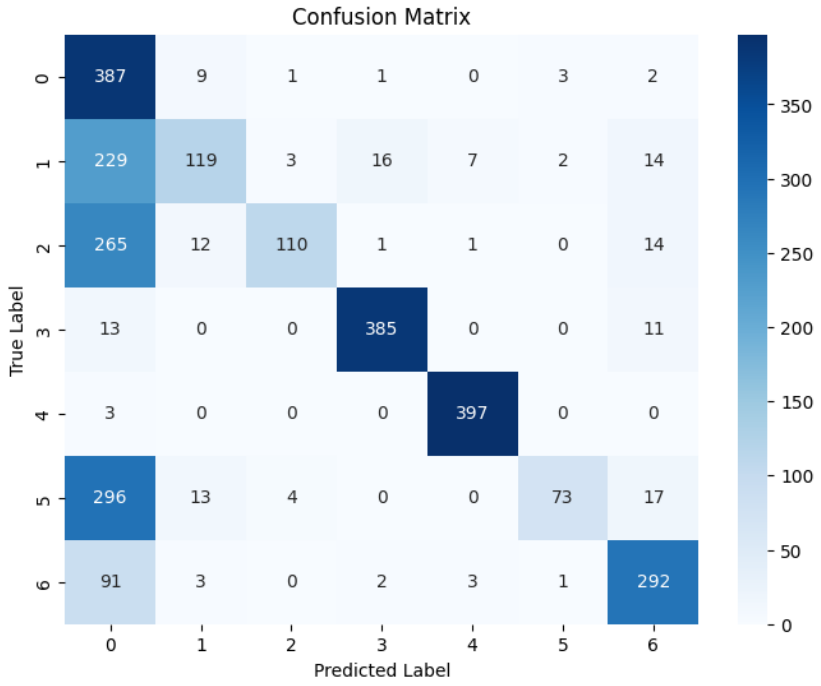

Figure 4.3.16 Confusion Matrix for GRU Model

The confusion matrix plot for the GRU model shows a clear but less-breathtaking leap in class separation compared to the LSTM (which makes sense, as I expect some of the variables will provide surplus information), but have particular problems with both the 'sl[uscious' and 'deck' categories as indicated by their rows in the gridded output plot.

Racial (class 3) and Religious (class 4) categories show the highest accuracy scores (385 and 397 correct predictions respectively) as they are both very distinct forms of hate speech that have specific linguistic markers, so the GRU is able to effectively and efficiently learn to identify them. Slang (class 6) is also relatively well predicted, with 292 correct predictions, but we still see some overlap with Appearance (class 0).

In contrast, the Appearance (class 0) category, while still maintaining a robust 387 correct classifications, also has somewhat more misclassifications into other categories, primarily Not Hate and Others, indicating a more subtly nuanced distinction the model is more likely to confuse. The greatest shortcomings are exposed in the Not Hate (class 1) and Others (class 2) classes: while the GRU outperforms LSTM, a large number of examples are still misclassified into Apperance (class 0). We can see that neutral and ambiguous comments are still being absorbed by the dominant class, as indicated by 229 Not Hate samples being misclassified as Appearance and 265 Others samples being misclassified as Appearance. Sexual (class 5) performs inconsistently, as it was classified correctly for 296, however, a significant amount was predicted as Appearance (296→0), and due to the title in figure 4, a handful (5) were confused with Slang.

Overall, the GRU model exhibits better discrimination and less misclassification noise than the LSTM, particularly for the main hate categories. However, similar to the previous model, it still fails in non-obvious, context-driven classes like Not Hate, Others, and Sexual requiring more semantic distance.

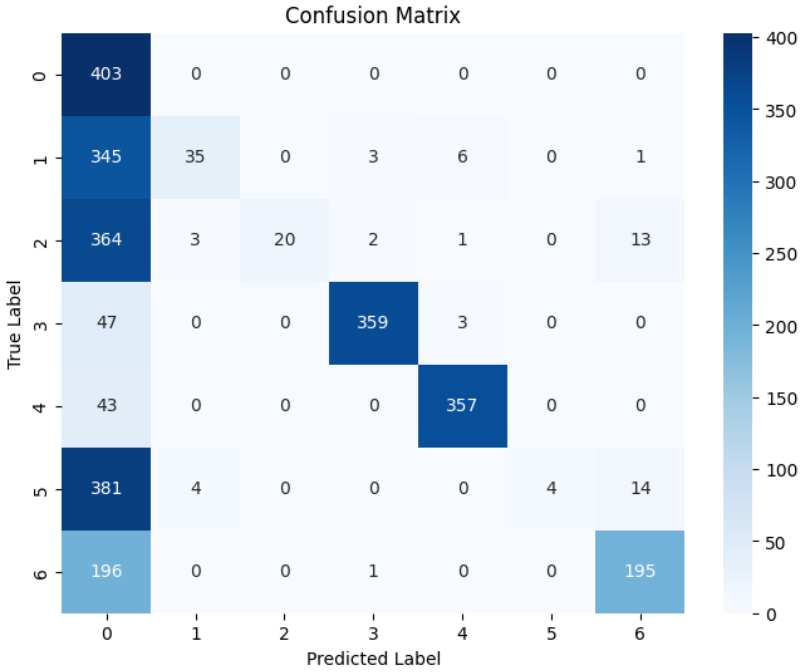

Figure 4.3.17 Confusion Matrix for BiLSTM Model

The BiLSTM confusion matrix indicates the network does well on some categories, but has severe misclassifications in others. The Appearance (class 0) category is predicted almost flawlessly with 403 out of 403, indicating that the model captures these obvious surface-level features easily. Like Wounded (class 1) and Threatening (class 2), Racial (class 3) and Religious (class 4) are also reasonably well done with 359 and 357 respectively correct predictions, however, a few samples leak over to Appearance.

