”Bilingual Hate Speech Detection: A Deep Learning Approach Using Bengali and English Datasets”

By

Mosfeq Ahamed Nayim,	ID:19203103057
Sumaiya Hossain Niha,	ID:19203103065
Bayzid Simon Sarkar,	ID:19203103082
Md. Riyad Hossain,	ID:19203103102
Mst. Kanij Fatema,	ID:19203103103

Submitted in partial fulfillment of the requirements of the degree of **Bachelor of Science** in **Computer Science and Engineering**

Department of Computer Science and Engineering

Bangladesh University of Business and Technology

June 2024

Declaration

We do hereby declare that the research works presented in this thesis entitled "Bilingual Hate Speech Detection: A Deep Learning Approach Using Bengali and English Datasets" are the results of our own works. We further declare that the thesis has been compiled and written by us. No part of this thesis has been submitted elsewhere for the requirements of any degree, award or diploma, or any other purposes except for publications. The materials that are obtained from other sources are duly acknowledged in this thesis.

Mosfeq Ahamed Nayim ____________________
ID: 19203103057 Signature

Sumaiya Hossain Niha ____________________
ID: 19203103065 Signature

Bayzid Simon Sarkar ____________________
ID: 19203103082 Signature

Md. Riyad Hossain ____________________
ID: 19203103102 Signature

Mst. Kanij Fatema ____________________
ID: 19203103103 Signature

Approval

We do hereby acknowledge that the research works presented in this thesis entitled ”Bilingual Hate Speech Detection: A Deep Learning Approach Using Bengali and English Datasets” result from the original works carried out by Md. Saifur Rahman , Assistant Professor and Chairman, Department of Computer Science and Engineering,Bangladesh University of Business and Technology. We further declare that no part of this thesis has been submitted elsewhere for the requirements of any degree, award or diploma, or any other purposes except for publications. We further certify that the dissertation meets the requirements and standard for the degree of Doctor of Philosophy in Computer Science and Engineering.

Supervisor: ________________________

Mr. Md. Ashiqur Rahman

Lecturer

Department of CSE

Bangladesh University of Business and Technology

Chairman: ________________________

Md. Saifur Rahman

Assistant Professor and Chairman

Department of CSE

Bangladesh University of Business and Technology

Acknowledgement

We would like to express our heartfelt gratitude to the almighty Allah who offered upon our family and us kind care throughout this journey until the fulfilment of this research. Also, we express our sincere respect and gratitude to our supervisor Mr. Md. Ashiqur Rahman, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Business and Technology (BUBT). Without his guidance, this research work would not exist. We are grateful to him for his excellent supervision and for putting his utmost effort into developing this project. We owe him a lot for his assistance, encouragement, and guidance, which has shaped our mentality as a researcher. Finally, we are grateful to all our faculty members of the CSE department, BUBT, to make us compatible to complete this research work with the proper guidance and support throughout the last four years

Abstract

With the increasing rate of hate speech on social media, we have came with a solution to regulate or some what prevent the outcome that a person suffers due to hate speech. We have majorly used a few deep learning models to find and exploit hate speech from a sentence analyzing the context of that sentence. The main challenge was to find and pre-process the dataset. Two datasets bengali and english has been collected by us from kaggle consisting of a total of 30,000 tuples having 2 classes for english and 7 classes for bengali. In the pre-processing section we have converted all the classes into binary classification. We were left with 25568 tuples after pre-processing of which 14565 are hate speech and 11003 are non hate speech. The models that were used by us to test and train the dataset was LSTM, BERT, DistilBERT and RoBERTa. After the application of the models DistilBERT perform slightly better than any other model so its been choosen to hyper tune the parameters. The hyperparameter techniques that are used to do so are Optuna, Random Search and Grid Search. Among these techniques Grid Search has performed slightly better. Considering the complexity and time limitations we were unable to perform no more than 10 epoch for each model and 15 epoch combinations for the hyperparameter tuned ones. Lastly XAI has been applied resulting a great explanation on the performance.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

1.1 Introduction

Social media usage has become a daily necessity for all individuals and makes it possible to share and receive thoughts and opinions from all over the world quickly, as well as to share them easily. Social media platforms like Facebook, Instagram, Twitter and YouTube have become a part of everyone's life, and in the last few years hate speech has grown significantly in the comment sections of these sites. In some cases, social media can play an even more direct role, video footage from the suspect of the 2019 terror attack in Christchurch, New Zealand, was broadcast live on Facebook.

In recent years, hate speech (HS) has increased on a number of social media platforms.Hate speech is speech that attacks a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, sex, disability, sexual orientation, or gender identity. Nevertheless, concurrently with this manifestation of liberty, hate speech and derogatory language have been steadily increasing in the gap that exists between community policies that are in writing and the practical implications of hate speech, as well as the corporate social media model, are contributing factors to this issue. So far, several investigations have been conducted to identify hate speech automatically, focusing mainly on the English language; therefore, an effort is required to determine and diminish such hateful content in low-resource languages.

With more than 210 million speakers, Bengali is the seventh most spoken language, with about 100 million Bengali speakers in Bangladesh and 85 million in India. Apart from Bangladesh and India, Bengali is spoken in many countries, including the United Kingdom, the United States, and the Middle East. Also, a current trend on social media platforms is that apart from actual Bengali, people tend to write Bengali using Latin scripts(English characters) and often use English phrases in the same conversation.

The detection of hate speech in many languages is a difficult task. Languages vary in their complexities, cultural contexts, and expressions of hate speech. Thus, it is crucial to create systems that can recognize and comprehend hate speech with precision in a variety of linguistic contexts. Although numerous works have been performed in the detection of hate speeches in English, German, and other languages, very few works have been carried out in the context of Bengali language. In contrast, millions of people communicate on social media in Bengali. The few existing works that have been carried out need improvements in both accuracy and interpretability. Multilingual hate speech detection has advanced significantly in recent years thanks to research. In order to offer focused diagnostic insights into hate speech detection algorithms, functional tests have been established. Nonetheless, there are still difficulties in identifying hate speech in other languages and gaining access to datasets that encompass its diverse aspects. Automated hate speech identification is further complicated by language variations and complexity.

The development of multilingual hate speech detection models has also been explored in the context of offensive language detection and religious hate speech detection. These efforts highlight the importance of addressing hate speech in specific domains and languages to provide targeted solutions. Machine learning plays a vital role in hate speech detection. It makes it possible to create automatic systems that can recognize and identify instances of hate speech in text messages, social media posts, and online comments, among other types of communication. These systems can analyze vast amounts of text data and identify trends and features connected to hate speech by using machine learning algorithms.Trained machine learning algorithms receive labeled datasets containing instances of both non- hateful and hateful speech. By identifying patterns and characteristics in the data, these algorithms are able to differentiate hate speech from other forms of speech. They take significant passages from the text and apply them to predict whether or not a certain text contains hate speech.

1.2 Problem Statement

As hate speech continues to be a societal problem, the need for automatic hate speech detection systems becomes more apparent. Multilingual hate speech detection faces linguistic challenges such as code-mixing (the combination of multiple languages in a single sentence), grammatical variations, and poorly written text. These challenges can affect the performance of machine learning models, as they may struggle to handle the linguistic complexities of different languages. Detecting hate speech is a challenging task. However, first, there are disagreements on how hate speech should be defined. This means that some content can be considered hate speech to some and not to others, based on their respective definitions. We start by covering competing definitions, focusing on the different aspects that contribute to hate speech. Our aim is simply to illustrate variances highlighting difficulties that arise from such. Competing definitions provide challenges for evaluation of hate speech detection systems; existing datasets differ in their definition of hate speech, leading to datasets that are not only from different sources, but also capture different information. This can make it difficult to directly access which aspects of hate speech to identify.

1.3 Motivations

The spread of hate speech is a persistent problem in today's Internet environment. Its impact on online conversation, personal well-being, and societal cohesion is impossible to overstate. As a result, several important reasons came together that led to the need to investigate hate speech detection in both Bengali and English.Hate speech affects people all over the world and is not limited by language. It is essential to identify and stop its spread in widely used languages such as English and regionally important languages such as Bengali in order to promote an inclusive online environment.The importance of controlling hate speech is heightened by the global importance of these languages. The common language is English, but Bengali is more common in Bangladesh and some areas of India.

1.4 Flow of the Research

Dissecting the research process into discrete parts is necessary in order to create a research flow diagram for multilingual hate speech identification. This is a generic flow diagram that you can modify to fit the details of your own research project.

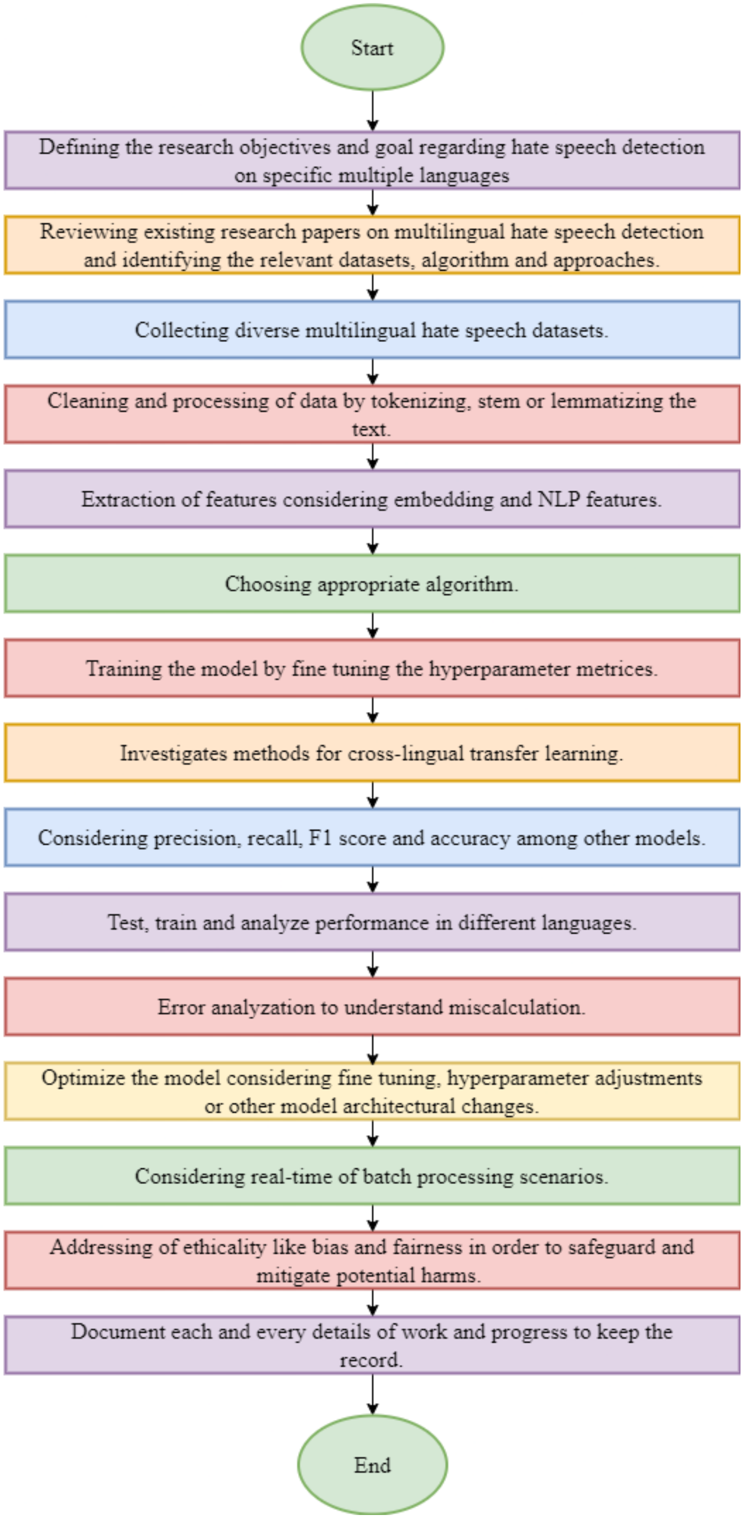

Start
Defining the research objectives and goal regarding hate speech detection on specific multiple languages
Reviewing existing research papers on multilingual hate speech detection and identifying the relevant datasets, algorithm and approaches.
Collecting diverse multilingual hate speech datasets.
Cleaning and processing of data by tokenizing, stem or lemmatizing the text.
Extraction of features considering embedding and NLP features.
Choosing appropriate algorithm.
Training the model by fine tuning the hyperparameter metrices.
Investigates methods for cross-lingual transfer learning.
Considering precision, recall, F1 score and accuracy among other models.
Test, train and analyze performance in different languages.
Error analyzation to understand miscalculation.
Optimize the model considering fine tuning, hyperparameter adjustments or other model architectural changes.
Considering real-time of batch processing scenarios.
Addressing of ethicality like bias and fairness in order to safeguard and mitigate potential harms.
Document each and every details of work and progress to keep the record.
End

Figure 1.1: The figure illustrates the flow of the thesis work.

1.5 Significance of the Research

The significance of multilingual hate speech detection research lies in its ability to highlight the opportunities and limitations of tackling hate speech in linguistically diverse environments. The following are some main ideas emphasizing the importance of this kind of research.

1.5.1 Global Nature of Hate Speech

The problem of hate speech is widespread and cuts across linguistic and national barriers. Hate speech can travel quickly across linguistic and cultural boundaries thanks to the growth of social media and online platforms. The development of instruments that can successfully counter hate speech globally depends on multilingual hate speech detection research.

1.5.2 Cultural Sensitivity

Hate speech can manifest differently across languages and cultures. Understanding the nuances of hate speech in different linguistic contexts is essential for developing accurate detection models. Research in this area contributes to the development of more culturally sensitive algorithms that can identify hate speech expressions specific to a given language or cultural context.

1.5.3 Bias and Fairness

Cross-linguistic and cross-cultural hate speech can take many forms. To create reliable detection models, it is imperative to comprehend the subtleties of hate speech in various linguistic situations. More culturally aware algorithms that can recognize hate speech expressions unique to a particular language or cultural environment are developed as a result of research in this field.

1.5.4 Resource-Poor Languages

Languages with limited resources may be underrepresented in hate speech detection models since many of the models currently in use were trained using data from major languages. By investigating strategies for creating efficient hate speech detection models for languages with little training data, multilingual research contributes to redressing the imbalance.

1.5.5 Cross-Lingual Transfer Learning

Research in multilingual hate speech detection contributes to the development of transfer learning techniques. Models trained on data from one language can be adapted to perform well on other languages, even with limited labeled data. This is particularly important for languages with fewer available resources for training machine learning models.

1.5.6 Legal and Ethical Implications

In many places, hate speech has moral and legal implications. In order to develop tools that adhere to local laws and ethical standards while taking into consideration the various legal frameworks that exist worldwide, multilingual hate speech detection research is essential.

1.5.7 Improved Online Safety

Detecting hate speech effectively helps to make online spaces safer. Online platforms can better protect users from harassment, discrimination, and the negative effects of discrimination by identifying and addressing hate speech in all languages.

In summary, investigations into multilingual hate speech detection are essential to tackle the worldwide and heterogeneous character of hate speech, foster cultural awareness, reduce prejudices, and improve online safety in various language contexts.

1.6 Research Contribution

The overall contribution of the research work are:

- Collected English and Bengali language dataset from Kaggle .
- Combined the English and Bengali language datasets for multilingual hate speech detection purposes.
- Tokenization for data pre-processing will be used.
- The BERT model in both English and Bengali language will be used.

1.7 Thesis Organization

In order to organize the work for the thesis, follow these steps. The thesis work is organized as follows. Chapter 2 explores the background and literature review of Multilingual Hate Speech Detection, establishing the foundation for understanding the complexities and challenges in the field. Chapter 3 introduces the proposed architecture for Multilingual Hate Speech Detection, providing a comprehensive walkthrough of the system's procedures and methodologies. Chapter 4 includes the details of the tests and evaluations performed to evaluate our proposed architecture. Chapter 5 explains the Standards, Impacts, Ethics, Challenges, the Constraints, Timeline, and Gantt Chart. Finally, Chapter 6 contains the overall conclusion of our thesis work Write to Personal Capstone Group.

1.8 Summary

Throughout this chapter, since hate speech continues to be a social problem, we examine the problem of hate speech, as well as background. and the inspiration for our research. In addition to describing how the research is conducted, this chapter also explains how we conducted our study.

Chapter 2

Background

2.1 Introduction

In this chapter, we explore the methods used by researchers in the complex world of multilingual hate speech detection. It's our road-map through various research papers. We focus on combining insights from different sources to build a strong foundation.

2.2 Literature Review

A technique to recognise hate speech in many languages was presented by Bunny et al[1].They gathered information from 16 publicly accessible sources in nine languages that the scientific community had supplied. By using the BERT model on their dataset, they were able to obtain an impressive 83% accuracy rate. But incorrect categorization can also result from hidden context (HC), complicating variables (CF), and annotation's dilemma (AD).

A strategy for identifying hate speech in Bengali was suggested by Jobber et al.[2].They gathered information from various social media platforms in five categories: religious, political, sports, entertainment, and others. By using the BERT model on their dataset, they were able to obtain a high degree of accuracy—97%. Unfortunately, not much data has been gathered.

Tian et al.[3] proposed a method that identify the hate speech. They collected data from Twitter which have approximately 10000 hateful tweets. In their Dataset, they applied the BERT model, and they achieved a high level of accuracy of 89% in English language and 87% in Malay language. However, the collected data has limited range of hyper-parameters of the BERT models.

Ioannis et al.[4] proposed a method that identify the hate speech. They collected data from 6 categories which are Gender, Race, National origin, Disability, Religion and Sexual orientation. In their dataset, they applied the DistilBERT model, and they achieved a high level of accuracy of 80.36%.However, the collected data is not very rich.

A technique for identifying hateful and offensive speech using meta-learning was presented by Marzieh et al[5]. They compiled data from 15 publicly available sources spanning 8 languages,incorporating input from scientific community. By using the Proto-MAML model on their dataset, they were able to obtain a high degree of accuracy—63.1%.But there isn't much rich data in the obtained data.

A method for identifying and recognizing hate speech in Bengali was proposed by Nauros Romim et al[6]. With 30,000 comments overall—10,000 of which are hate speech—and 7 distinct categories. They used the Support Vector Machine (SVM) approach to analyze their dataset and were able to obtain an impressive 87.5% accuracy rate. So, there is sufficient data.

A technique for identifying and recognizing hate speech detection was presented by Sean MacAvaney et al.[7] The method presents challenges as well as answers. Data was gathered from eight different categories. They used the Multi-view SVM approach on their dataset and were able to obtain a high degree of accuracy of 82.01%.Further study is yet required in both technical and practical areas.

Using an attention-based recurrent neural network, Amit Kumar Das et al.[8] presented a technique that makes it possible to identify and recognize hate speech in Bengali on social media.7,425 Bengali comments were included in their dataset. They used recurrent neural networks (RNNs) and CNNs on their dataset, and they were able to reach 77% accuracy.However,more datasets are needed.

Mithun Das et al [9] suggested a method for identifying and detecting hate speech and abusive language in Bengali. It consists of 5,071 tweets in actual Bengali (1,341 offensive, 825 hateful) and 5,107 tweets in Roman Bengali (2,063 offensive, 510 hateful).With their dataset, they applied m-BERT, XLM-Roberta, IndicBERT, muRIL, and ELFI, and they achieved an accuracy level of 83%. However, they only looked at data that was obtained through the Twitter API and made available to the public.

To detect hate speech in Bengali, Nauros Romim et al. [10]presented a technique that allows social media comments to be identified and recognized. The following is a binary class hate speech (HS) dataset in Bengali with over 50,000 tagged comments, of which 40.17% are classified as hate speech and the other remarks as non-hate speech. By implementing BERT on their dataset, they were able to attain an accuracy level of 86.78%.They found that emotions and punctuation had no impact on HS recognition.

Ali et al.[11] had proposed a method on multilingual racial hate speech detection using transfer learning by the application of BERT and HateXplain models. They were able to gain a 88% precision rate and an F1-score of 86%. They were unable to research further due to the limitation of time. They claimed that there is room for improvement.

A method was suggested by Manuvie et al.[12] on automated sentiment and hate speech analysis of facebook data, where they implemented the XLM-T and NLP models. They were able to achieve an F1-score on Hateful speech was 71% and 83% on Non-hateful speech. They have kept the room for improvement by achieving a precision of no more than 55%.

Yadav et al.[13] had introduced a model on large annotated dataset for multi-domain and multilingual hate speech identification using the application of BERT-Zero Shot. They had achieved an accuracy of 93%. Only obstacle they faced was that the model could not detect leetspeak.

A method was introduced by Srikissoon et al.[14] on how multilingual transformers can help detect topical hate speech, by implementing the models LASER + SVM Baseline, mBERT and XLM-RoBERTA. They had achieved an accuracy of at least 68.2% and at most 85% because the model has smaller datasets with class imbalance, which resulted in producing a weak fit.