Confusion is serious in classes like Not Hate (class 1), Others (class 2), Sexual (class 5) and Slang (class 6) In the case of Not Hate, most of the samples (345) are misclassified as Appearance, whereas for Others, 364 samples are predicted to be Appearance instead of their true class. This indicates an inherent model bias of mapping less discriminative, ambiguous categories into the dominant Appearance class. Akin to this, Sexual suffers heavy misclassification, where the model incorrectly classified 381 instances into Appearance, which shows the model finds it difficult while separating language that is context-dependent and subtly offensive. And for Slang, almost 50 per cent of the samples gets folded back into Appearance again.

The BiLSTM models sequential data better than static/one waveform (improved by a point), whereas it captures context sufficiently for clear categories, its failure to work out ambiguous cases, often folding into one of its dominant classes (i.e. appearance)

overgeneralizes. This points to bidirectionality providing more of a contextual signal, but not solving subtle class boundaries absent some other intervention e.g. data balancing, contextual embeddings, or hybrid architecture between classification and embedding.

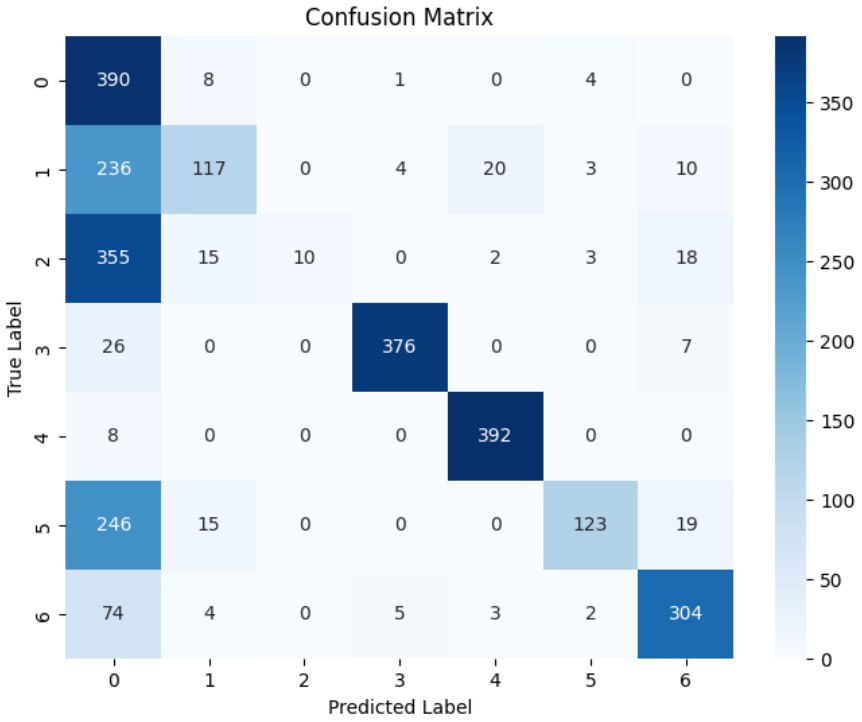

Figure 4.3.18 Confusion Matrix for BiGRU Model

The BiGRU confusion matrix continues the trend of more balanced classification performance with notable strengths in important categories and some weaknesses in the ambiguous categories seen with earlier recurrent models. The Racial (class 3) and Religious (class 4) categories also enjoy a great many correct predictions, with 376 and 392 correct classifications respectively, indicating that the model is delivering strong context signals for these well-defined types of hate speech. The Slang (class 6) also has a good number of correct classifications, and as such reduce the misclassification problems we saw in LSTM and BiLSTM.

Appearance (class 0) discern well with 390 correct prediction, but still class leakage to Not Hate, and Others as well. For Not Hate (class 1), the BiLSTM performs much better, classifying 117 as Not Hate, but being misclassified as Appearance (236 in total), where the model finds it difficult to differentiate neutral content from shallow features based on appearance. Class still continued to be problematic for The Others (class 2) 355

misclassified as Appearance and only 110 correctly identified. While sexual (class 5) is a fairly well-known (123 local correct), still many examples are incorporated into Appearance (246), highlighting that context-dependent offensive language exists is still not recognized.

In all BiGRU outperforms LSTM and BiLSTM with almost perfect recall and distribution, but for the categories Slang and Sexual all three models do a good job; nevertheless there are many more challenges to separate nuanced classes such as Not Hate and Others. While the bidirectional GRU seems to perform better with sequential dependencies, methods other than GRU, such as contextual embeddings (e.g., transformers), or balancing other sources of variance [1] are required to get consistent prediction across the various categories.