A proposed method called multi-dataset training for cross-domain hate speech detection using a system known as LMU at HaSpeeDe3 by Hangya et al.[15] helped us understand the use of NLP better. AlBERTo, UmBERTo, mBERT and XML-R models were used by them. They were able to achieve only an accuracy of approximately 64% due to the lack of datasets.

Pistori et al.[16] proposed a natural language processing (NLP) for Hate Speech Detection in Italian Social Media Text.They used a large set of English data and Italian datasets.Several deep learning and machine learning technologies were employed in their experiment. Above them XML-Large was the most accurate, with a score of 88.63 percent. They need more investigation about their model for increase the accuracy.

Nazri et al.[17]suggested a technique to detect hate speech in many languages. They carried out experiments with four different tweets corpora: InterTASS,EmoEvent,HatEval,MEX-A3T. They had used the BERT and BETO model in their method. The BETO model achieved the maximum accuracy of 86.58 percent. But there was limited focus on improving dataset quality for enhanced model learning.

Chiwamba et al.[18] proposed a NLP model for Identify Hate Speech. They were able to collect a corpus of 9352 tweets. Using the BERT system in their dataset, they achieved the maximum accuracy of 98.6 percent. Even though their accuracy was good, they did not thoroughly investigate how to optimize the BERT model for better performance.

A strategy for Emotionally Informed Hate Speech Detection was suggested by Wang et al [19] They experimented with seven available HS corpora: Davidson, Founta, Waseem, Evalita, IberEval, HatEval. By using the BERT system in their dataset, they achieved the maximum accuracy of 91.3 percent. But the research does not fully consider how the meaning of words and context affects accurately identifying hate speech.

Lagarteja et al.[20] proposed Convolutional Neural Networks(CNN) for Detect Hate

Speech. They were able to collect a corpus of 9352 tweets. By using the CNN model on their dataset, they achieved the maximum accuracy of 91 percent. Even though their accuracy was good, they might enhance accuracy by expanding the number dataset.

Mirko et al.[21] suggested a technique for identifying hate speech. They utilized two distinct datasets—PolicyCorpsXL and Religious hate—for shared tasks.The BERT model is being used within their dataset, and they are attaining excellent levels of accuracy—91 percent—but the out-of-domain setting is not functioning properly.

Ashfia et al.[22] developed a method for recognizing hate speech in Bengali. They used the dataset G-BERT and the BERT model. They attained the highest accuracy possible, 95.56%. Despite the fact that the dataset contains sentences in many languages. Certain languages have fewer records.

A method developed by Jobair et al.[23] was used to identify hate speech in the Bengali language. They created a brand-new dataset that is split into five categories: sports, politics, religion, entertainment, and other topics. They used the dataset and the BERT model. They attained the highest accuracy possible, eighty percent. Still, the data set is not particularly rich.

Juan et. al.[24] introduced a technique for identifying hate speech in Bengali. They started gathering data from the official Twitter account. For this dataset, they employed the BERT model. With 61.3% accuracy, they achieved a high level of success. Nevertheless, datasets have a lengthy context, are constrained, and are complex.

Amit et.al.[25] introduced a technique for identifying hate speech in Bengali. 7,425

Bengali comments have been compiled into a dataset. Their dataset G-BERT model approach was employed. It was successful in reaching a high accuracy level of 95.56%. However,the combination of Bengali and English language is not used.

Table 2.1: Overview of the literature review.

Reference	**Research Purpose**	**Research Methods**	**Results**	**Challenges/Research Gaps**
[1]	Multilingual Hate Speech Detection	BERT	83%	Wrong classification due to annotation's dilemma (AD), Wrong classification due to confounding factors (CF), Wrong classification due to hidden context (HC).
[2]	Bengali Hate Speech Detection	BERT	97%	Limited dataset.
[3]	Multilingual Hate Speech Detection	BERT	89% in English, 87% in malay.	Limited range of hyper-parameters of the BERT models
[4]	Multi-label hate speech detection	DistilBERT	80.36%	Shortage of collected data
[5]	Cross-Lingual Few-Shot Hate Speech and Offensive Language Detection	Proto-MAML	63.1%	Shortage of collected data

[6]	Hate Speech detection in the Bengali language: A dataset and its baseline evaluation	SVM	87.5%	Abundance of data.
[7]	Hate speech detection: Challenges and solutions	BERT, Multi-view SVM	82.01%	Need for more research on technical and practical matters.
[8]	hate speech detection on social media using neural network	Recurrent neural network (RNN), CNN	77%	The lack of sufficient data.Further,need more datasets.
[9]	Hate Speech and Offensive Language Detection in Bengali	m-BERT, XLM-Roberta, IndicBERT, muRIL, ELFI	83%	Only analyzed publicly available data crawled via Twitter API.

[10]	SOCIAL MEDIA COMMENTS FOR HATE SPEECH DETECTION IN BANGLA	BERT	86.78%	Found that emoji and punctuation do not affect HS detection
[11]	Multilingual Racial Hate Speech Detection Using Transfer Learning	BERT and HateXplain	88%,F1-score 86%	Due to the limitation of time they were unable to research further.
[12]	Automated Sentiment and Hate Speech Analysis of Facebook Data by Employing Multilingual Transformer Models.	XLM-T and NLP	F1-scoreHateful 71% Non hateful 83%	Hateful Speech detection Precision is only 55%
[13]	Large Annotated Dataset for Multi-Domain and Multilingual Hate Speech Identification.	BERT-Zero Shot	93%	Cannot detect leetspeak

[14]	Combating Hate: How Multilingual Transformers Can Help Detect Topical Hate Speech.	LASER + SVM Baseline, mBERT and XLM-RoBERTA	Lowest 68.2% and highest 85%	Smaller datasets with class imbalance produced a weak fit.
[15]	LMU at HaSpeeDe3: Multi-Dataset Training for Cross-Domain Hate Speech Detection.	AlBERTo, UmBERTo, mBERT and XML-R	Around 64%	The accuracy is not that great. More data required.
[16]	Hate Speech Detection in Italian Social Media Text	XML-Base, XLM-Large, mBERT, ITA-Base-XXL	88.63%	Limited evaluation of hate speech detection models on Italian language subset.
[17]	Multi-Task Learning Approach to Hate Speech Detection	BERT, BETO	86.58%.	Limited focus on improving dataset quality for enhanced model learning.

[18]	BERT-Mini Model for Hate-Speech Detection	BERT	98.6%	Does not thoroughly investigate how to optimize the BERT model for better performance.
[19]	Emotionally Informed Hate Speech Detection	ELMo, CNNFastText, BERT	91.3%	The research does not fully consider how the meaning of words and context affects accurately identifying hate speech.
[20]	Hate Speech Detection Using Natural Language Processing Techniques	CNN	91%	Limited dataset
[21]	Political and Religious Hate Speech Detection	BERT	91%	Out-of-domain setting does not work properly.
[22]	Hate Speech	BERT,G-BERT	95.56%	Multiple language sentences are included in the Dataset. Some languages have less data .
[23]	Bengali Hate Speech Detection	BERT	80%	Shortage of collected data

[24]	Contextual Information in Hate Speech Detection	BERT	61.3%	we can observe that the dataset has complex,limited dataset, long contexts.
[25]	Bangla hate speech	G-BERT	95.56%	Banglish sentence is not included.

2.3 Problem Analysis

In real life, the problem we face with hate speech in social media is the connection between online inflammatory speech and real-world violence. Hate speech disseminated on social media platforms has contributed to violence and discrimination against minorities. The response to this problem has been uneven, with the task of deciding what to censor falling to a handful of corporations that control the platforms.The spread of hate speech and misinformation online can have dangerous offline implications, creating an environment of intimidation and exclusion.But now Deep machine learning can help us to address the problem of hate speech in social media by enabling the development of more effective hate speech detection and moderation systems.

2.4 Summary

Social Media Problems This chapter also included the disadvantages of analyzing and reviewing the latest strategies and suggestions. We want to improve the machine learning model, introduce a new dataset, build a software application base, eliminate the threat and inconsistencies of hate speech as much as possible.

Chapter 3

Methodology for Hate Speech Detection

3.1 Introduction

In this part,we support the feasibility study of Hate speech identification with Text suggestion by evaluating in many languages. A web application for detecting hate speech that is multilingual is an effective tool that can recognize and flag damaging and offensive words on the internet in a variety of languages. To effectively identify and categorize objectionable language, the web program makes use of a sizable library of multilingual hate speech patterns and linguistic markers. It can distinguish between several types of hate speech, such as visible offensive insults, disrespectful language, vulgar jokes, and discriminating statements.

The development of multilingual hate speech detection involves various components and techniques. One approach is to use deep learning algorithms, such as Long Short-Term Memory (LSTM), and BERT to train models on labeled dataset that contain dataset of hate speech in different languages. To increase the accuracy, we used our dataset to apply this kind of various deep learning models. Next, we select the model with the highest accuracy. Researchers frequently use pre-trained word embedding or language models that capture the linguistic significance of words and phrases across several languages to construct efficient multilingual hate speech detection models. These already-trained models can be adjusted or modified for use in other languages with particular hate speech detection tasks.

Finally, this chapter focuses on the methodology for multilingual hate speech detection, emphasizing the methods and strategies used to recognize and moderate hate speech in many languages. The website we are intending to create will be able to identify hate speech in several languages from various dataset categories.

3.2 Algorithm

3.2.1 Step 1: Load Datasets

- Input: Paths to English dataset (`labeled_data.csv`)
- Input: Bangla dataset (`Bengali_hate_speech.csv`).
- Load English and Bangla datasets using Pandas.
- Map labels:
  - Replace ”non-hate speech” with 0.
  - Replace ”hate speech” with 1.
- Output: `english_df`, `bangla_df`.

3.2.2 Step 2: Translate Bangla Dataset to English

- Input: `bangla_df['sentence']` (list of Bangla sentences).
- Load the MarianMT model and tokenizer for Bangla-to-English translation.
- Define a function `translate(texts)`:
  - Tokenize Bangla sentences using the MarianMT tokenizer.
  - Use the MarianMT model to generate translations.
  - Decode translations into English sentences.
- Apply `translate()` to all Bangla sentences.
- Output: `translated_bangla_sentences`.

3.2.3 Step 3: Combine Datasets

- Replace Bangla sentences in `bangla_df` with `translated_bangla_sentences`.
- Extract sentences and label columns from both datasets (`english_df`, `bangla_df`).
- Concatenate the English and Bangla datasets into a single DataFrame.
- Output: `combined_df`.

3.2.4 Step 4: Text Preprocessing

- Input: `combined_df['sentence']` (list of sentences).
- Perform general cleaning:
  - Remove numeric values.

  - Convert to lowercase.
  - Remove HTML tags using regex.
  - Remove URLs.
  - Remove punctuation.
  - Remove emojis.
  - Remove social media chat words.
  - Correct spelling using TextBlob.
- Additional preprocessing for LSTM:
  - Remove stop words using NLTK.
  - Tokenize sentences using SpaCy's tokenizer.
  - Apply lemmatization using NLTK.
- Output: `cleaned_sentences`.

3.2.5 Step 5: Manual Validation

- Randomly sample a subset of `cleaned_sentences`.
- Validate correctness:
  - Check translation accuracy.
  - Ensure proper noise removal and preprocessing.
- Output: Verified and corrected preprocessed dataset.

3.2.6 Dataset Overview

Initially, we collected an English dataset (`labeled_data.csv`) and a Bangla dataset (`Bengali_hate_speech.csv`) from Kaggle. Both datasets were converted into a binary classification system where 0 represents non-hate speech and 1 indicates hate speech. The Bengali dataset was translated into English using the Helsinki-NLP MarianMT model. After translation, both datasets were merged into a single DataFrame. Preprocessing involved converting text to lowercase, removing noise (numeric values, URLs, HTML tags, punctuation, emojis), and correcting spelling errors. For LSTM models, additional preprocessing steps included removing stop words, tokenization, and lemmatization. Finally, manual validation ensured data quality.

Table 3.1: Dataset Category Distribution

Category	Count
Hate Speech	14565
Non-hate Speech	11005

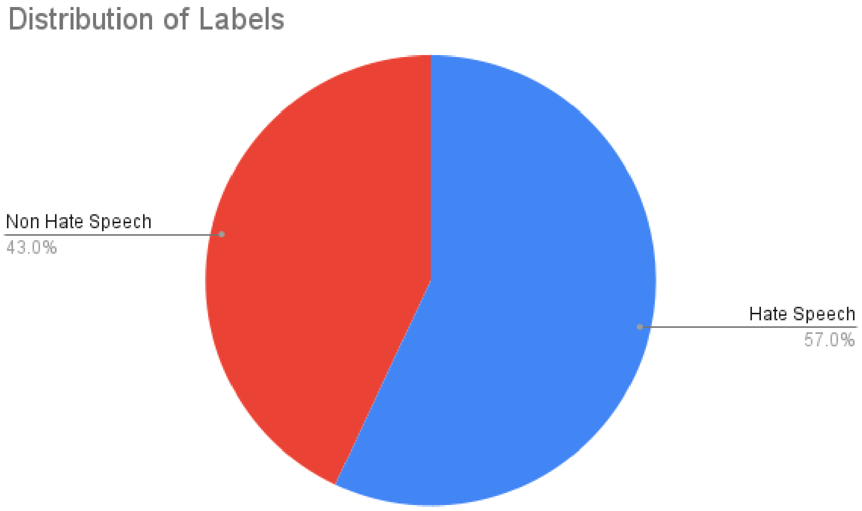
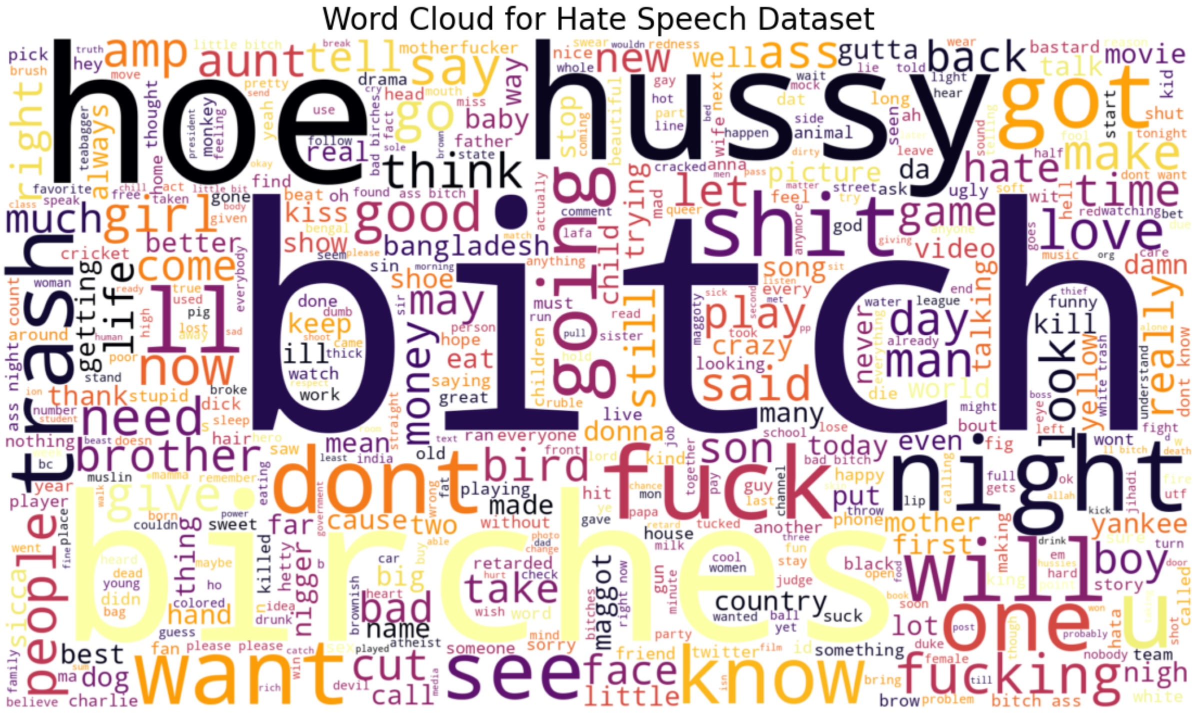


Figure 3.1: Visualization of Table 3.1

3.3 Word Cloud

A Word Cloud is a visual representation of text data where the size of each word indicates its frequency or importance in the dataset. It is widely used in text analysis to highlight key themes and trends. The process begins with text preprocessing, where common stop words, punctuation, and special characters are removed. Then, the frequency of each unique word is calculated, with more frequently occurring words being displayed in larger sizes. The visualization is generated in a cloud-like format, often with customizable fonts, colors, and layouts to enhance readability. Word clouds are commonly used in social media analysis to identify trending topics, text summarization to extract key insights from documents, market research to analyze customer feedback, and news or literature to highlight recurring themes. This simple yet powerful tool helps in quickly understanding large volumes of text in an intuitive manner.



Word Cloud for Hate Speech Dataset

Figure 3.2: Visualization of the Dataset via Word Cloud

3.4 Feasibility Analysis

The completion of this eleven-month-long study project required one researcher and one supervisor. Hardware and software support was required for the thesis research. The researchers also completed the dataset creation and assessment process that was required by the study activity. The enormous amount of data being collected for the project is done so while keeping in mind the dataset's legal feasibility. The thesis study also didn't require any funding from the supervisor or university.

3.5 Requirement Analysis

Required requirements are analyzed and listed below:

3.5.1 Computational Requirements

- Use a powerful computer for quick analysis.
- Ensure enough processing power and memory for analyzing text in real-time.

3.5.2 Data Input

- Accept text from social media and messages.
- Handle texts of different lengths and formats.

3.5.3 Training Data

- Use diverse data for training.

- Include labeled hate speech examples for each language.

3.5.4 Deep Learning Frameworks

- Use open-source frameworks.

3.5.5 Mobile Application Development

- Build the app with open-source libraries.
- Make it work on Android and iOS.
- Design a user-friendly interface.

3.5.6 Real-time Processing

- Develop fast algorithms for quick hate speech detection.

3.5.7 Accuracy and Robustness

- Create a highly accurate model.
- Continuously improve its performance.

3.5.8 Documentation and Training

- Provide clear instructions for developers and users.
- Train users on responsible use and system limitations.

3.5.9 Ethical Considerations

- Follow ethical guidelines for fair detection.
- Consider cultural biases and freedom of speech impacts.

- High-performance computer device.
- Images are fed into this gadget.
- Open-source software libraries for scientific computing.
- Open-source software libraries are utilized to implement the deep learning approach.
- Open-source software libraries are utilized to create the mobile application.

3.6 Research Methodology

We collected our required dataset from various sources in the first step. After assembling the dataset, we pre-processed data by using noise removal and tokenization as we mentioned it in Chapter 3.2.4. Then we applied different deep learning models using our dataset to improve the accuracy. On our dataset, we used LSTM and BERT models. Then we take the model which gives the best accuracy. Following that we created our web application. As part of our web-development we will be using HTML5 and CSS3 in the front-end section. In the back-end PHP.