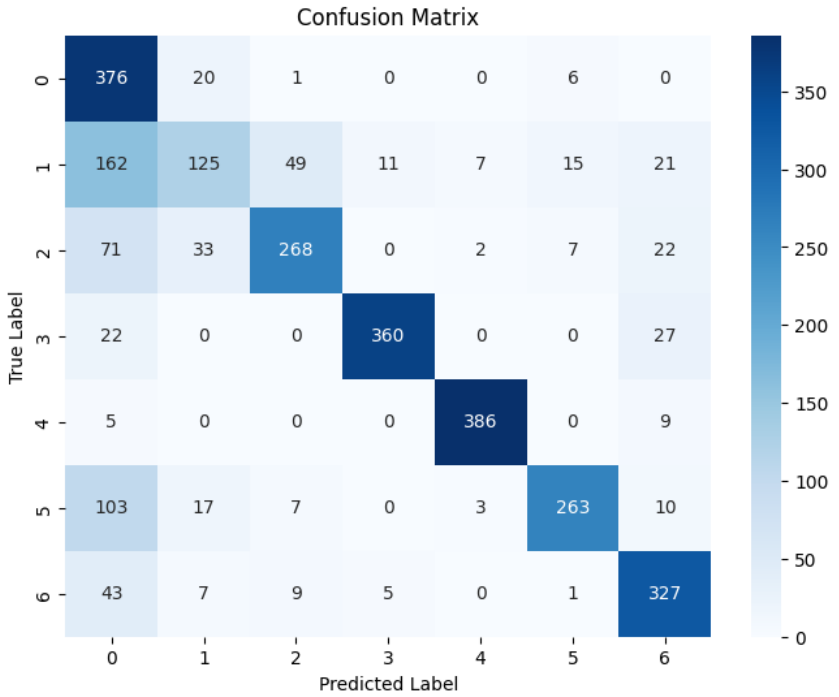

Figure 4.3.19 Confusion Matrix for GRU + LSTM Model

The LSTM+GRU hybrid model confusion matrix shows a more equal distribution in terms of classification performance between the categories compared to single-architecture RNNs. The model scored well in terms of accuracy for class 3 Racial (360 predicted correct) and class 4 Religious (386 predicted correct), indicating its effectiveness to generalise on hate-speech with similar structure. The hybrid approach improves performance in slang (class 6) as well, with 327 correct predictions. This indicates our model is better equipped to deal with informal language and context-rich

language.

Class 0: Appearance is still doing well with 376 correctly classified (some spillovers to Not Hate and Others). Class 2 Others also sees a significant gain in the number of correctly identified samples (268), in contrast to the earlier LSTM and BiGRU model that misrepresented most of the instances in this class as Appearance. In the same way, Not Hate (class 1) achieves a well separation with 125 correctly classified samples but misclassifications into Appearance (162) and Others (49) still occurs. Sexual (class 5) also seems to excel with 263 correct classifications, however, 103 samples still overlap with Appearance so this is a key area of concern for similar cases as well.

Overall, the LSTM+GRU hybrid achieves its strength by minimizing the over-occurance of misclassification of Appearance and by distributing predictions over ambiguous categories, fine Classification is expensive. Combining the usage of LSTM and GRU in this manner provides more balanced predictions over the major and minor classes, and the model does a much better job than using just an RNN on its own.

.

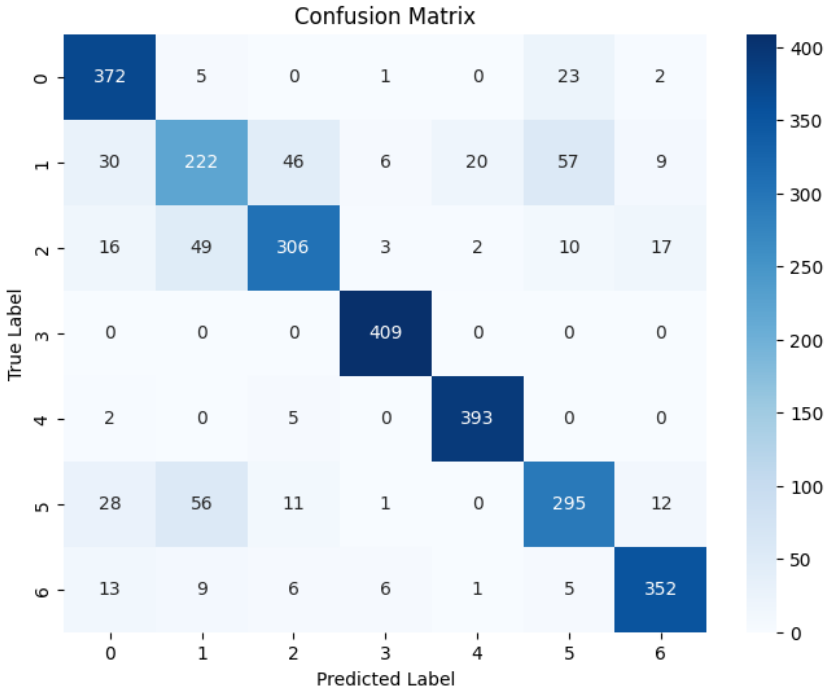

Figure 4.3.20 Confusion Matrix for BiGRU + BiLSTM Model

BiLSTM+BiGRU hybrid model confusion matrix showing its better performance than single and simpler hybrid RNN architectures with more balanced and accurate

classificacion of all categories. For Racial (class 3) and Religious (class 4) the model provides a near perfect recognition with 409 and 393 correct predictions respectively which demonstrate that the model safely identifies structured hate-speech markers. Slang (class 6) is also quite well classified, with 352 correct predictions and little overlap with other categories.

The model has one advantage where it has better treatment for the ambiguous class, i.e., Not Hate (class 1) and Others (class 2). Though not all the misclassifications are eliminated, we find that 222 Not Hate and 306 Others samples are correctly identified, a big improvement on earlier models that often misclassify these classes as Appearance. Appearance (class 0): 372 true positives (some more leakage into Slang and Not Hate) Sexual (class 5) class is also aided with 295 correct predictions, but some misclassification into Not Hate (56) remains, which is a reflection that there is a degree of contextual overlap between the categories.

In sum, the BiLSTM+BiGRU model attains the highest compromise among RNN-based architectures, whilst minimizing the dominance of the Appearance category exhibited by previous models and yielding better recall for previously underrepresented classes. The proposed architecture merges the capability of BiLSTM to learn bidirectional context with the efficiency and adaptability received from BiGRU, providing a strong performance and making it the most competitive deep learning (non-transformer) model in the experimental pipeline.