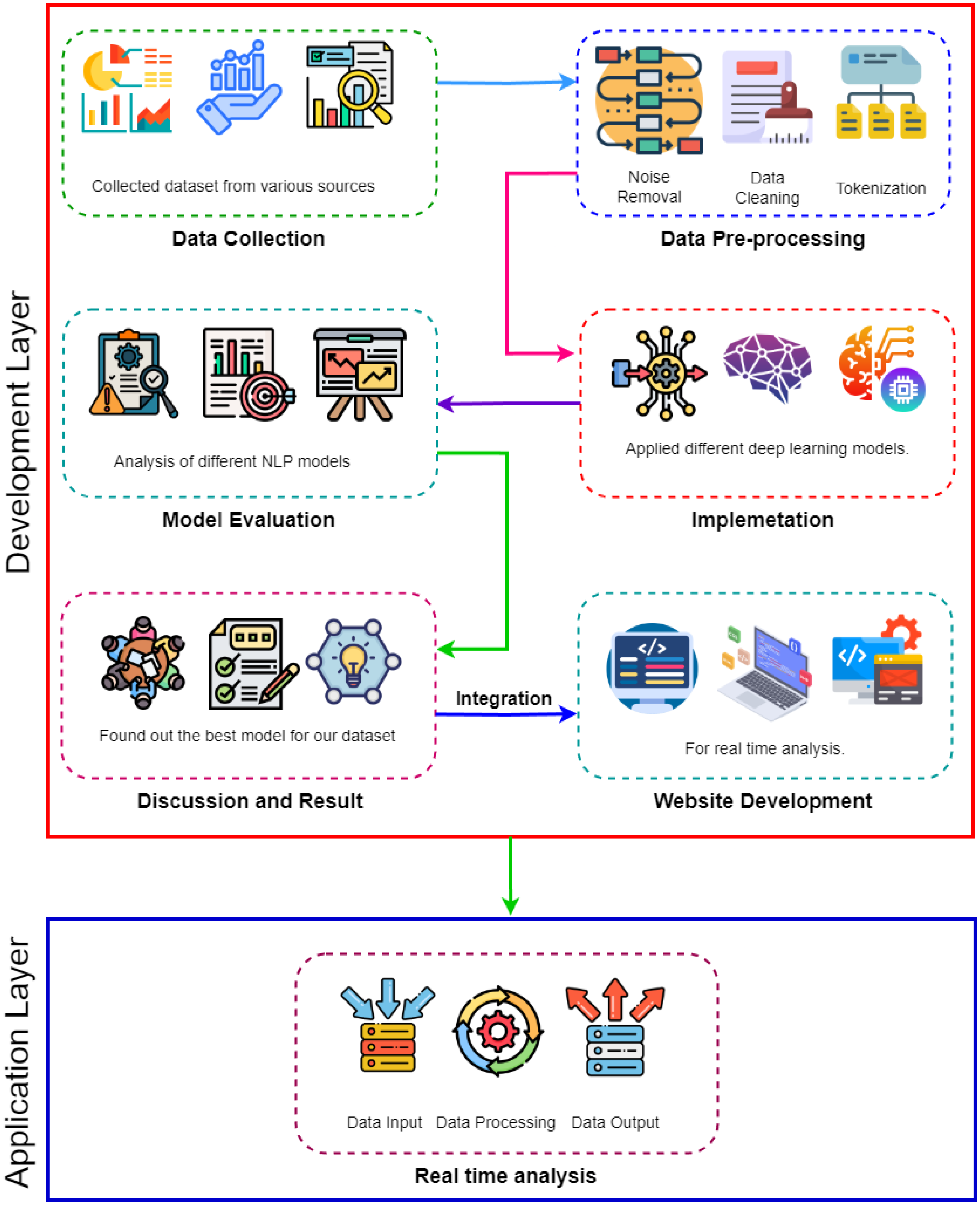
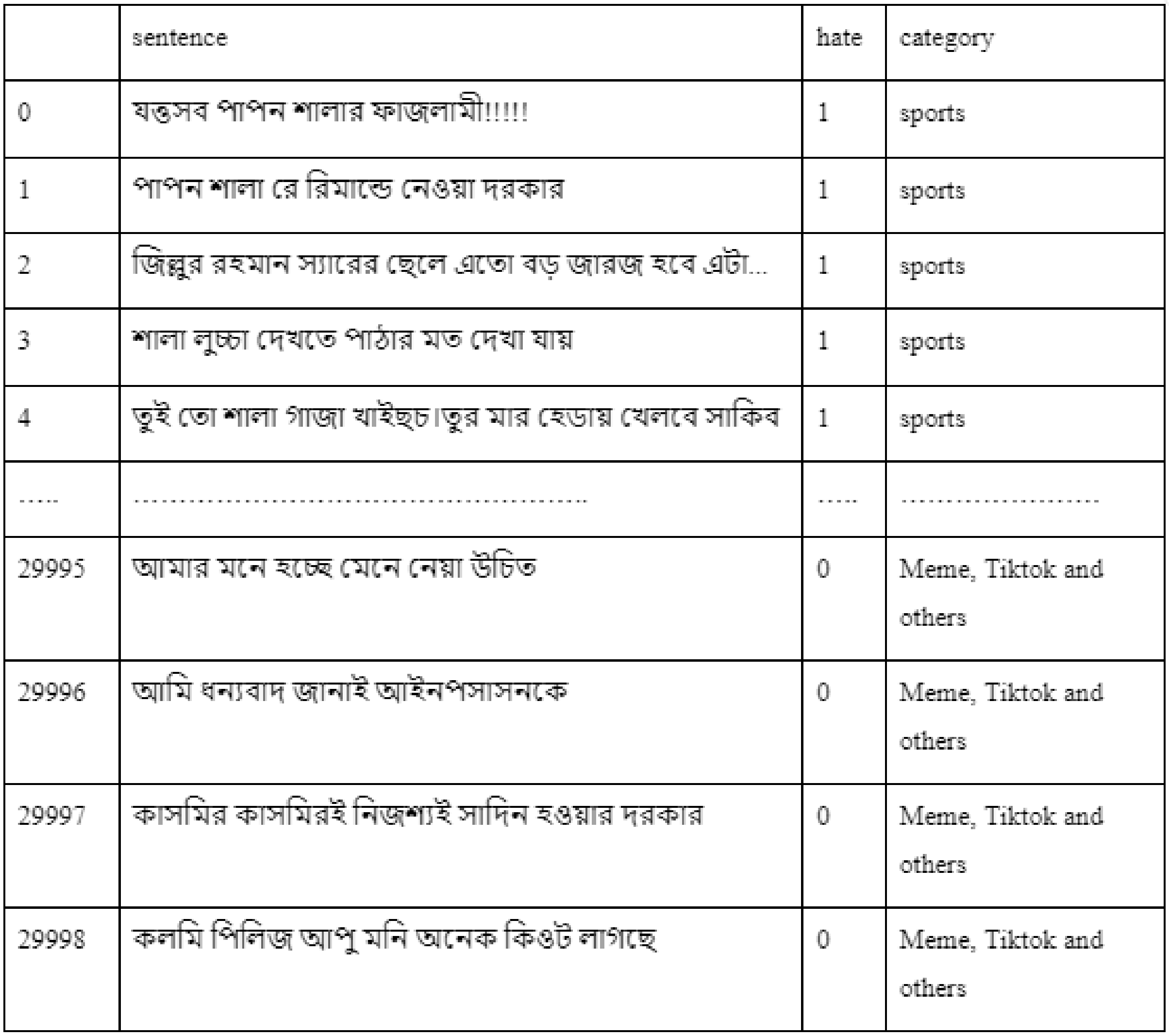
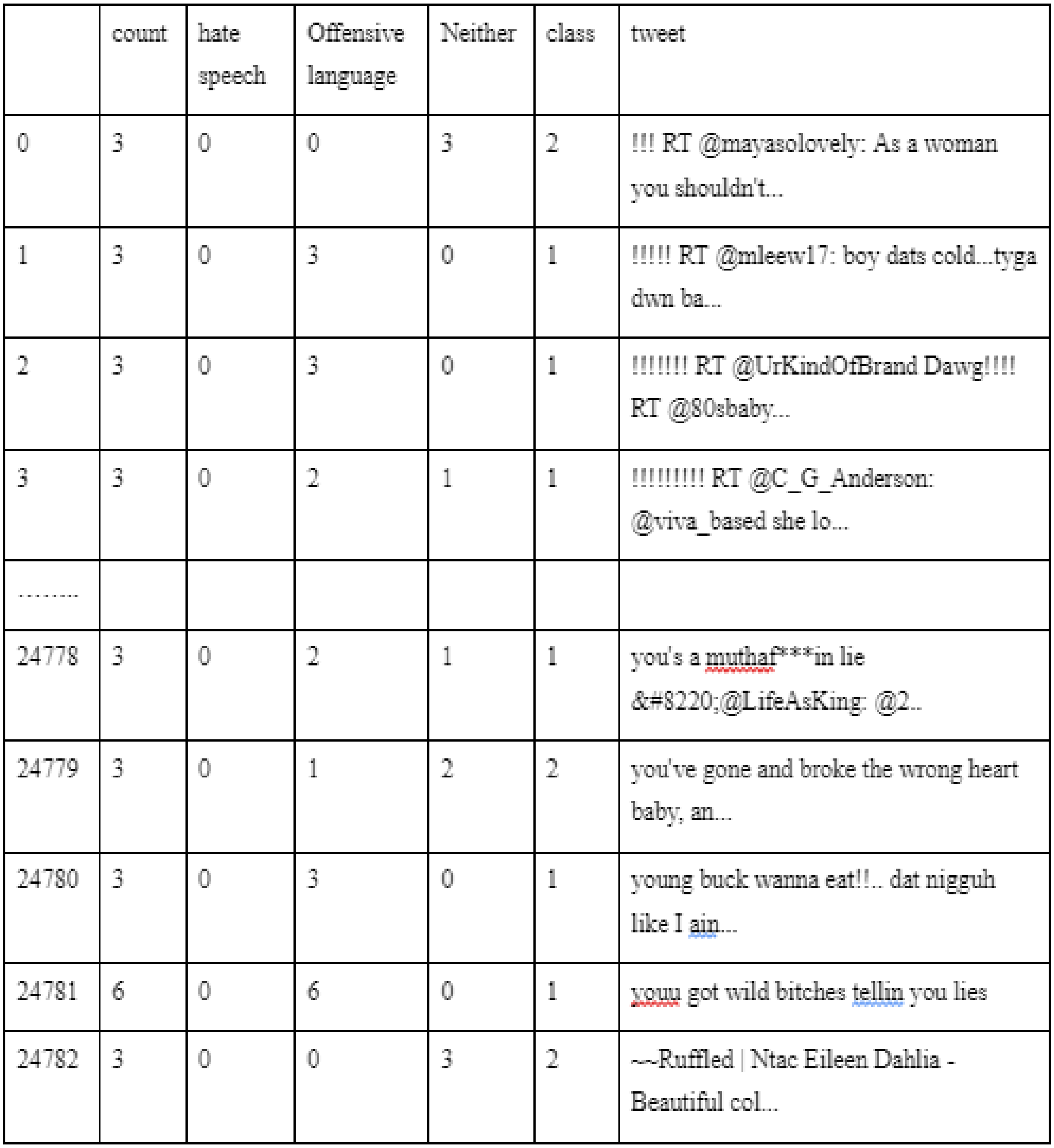


Figure 3.3: The figure illustrates the workflow of the proposed system.



	sentence	hate	category
0	যত্তসব পাপন শালার ফাজলামী!!!!!	1	sports
1	পাপন শালা রে রিমান্ডে নেওয়া দরকার	1	sports
2	জিল্লুর রহমান স্যারের ছেলে এতো বড় জারজ হবে এটা...	1	sports
3	শালা লুচ্চা দেখতে পাঠার মত দেখা যায়	1	sports
4	তুই তো শালা গাজা খাইছচ।তুর মার হেডায় খেলবে সাকিব	1	sports
…..	………………………………………………..	…..	……………………
29995	আমার মনে হচ্ছে মেনে নেয়া উঁচিত	0	Meme, Tiktok and others
29996	আমি ধন্যবাদ জানাই আইনপসাসনকে	0	Meme, Tiktok and others
29997	কাসমির কাসমিরই নিজশ্যাই সাদিন হওয়ার দরকার	0	Meme, Tiktok and others
29998	কলমি পিলিজ আপু মনি অনেক কিওট লাগছে	0	Meme, Tiktok and others

Figure 3.4: Dataset of Bengali language.



	count	hate speech	Offensive language	Neither	class	tweet
0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't...
1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3	3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
……..						
24778	3	0	2	1	1	you's a muthaf***in lie &#8220;@LifeAsKing: @2..
24779	3	0	1	2	2	you've gone and broke the wrong heart baby, an...
24780	3	0	3	0	1	young buck wanna eat!!.. dat nigguh like I ain...
24781	6	0	6	0	1	youu got wild bitches tellin you lies
24782	3	0	0	3	2	~~Ruffled | Ntac Eileen Dahlia - Beautiful col...