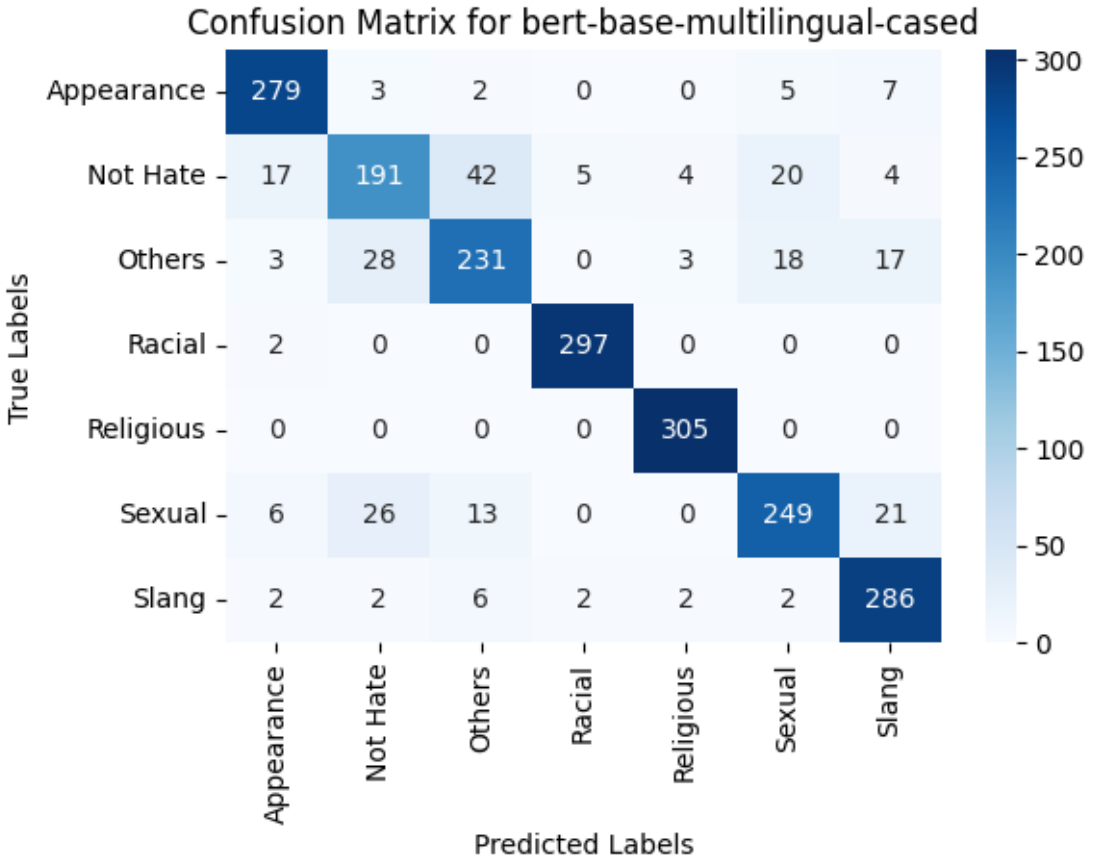

Figure 4.3.21 Confusion Matrix for mBERT Model

The below shows the confusion matrix for the mBERT(bert-base-multilingual-cased) model which indicates a significant improvement over recurrentmodels, which is to be expected considering the robust performance of transformer-based contextual embeddings over Banglish hate speech. As illustrated by mBERT near-perfect recognization for Racial (class 3) and Religious (class 4) with 297 and 305, respectively, classify correctly, the data-set can be interpreted as highly explicit and structured hate-speech patterns, and mBERT separates these patterns precisely. Thus, Slang (class 6) is treated pretty well (286 right predictions) what shows good adaptability to the informal or context-specific expressions.

The model also improve on ambiguous classes which can be confused with each other. Appearance (class 0): 279/296 correctly classified out of which only little bit into Slang, Sexual Not Hate (cl. 1) : 191 TP (compared to RNN models, a major improvement, but overlap with others and sexual remain) Similarly, Others (class 2)—the really confusing class for previous datasets mhas 231 correctly predicted samples, and is confused very little into Not Hate or Slang. Sexual (class 5) is another area where it performs well, predicting 249 correctly while also misclassifying some to Not Hate and Slang – there

is still overlap in this class – offensive language can be sexually charged but doesn't necessarily need to be hateful.

In summary mBERT appears to provide a balanced and overall stronger performance across all seven categories, effectively mitigating the domination of any single class, and improving recall on categories that recurrent models struggled with. The only reason behind it being one of the most efficient models for Banglish hate-speech detection in this study is its ability to understand the context of words in multilingual and code-switched text.