Figure 3.5: Dataset of English language.

3.6.1 Dataset Visualization

In the Bengali language dataset there are 30000 rows and 3 columns. First column represents the index number. In the second column, there are sentences which are collected from various sources. If the sentence indicates hate speech, the hate column is 1 otherwise 0. Finally the fourth column represents the category of hate speech. Categories are divided into 7 types of hate speech which are- crime, entertainment, religious, meme-tik tok and others, politics and celebrity.

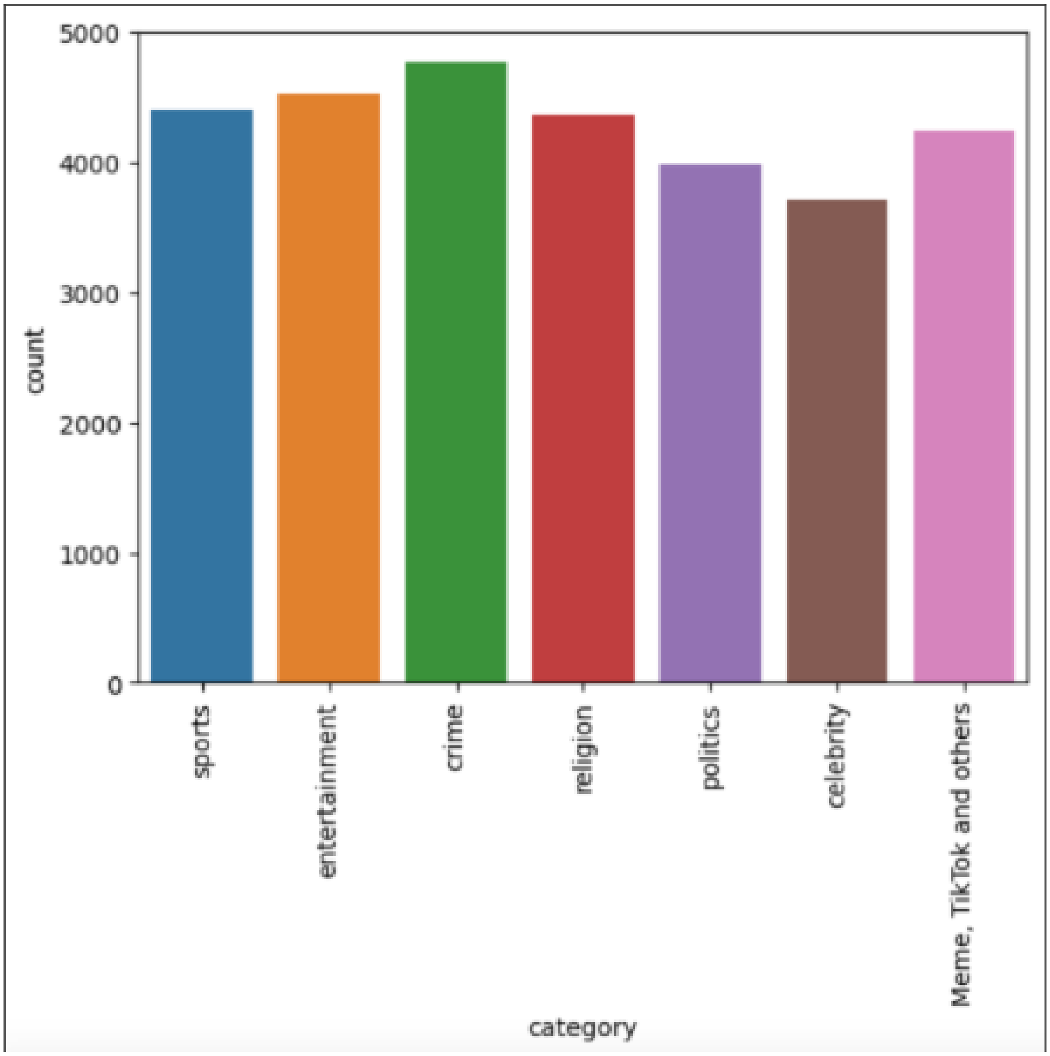


Figure 3.6: The figure illustrates different categories of hate speech

In the English language dataset there are 24783 rows and 7 columns. First column represents the count of people who judge whether a sentence contains hate speech or not. The sentences are classified into three types which are hate speech, offensive and neither. Sixth column represents what class of sentence it is. If the class is 0 then it represents hate speech. If the class is 1 then it represents offensive language and if it represents 2 then it is neither hate speech nor offensive language.

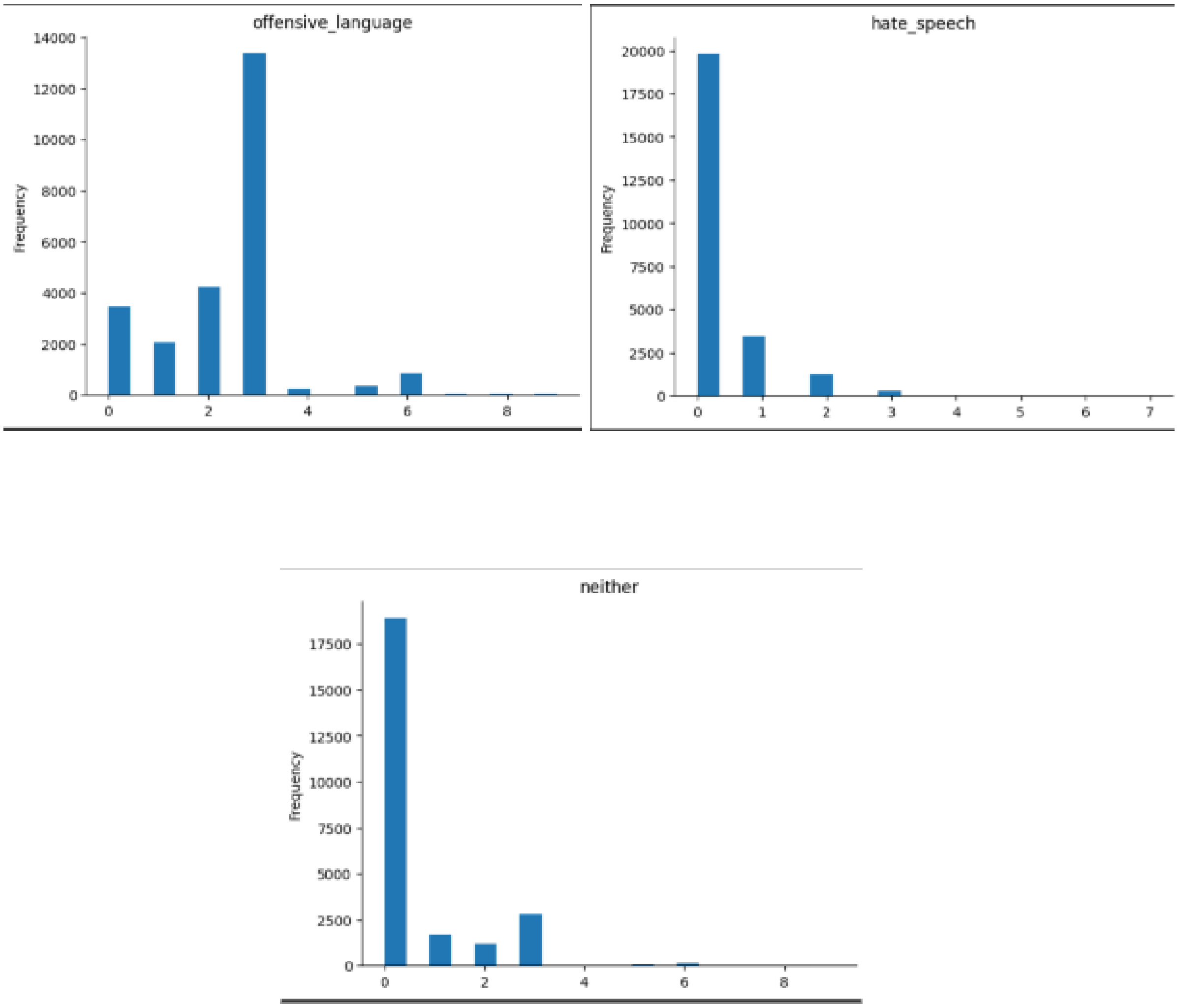


Figure 3.7: The figure illustrates different categories of hate speech

3.6.2 Data Pre-processing

The data pre-processing has occurred in three stages-

- Noise removal
- Data cleaning
- Tokenization

These two techniques are described below.

3.6.2.1 Noise removal

In this part we remove the stopwords from the text. Stop words are common words in a language that are filtered out before or after natural language processing (NLP) to improve text analysis. These words, like "the," "and," "is," and "in," don't usually carry significant meaning on their own in the context of text analysis or natural language understanding.

3.6.2.2 Data Cleaning

In this part we remove the special characters from the text and convert the data into lowercase letters. Unnecessary characters or special symbols make the model complex. So, data cleaning is an important process before tokenization.

3.6.2.3 Tokenization

Tokenization can separate words from sentences. So, this method is called word tokenization. This process is crucial because it allows the analysis of the text at a granular level, enabling the identification of specific words or phrases that might indicate hate speech. Tokenization allows algorithms to identify offensive terms easily. Each word can be assigned specific characteristics or features that aid in the classification of hate speech. Tokenization also helps to capture the context in which certain words or phrases appear.

3.7 Application Development

We have developed a web application in our app development section that can detect hate using text data from the user input. This project's work is divided into two phases: front-end development and back-end development. For the front-end of our application, we used HTML5, CSS3, Wordpress and for the back-end, we used PHP.

3.7.1 Front-end development

HTML5, CSS3 and Wordpress are widely used to develop and design web applications. The user can input text data in the application and hit scan to find out whether this is a hate speech or not. Then the text is delivered to the backend. The detection results are then provided from the backend. The application then displays the results to the user.