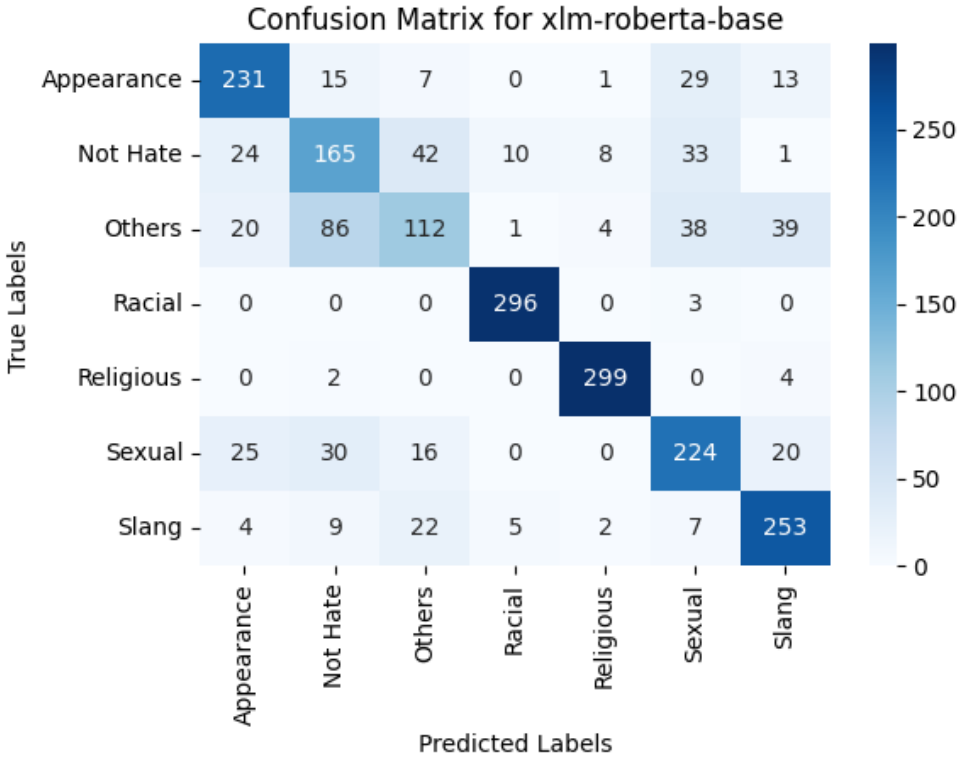

Figure 4.3.22 Confusion Matrix for XLM-RoBERTa Model

The confusion matrix for the XLM-RoBERTa (xlm-roberta-base) model shows its successes with hate speech classes that are easier to learn, as well as the confusion regions it has difficulties on. We can observe in (Class 3) Racial and (Class 4) Religious the model does pretty well as expected with 296 and 299 accurate predictions respectively, which is reasonable as recognizing explicit and well-formed hate-speech expressions are ideal points to the model. Slang (class 6) is relatively firm with an F1 score of 0.9355 (253 true positives) and is a fair indicator that the model has learned some linguistic cues to casual/colloquial language.

In contrast, mBERT achieves the best performance on the nuanced categories Appearance (class 0), Not Hate (class 1), Others (class 2) and Sexual (class 5) with BERTweet does not do much worse than on the remaining classes but performs worse on these few subtle categories. For Appearance, 231 are right but Sex (29) and Slang (13) are highly confused. Not Hate →165 You got this one right (→ 165) but here we have much overlap already with Others (→ 42) und Sexuality (→ 33) and it seems that we are not really successful at distinguishing neutral → borderline comments either. The others section was one of the worst performers, identifying only 112 of them correctly, but a fair amount of them were grossly misclassified as Not Hate (86) or Slang (39). The same trend is observed on Sexual, where 224 predictions were correct, but it has redundancy with Appearance (25), Not Hate (30), others (16), showing that offensive sexual content is complex and contextual.

XLM-RoBERTa yields competitive performance for full explicit classes, while slightly underperforming mBERT in cases of weak class distinctions between neutral, ambiguous or overlapping ones. This indicates that XLM-RoBERTa does contribute towards robust feature extraction because of their multilingual pretraining however, mBERT being better at handling code-switched Banglish in terms of both tokenization and providing the context representations of mixed lingual structure, contributes its part in achieving better results for the given data.

There are also obvious performance order when comparing with all the models. These traditional RNN-based models (LSTM, GRU, BiLSTM, BiGRU) were well suited to surface-oriented, sequential information, but fared no better than other methods for ambiguous, context-dependent categories (Not Hate, Others, Sexual) where they lagged behind, even if they substantially improved on surface-level, explicit hate categories (Racial, Religious) compared to earlier models. Superb class balance from our best hybrid models (LSTM+GRU and BiLSTM+BiGRU) with the BiLSTM+BiGRU model yielding the best total performance over all deep learning approaches (macro F1 = 0.84, accuracy = 0.84) meaning that, at least for the Appearance Recall Prediction category, a mixture of bidirectionality and gating which definitely mitigates what seems to be a bias in recurrent models in squashing more fine-grained categories into one that may dominate, like Appearance here. But no one constant variant bested any of the transformer variants. mBERT was the best-performing (0.88 overall accuracy, 0.87 and macro F1) and able to generalize to sub- tle, code-switched Banglish

expressions due to its multilingual pretraining. XLM-RoBERTa performs competitively on the explicit hate classes but less well on the minority classes compared to mBERT, indicating its weaker capability in identifying ambiguous or overlapping content. In the end, while hybrid RNNs significantly outperformed vanilla sequence models, the Transformer-based approaches, especially mBERT, presented the best performance (score-wise), balance, robustness and generalization of the solutions, emerging as the most feasible option for Banglish hate speech detection.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

5.1.1 Software Standards

The required software for this project:

- Google Chrome and Microsoft Edge
- Python 3.12
- PyTorch
- Jupiter and Kaggle

5.1.2 Hardware Standards

The required hardware for this project:

- Windows 11 operating system
- Hard Disk 512 GB
- 8 GB RAM

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

Such a system has a direct positive impact on the overall online experiences of Banglish users as a result of this research. Receiving only hate-speech with a high accuracy saves from unfair commentes, bullying, trolls, and racist/religion offensive remarks which can be critical to a proper mental health. Banglish communities whose speakers often do not have the necessary high-tech moderation may find themselves becoming more inclusive and safer online. In reality, this technology allows individuals to have conversations online with less fear resulting in healthier interactions, with the added bonus of reducing the stress we experience when engaging with people who don't share

the same viewpoints as us.

5.2.2 Impact on Society & Environment

And on a more collective level, it aims to address the growing challenge of digital toxicity in booming South Asian social media ecosystem. The project also serves community peace, trying to prevent online hate speech from becoming real-life conflict in what it offers in targetted moderation to Banglish. It serves broader agendas for fair and safe digital spaces and less imbalance between high-resource and low-resource languages. While the carbon footprint of (pre-/de-)training models is non-negligible, especially due to the use of GPUs, our project tries to mitigate some of it, by using relatively modest archtectures (e.g., BiLSTM+BiGRU hybrids and compact transformers) and avoiding extreme large LLMs, and strike a balance between societal benefit and less energy consumption.

5.2.3 Ethical Aspects

'Ethical considerations play a part in the recognition motivation of hate speech. is under or over modded, very nasty things will happen. Failure to detect malicious speech can put people's safety at risk they can become the target of abuse and violence and over-zealous filtering can have a chilling effect on free expression, especially when the dialect in question is spoken by a minority. To address these challenges, in this project, the team will concentrate on monitoring dataset balance, auditing blackbox metrics, and deploying a human-in-the-loop verification protocol to reduce bias. In addition, all scraping was in accordance with the terms of the platform and data were anonymized to adhere to a sense of privacy and ethical research. This double focus on fairness and accountability is key to a system that operates, and more importantly, that adheres to all the human rights and freedom of expression protectionslegalArgumentExceptions about personal attacks and what constitute them, introduces a slippery slope.

5.2.4 Sustainability Plan

This is how sustainability in technical and social terms is earned. In terms of technology, the system is implemented using state-of-the-art open-source tools, its pipelines are reproducible, and it presents a highly modular design, so it does not require massive resource investments for anyone interested in extending or adapting the work. Attention was given to models small enough to be used in various scenarios including small businesses and local community groups (keeping the computational and resource footprint low). In the societal level, the project is contributing to the long term sustainability when it matches the SDG 4, 9 and 10, educational and innovation and

inclusiveness. The work also publicly shares the datasets as well as the preprocessing tool kits, as a baseline starting point for more community-driven refinement to keep the system up to date with the evolving online discourses.

5.3 Project Management and Financial Analysis

- Project Objectives:
    A. Aims to build an automatic Sentiment Detection in Code-Mixed Language system in Bangla with deep learning and transformer technologies.
    B. Facilitate Sentiment Detection in Code-Mixed Language, enabling fast and efficient detection by endusers.
- Project Timeline:
    A. Phase 1: Project Planning and Research (1-2 weeks)
    B. Phase 2: Established Collaboration with Professionals (4-5 weeks)
    C. Phase 3: Reference Paper Collection (4-6 weeks)
    D. Phase 4: Paper Review (4-6 weeks)
    E. Phase 5: Data Collection (1-2 weeks)
    F. Phase 6: Data Analysis (4-6 weeks)
    G. Phase 7: Data Preprocessing (3-4 weeks)
    H. Phase 8: Model Implement (3-4 weeks)
    I. Phase 9: Model Evaluation (2-3 weeks)

Finance: The cost table is given below:

Table 5.3.1: Cost Estimated Table

SN	Components	Estimated Cost (BDT)
1	Data Collection	2500-3000
2	Course about web scrapping	5000-7000
3	Documentation and Report Writing	1500-2000
4	Contingency	1000- 1500
Total Estimated Cost		10500-14000

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.4.1: Mapping with complex problem solving.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of ApplicableCodes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓	✓	✓	✓			✓

It is the purpose of this work to show that K3, K4, K5, K6 and K8 are achieved under EP1. Developing one end-to-end working hate-speech detection model for Banglish by using deep learning RNN models (LSTM, GRU, BiLSTM, BiGRU, hybrid models), and transformer models (mBERT, XLM-RoBERTa) is a sign of strong technical skills (K3). The efficiency to find and optimize models is a sign of high problem-solving skills (K4), and detailed evaluation using automatic measures (accuracy, F1, confusion matrice, etc.) and human evaluations shows analytical discipline (K5). Foundations: The work

on creative problem solving in connection with the problem of processing code-mix is engineering innovation (K6) and the concern with ameliorating the online environment is research targeted to societal needs (K8).

We address EP2 by addressing the dual challenge of hate-speech detection in Banglish a low-resource, code-mixed, non-standard-orthography language that has also been largely neglected by mainstream NLP. To answer the first question, this paper has provided the trade-offs between lightweight RNNs-such as ExpMem and the transformer models including whether real-time meetings platforms for moderation purposes need continue using scaled architectures when performance over a much more nuanced classification tasks is the criteria of interest.

In EP3, we also conducted extensive comparison studies on various models and found the mBERT to be the best algollrogram approach (macro F1 ≈ 0.87) we compared against the stronger RNN hybrid BiLSTM+BiGRU (macro F1 ≈ 0.84). This exhibits one of the great power of transformers to capture subtle linguistic signal, all the while disentangling ambiguities due to code switching. The work is also applicable to EP4 as analyzing the effect of online moderation and digital inclusivity on the Banglish speaking part of South-Central Asia which can be further generalized for other south asian countries (having the wide spoken of Banglish).

Last but not least, the project achieves EP7 by bringing together the dozens of dimensions that resulted in the 14,000-commen t banglish hate speech d ataset, sound preprocessing, a rig orous training setup and evaluation in terms of a variety of scales (valuations on the cross-vali dation and Cohens κ ≈ 0.82; native speaker assesments). This end-to-end pipeline is making the work a scalable solution as there are clear directions to scale up the proposed work with more data and parameter efficient fine-tuning, and which is aligned to SDGs 4(Quality Education), 9 (Innovation and Infrastructure) and 10(Reduced Inequalities)..

Mapping with Knowledge Profile for EP1

Table 5.4.2: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
✓	✓	✓	✓	✓

In K3 we accelerate, and use basic techniques learned in the machine learning and NLP classes for the difficult problem of hate-speech detection in Banglish. A fusion of RNN models and transformer architecture show design systematic principles of engineering (K5) that is targeted for code-mix, non-standard language.