3.7.2 Back-end development

The server gets the detection request with a text in the backend. The server next does pre-processes as we stated in section , followed by detection using the best methods we've learned from the results and an evaluation section on the text. Once the detection result is obtained, the server identifies the hate speech. It delivers the output results to the users' end through the internet when it has completed all of the prerequisites. PHP

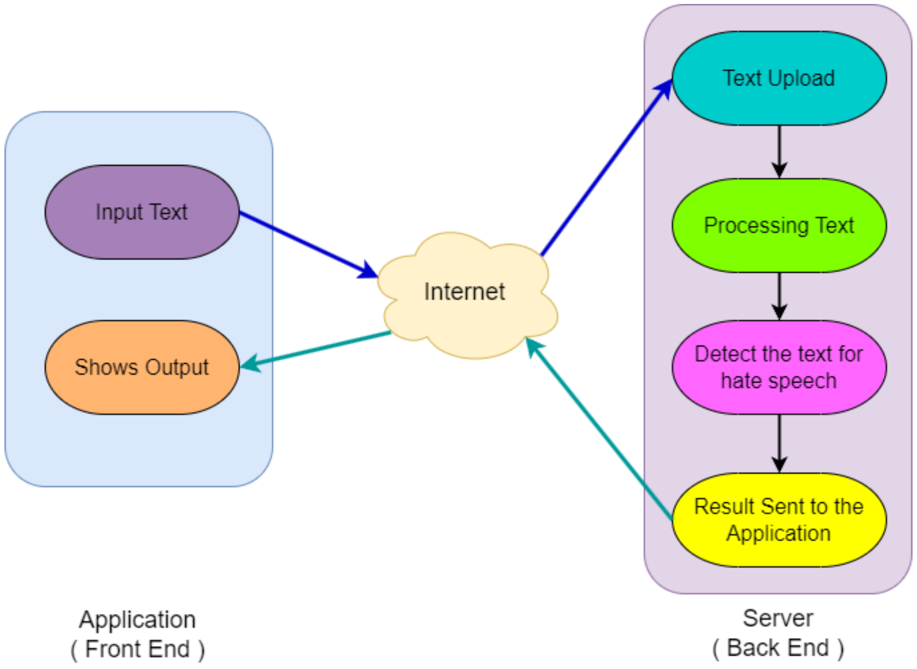

Input Text
Shows Output
Internet
Text Upload
Processing Text
Detect the text for hate speech
Result Sent to the Application
Application (Front End)
Server (Back End)

Figure 3.8: Front-end and back-end flow of mobile apps.

was used to do all of this work. PHP works as a bridge between the server and web applications.

3.8 Summary

There is a lot of hate speech on social media. The BART model is chosen among several models. So we will explain the working mechanisms of disgusting speech sentences based on the Deep Learning website application.

Chapter 4

Implementation,Testing and Result Analysis

4.1 Introduction

This section describes the architecture of the various machine learning and deep learning models used to explain the proposed natural language classification for "Hate Speech Detection" from Bengali and English data set. The method is broken down into phases that use our gathered data set to apply various machine learning and deep learning techniques.

4.2 System Setup

Pre-processing data, conducting experiments, and evaluating models are all done with the Python programming language. The architectures described are implemented using Google Colab. Installation of transformers torch datasets used together for efficient loading, preprocessing, training, fine-tuning, and evaluating transformer-based models on a variety of NLP and machine learning tasks, pandas spacy tensorflow imbalanced-learn are used for data manipulation, natural language processing, building and training machine learning models, and handling imbalanced datasets, respectively, python -m spacy download en core web sm is used to download and install spaCy's English language model (en core web sm), which is a small, general-purpose model suitable for various natural language processing tasks like tokenization, named entity recognition, and part-of-speech tagging is done. Pandas is used for efficient data manipulation and analysis, Seaborn simplifies creating statistical visualizations, and 'sklearn.metrics' provides tools for evaluating the performance of machine learning models. NumPy is also used to perform mathematical operations on the architecture.

4.3 Evaluation Matrices

A binary classification confusion matrix represents the four possible outcomes: true positive, false positive, true negative, and false negative. The actual values are represented by the columns, and the anticipated values by the rows. One of the four outcomes is represented by the intersection of the rows and columns. A typical confusion matrix is shown below:

The following metrics are derived from the confusion matrix:

- **Accuracy**: The ratio of correctly predicted objects to all objects.

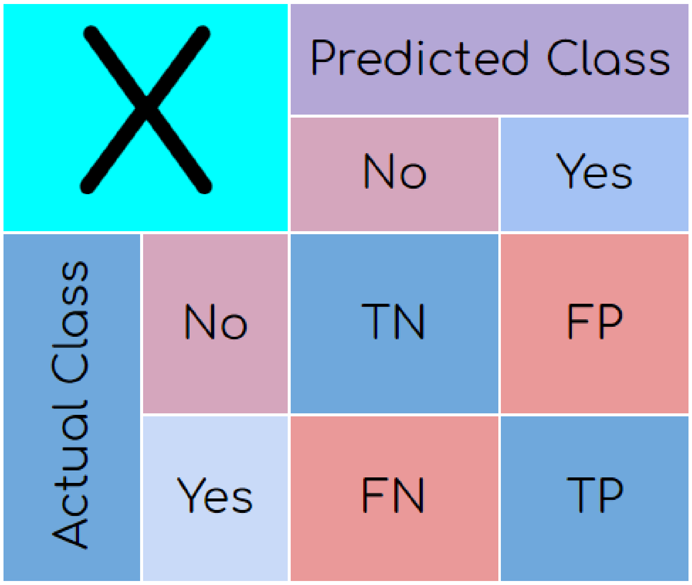
$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Precision**: How many of the picked items are relevant.

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

- **Recall**: How many relevant items are chosen.

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



		Predicted Class	
		No	Yes
Actual Class	No	TN	FP
	Yes	FN	TP

Figure 4.1: A typical confusion matrix

- **F1 Score**: A harmonic mean of precision and recall.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.4 Results and Discussion before Hyperparameter Tuning

We have collected two unbalanced dataset from kaggle which are Bengali and English for our study on Hate Speech detection with various deep learning and machine learning models. We applied several deep learning and machine learning models after balancing and pre-processing the dataset .

4.4.1 Model evaluation across different Metrices

Table 4.1 presents the results of 4 deep learning and machine learning models: LSTM, BERT, DistilBERT and Roberta. When focusing on accuracy, which measures how accurate all of the predictions are overall, ”BERT” has the highest accuracy in this comparison at ”0.91”, and ”LSTM (Long Short-Term Memory)” has the lowest accuracy at 0.82, making them the most dependable models. With ”0.90” accuracy, ”DistilBERT” and ”RoBERTa” trail closely behind ”BERT”, but they both continue to perform well on the contrary way ahead of ”LSTM”. All these models performances are recorded after balancing and transforming the dataset into appropriate format.

Table 4.1: Results of different models after pre-processing

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
LSTM	0.82	0.82	0.82	0.82
BERT	0.91	0.90	0.91	0.91
DistilBERT	0.90	0.90	0.90	0.90
RoBERTa	0.90	0.90	0.90	0.90

A graphical representation has been provided for the Table 4.1 which visualize the models accuracy, precision, recall and f1-score versus on a scale of 0 to 1.

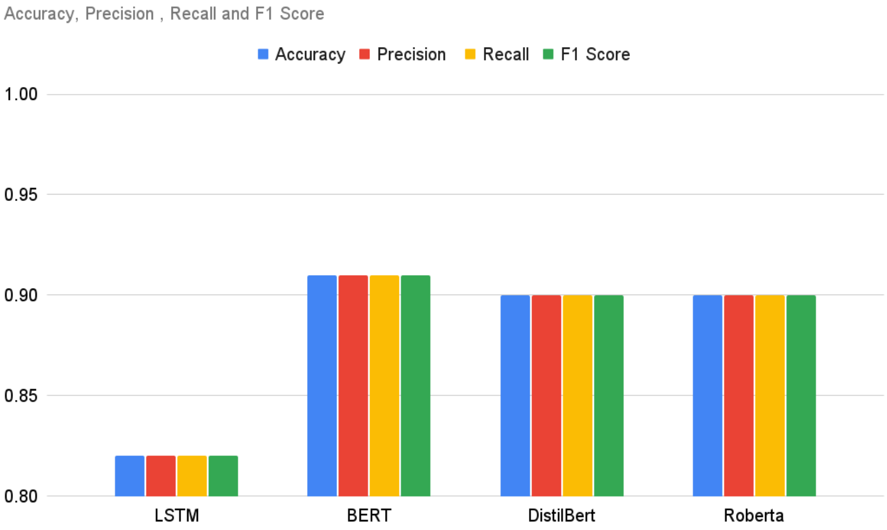


Figure 4.2: Graphical Visualization of Table 4.1

Table 4.2: Comparison Literature Review and Proposed Models

Papers	**Dataset**	**Linguility**	**Methodology**	**Accuracy**	**Catagories**
[1]	Collected	Bangla, English	BERT	83%	2
[2]	Collected	Bangla	BERT	97%	5
[3]	Collected	Eng. & Malay	BERT	89% and 87%	2
[4]	Collected	English	DistilBERT	80.36%	6
[5]	Collected	8 Languages	Proto-MAML	63.1%	8
[6]	Collected	English	SVM	87.5%	7
[7]	Collected	English	BERT, Multi-view SVM	82.01%	8
[8]	Collected	Bengali	RNN, CNN	77%	1
[9]	Collected	Bengali	mBERT, XLM-Roberta, IndicBERT, muRIL,ELFI	83%	2
[10]	Collected	Bengali	BERT	86.78%	1
[11]	Collected	English	BERT & HateXplain	88%	1
[12]	Collected	English	XLM-T & NLP	77%	1
[13]	Collected	Multilingual	BERT Zero Shot	93%	Multiple
[16]	Collected	English and Italian	mBERT, ITA-BaseXXL	88.63%	2
[22]	Collected	Bangla	BERT, GBERT	95.56%	1

[23]	Collected	Bangla	BERT	80%	5
[24]	Collected	Bangla	BERT	61.3%	1
[25]	Collected	Bangla	G-BERT	95.56%	1
Proposed Model	Self Prepared	Bengali & English	DistilBERT	90.96%	2

In Table 4.2 we have represented a comparison of different models used in different papers with different dimensions of dataset up to decimal in percentages to get a brief idea about which model performs better in terms of accuracies including the number of classes and whether they collected the dataset or created it. In order to investigate the literature review is briefly discussed in Table 2.1.