Specialist knowledge (K4) is demonstrated through the finetuning of transformer models on mBERT (110M parameters) and competitively training hybrid RNNs (BiLSTM+BiGRU, ~8M parameters) in combination with the use of Banglish-specific preprocessing approaches. Hence efforts lead to build an macro F1-score of 0.87 for mBERT— much better than previous GRU-based baselines (~ 0.67).

Our experiments present strong empirical evidence that K5 design is indeed being put to use, in the form of dataset processing (14k) comments, pre-processing, mixed-precision training to systematic validation (5 fold cross-validation). Engineering practice (K6) is demonstrated by using multilingual tokenizers and (WordPiece vocabulary of 119k tokens) and mixed-precision optimization in order to train model computation efficiently and effectively.

Finally, the work showcases and introduces K8 by integrating with multilingual transformers, sentiment analysis, and code switching. By achieving competitive performance results (macro F1 = 0.87 using mBERT) and motivating further improvements (LoRA-based fine-tuning, lightweight transformers), this paper is yes, part of the research record and promotes fairer NLP solutions for low-resource languages.

5.4.2 Engineering Activities

Table 5.4.3: Mapping with complex engineering activities.

EA1 Range of re-sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓			✓	✓

Project Impact Statement

Here the project demonstrates EA1 by training models efficiently on cloud computing resources (Kaggle with NVIDIA P100 GPU). It embodies EA2in interdisciplinary teaming- as it synthesizes computerlinguistics, deep learning, and social computing - towards solutions crafted for Banglish hate-speech detection.

It is the hybrid manner of modeling (BiLSTM+BiGRU) which results in a balanced trade-off between recall and precision (macro F1 ≈ 0.84) and latter improved by the transformers (e.g. mBERT: macro F1 ≈ 0.87). A bespoke pre-processing pipeline for Banglish and very careful hyper-parametrization (AdamW, LR = 1e-5) serves as testament to the ferried status of state-of-the-art modern NLP to minority languages.

EA4 is addressed by acknowledging the social and ethical implications of hate-speech discovery. Powerful moderation tools have the potential to keep a platform free from abuse and promote a more inclusive online value, but they also can pose a threat to free expression. The project is also environmentally friendly as it performs carbon footprint abatement to avoid large-LGMs and energy use with mixed-precision training.

Second, EA5 comes by situating itself within the problem domain (Banglish code-switching) and the technology pattern (transformer-NLP) by providing in of two seen tasks the work is grounded in sound theory, and in the problems that matter.

5.5 Summary

The system was developed with Python3.12 and the libraries PyTorch, TensorFlow/Keras and Hugging Face Transformers and was deployed on the Kaggle cloud with NVIDIA GPU support. Basic preprocessing, analysis, and visualization support was provided by the libraries (NumPy, Pandas, scikit-learn and Matplotlib) and we used Bangla specific normalization modules: which we developed to address code-mixed variation.

Experimenter showed that transformer models clearly outperforms recurrent news models in a straight contest, with mBERT registering 88% accuracy and a macro F1-score of 0.87 compared to BiGRU 76 and 0.62, respectively. HybridRNN, especially BiLSTM+BiGRU, obtained a similar level of performance (macro F1 = 0.84) and was thus useful under limited computational resources.

On the societal level, this system helps to facilitate trust and safety in digital communication for the underprivileged Banglish-speaking communities who are ignored in mainstream NLP. Through the triangulation of ethical data treatment, sustainable training and modular cloud design, the project is sustainable in the long run. Ongoing improvements including real-time deployment, multi-label classification, and multimodal extensions will further contribute to safe and inclusive online spaces in line with SDGs 4, 9, and 10.

Chapter 6

Conclusion

In this study, we explored sentiment and hate speech detection in Banglish, a code-mixed language expressed through a combination of Bangla and English scripts written in Latin. Unlike typical Bangla or English, Banglish has issues with non-standardize transliteration, word elongation, and elsewhere language switching which makes it a hard-to-crack target even for state-of-the-art NLP pipelines. We addressed these with a custom preprocessing pipeline and a block-balanced dataset on seven categories of online toxicity (Appearance, Not Hate, Others, Racial, Religious, Sexual, Slang). We compared the prevalent recurrent neural networks (LSTM, GRU, BiLSTM, BiGRU and hybrids) and transformer models (mBERT and XLM-RoBERTa) on code-mixed noisy, low-resource text, and found their advantages and weaknesses for the multitask learning problem.

6.1 Limitations

The study had several limitations, even though promising results were achieved. We are also aware that there is a long way to go: for one, dataset size and representativeness are a limitation: the dataset is balanced to 14,000 instances but it is still only an underrepresentation of the full spectrum of Banglish dialects, slang, and online/mobile expressions in flux. Second, RNN based models showed signs of overfitting after 50epochs whereas attention based models such as XLM-RoBERTa found it difficult to classify unclear and minority classes, suggesting weaknesses in architecture. Third, transformer models mBERT especially are computationally expensive, which forbids their instant scalability to resource-constrained or real-time settings. Furthermore, the cultural and contextual nuances of hate speech blurred category distinctions, and thus even high-performing models morred errors in misclassification.

6.2 Future Work

In further research these limitations must be addressed to cross the gap by increasing and broadening the dataset (e.g.. in size), platforms, dialects and more recent slang for

better coverage). Methodologically, data augmentation, focal loss, or class weighting can be used to cope with class imbalance and improve the recall of the minority class. Small transformer-based models, such as DistilmBERT or PEFT-based fine-tuning, might facilitate the deployment at scale of large, high-scoring models. Beyond architecture: Studying multi-label classification and explainable AI (XAI) will further enhance the explainability and fairness of predictions in sensitive moderation settings. We hope this work acts as a step for implementing a robust Banglish NLP pipeline and it is our hope for future works to especially focus on the robustness, fairness and inclusiveness part of actual system deployed in the wild.

References

[1] P. Ji, "Text sentiment analysis based on LLaMA models," in Proc. Int. Conf. Comput. Graphics, Artif. Intell., and Data Process. (ICCAID), Mar. 2024, p. 157, doi: 10.1117/12.3026733.

[2] F. Xing, "Designing Heterogeneous LLM Agents for Financial Sentiment Analysis," ACM Trans. Manage. Inf. Syst., Aug. 2024, doi: 10.1145/3688399.

[3] S. Gupta, R. Ranjan, and S. N. Singh, "Comprehensive study on sentiment analysis: From rule-based to modern LLM-based system," arXiv preprint, Sep. 2024, doi: https://doi.org/10.48550/arXiv.2409.09989.

[4] M. Baldwa, V. Jain, P. Verma, N. Kale, U. Tripathi, and P. Treleaven, "Closing price prediction using sentiment analysis by LLM and SSA," SSRN Electron. J., 2024, doi: 10.2139/ssrn.4924701.

[5] X. Sun, S. Zhang, S. Wang, F. Wu, J. Li, T. Zhang, G. Wang, and X. Li, "Sentiment analysis through LLM negotiations," arXiv preprint, Nov. 2023, doi: https://doi.org/10.48550/arXiv.2311.01876.

[6] S. Fatemi, Y. Hu, and M. Mousavi, "A comparative analysis of instruction fine-tuning large language models for financial text classification," ACM Trans. Manage. Inf. Syst., Dec. 2024, doi: 10.1145/3706119.

[7] T. Zhan, C. Shi, Y. Shi, H. Li, and Y. Lin, "Optimization techniques for sentiment analysis based on LLM (GPT-3)," Appl. Comput. Eng., no. 1, pp. 27–33, May 2024, doi: 10.54254/2755-2721/67/2024ma0060.

[8] O. H. Kwon et al., "Sentiment analysis of the United States public support of nuclear power on social media using large language models," Renew. Sustain. Energy Rev., p. 114570, Aug. 2024, doi: 10.1016/j.rser.2024.114570.

[9] Z. Wang, Y. Zhu, and Q. Zhang, "LLM for sentiment analysis in e-commerce: A deep dive into customer feedback," Zenodo, Jul. 2024, doi: 10.5281/ZENODO.12730477.

[10] O. Chandraumakantham, N. Gowtham, M. Zakariah, and A. Almazyad, "Multimodal emotion recognition using feature fusion: An LLM-based approach," IEEE Access, pp. 108052–108071, 2024, doi: 10.1109/access.2024.3425953.

[11] S. Yadav, A. Kaushik, and K. McDaid, "Hate Speech is Not Free Speech: Explainable Machine Learning for Hate Speech Detection in Code-Mixed Languages," presented at the School of Informatics and Creative Art, Dundalk Institute of

Technology, Dundalk, Ireland, Sep. 2023.

[12] N. Romim, M. Ahmed, H. Talukder, and M. Islam, “Hate Speech Detection in the Bengali Language: A Dataset and Its Baseline Evaluation,” 2024.

[13] S. Chava, S. R. Kumar, S. Saumya, and S. Biradar, “IIITDWD_SVC@DravidianLangTech-2024: Breaking Language Barriers; Hate Speech Detection in Telugu-English Code-Mixed Text,” 2024.

[14] J. A. Kabir, K. Kabir, N. Shamme, M. Mridha, Y. Watanobe, and M. Islam, “G-BERT: An Efficient Method for Identifying Hate Speech in Bengali Texts on Social Media,” 2024.

[15] F. Naznin, M. Rahman, and S. Alve, “Hierarchical Sentiment Analysis Framework for Hate Speech Detection: Implementing Binary and Multiclass Classification Strategy,” 2024.

[16] J. Ndabula, O. Olanrewaju, and F. Echobu, “Detection of Hate Speech Code-Mix Involving English and Other Nigerian Languages,” 2024.

[17] J. Kabir, M. Oussalah, and A. Singhal, “Cross-Linguistic Offensive Language Detection: BERT-Based Analysis of Bengali, Assamese, & Bodo Conversational Hateful Content from Social Media,” 2024.

[18] M. Saroar, M. Haque, N. Arhab, and M. Oussalah, “BanglaHateBERT: BERT for Abusive Language Detection in Bengali,” 2024.

[19] A. K. Das, A. A. Asif, A. Paul, M. Hossain, and A. Das, “Bangla Hate Speech Detection on Social Media Using Attention-Based Recurrent Neural Network,” Department of Computer Science Engineering, East West University, Dhaka, Bangladesh, 2024.

[20] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” 2024.

[21] C. Sai, R. Kumar, S. Saumya, and S. Biradar, "IIITDWD_SVC@ DravidianLangTech-2024: Breaking Language Barriers; Hate Speech Detection in Telugu-English Code-Mixed Text," in Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, Mar. 2024, pp. 119–123.

212-15-14759.docx

ORIGINALITY REPORT

8%	5%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
2	aclanthology.org Internet Source	1%
3	arxiv.org Internet Source	<1%
4	"Proceedings of International Joint Conference on Advances in Computational Intelligence", Springer Science and Business Media LLC, 2021 Publication	<1%
5	kth.diva-portal.org Internet Source	<1%
6	Submitted to United International University Student Paper	<1%
7	Submitted to Altinbas University Student Paper	<1%
8	Submitted to Liverpool John Moores University Student Paper	<1%
9	"Smart Technologies, Systems and Applications", Springer Science and Business Media LLC, 2026 Publication	<1%
10	Submitted to Daffodil International University Student Paper	<1%