4.4.2 Training Loss, Validation Loss, Training Accuracy and Validation Accuracy Curves

Training Loss and Validation Loss are metrics that indicate how well a model is learning. Training Loss is calculated on the training data and represents the model's error in making predictions during training. Validation Loss, on the other hand, is calculated on unseen validation data and helps evaluate how well the model generalizes to new data. Training Accuracy measures the proportion of correct predictions made by the model on the training set, while Validation Accuracy assesses its predictive performance on the validation set. By analyzing the curves of these metrics over training epochs, you can detect overfitting (when validation loss increases despite decreasing training loss) or underfitting (when both losses remain high and accuracies are low).

4.4.2.1 LSTM(Long Short-Term Memory)

The two images display the training and validation performance metrics for an LSTM model:

Training and Validation Accuracy: The training accuracy curve (blue) increases consistently, confirming that the model's performance on the training set improves with each epoch. The validation accuracy curve (orange) starts high, oscillates, and gradually decreases, showing that the model's generalization worsens due to overfitting.

Training and Validation Loss: The training loss curve (blue) decreases steadily, indicating that the model is learning and improving its predictions on the training data over epochs. The validation loss curve (orange) initially decreases but starts to increase after a few epochs, showing that the model begins to overfit the training data. The increasing validation loss implies poorer generalization to unseen data.

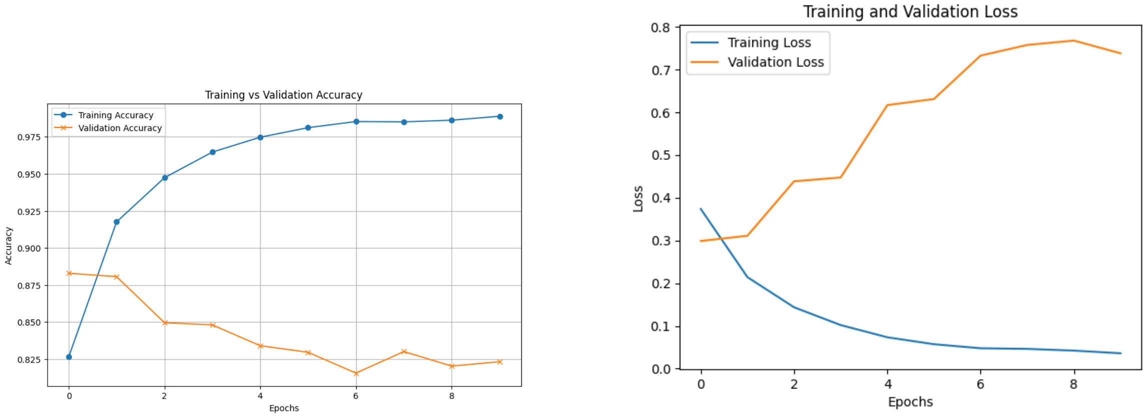
In summary, these plots highlight that the LSTM is overfitting after a certain number of epochs, as evidenced by the diverging validation loss and accuracy trends. Early stopping or regularization techniques could improve generalization.

4.4.2.2 BERT

The images represent metrics related to training and validation of a BERT model. Here's an explanation of each:

Training Loss per Epoch: The first graph displays the training loss over epochs. Loss is a measure of the model's error on the training data. The loss decreases steadily as training progresses, indicating that the model is learning and improving. The loss becomes more stable at later epochs, suggesting convergence.

Validation Loss per Epoch: The second graph represents the validation loss over a



(a) Training Vs. Validation Accuracy

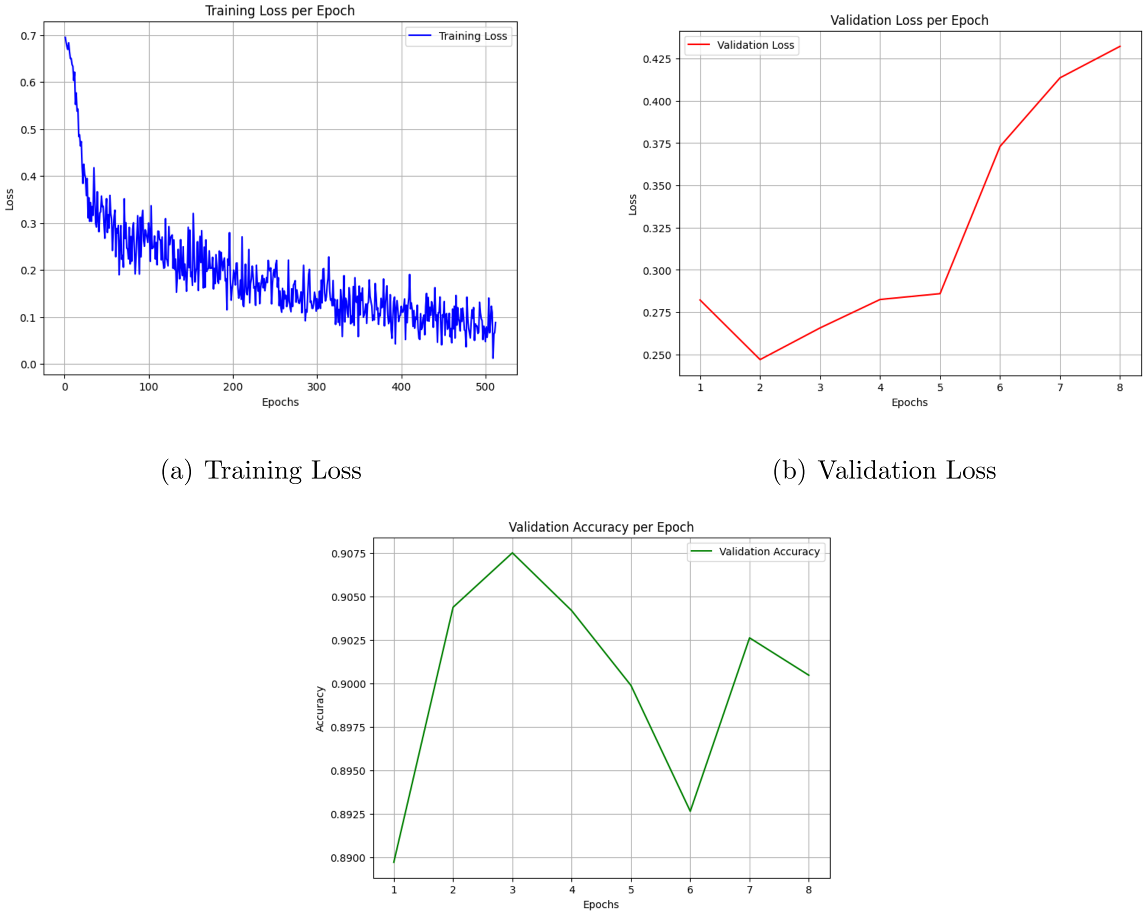
(b) Training Vs. Validation Loss

Figure 4.3: Training Vs. Validation Accuracy and Loss Visualization for LSTM

smaller number of epochs. Validation loss measures the model's error on unseen data (validation set). Initially, the validation loss is low and improves slightly (e.g., around epoch 2). After a few epochs, the loss increases significantly, which suggests overfitting, the model is performing well on the training data but not generalizing to unseen data.

Validation Accuracy per Epoch: The third graph illustrates the model's accuracy on the validation set over epochs. Accuracy measures the percentage of correct predictions. Validation accuracy increases in the initial epochs (up to around epoch 3). After epoch 3, it starts fluctuating and drops at later epochs, further indicating overfitting.

Overfitting, the divergence between training loss (which continues to decrease) and validation loss (which increases after a few epochs) suggests overfitting. Optimal Epochs, the best balance of training and validation performance might occur around epoch 2 or 3, as the validation loss is at its lowest and validation accuracy is highest.



(a) Training Loss

(b) Validation Loss

(c) Validation Accuracy

Figure 4.4: Visualization of Training and Validation Loss, and Validation Accuracy of BERT

4.4.2.3 DistilBERT

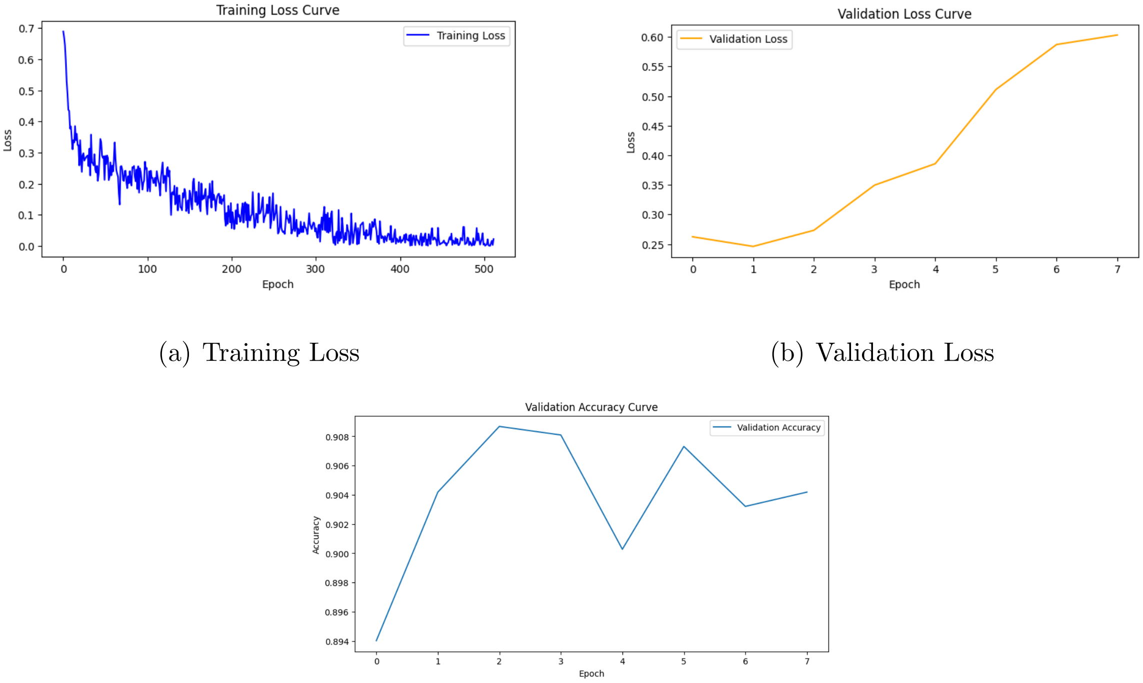
These plots represent the training and evaluation metrics for a DistilBERT model. The explanation of each of these:

Validation Loss Curve: The validation loss increases steadily as the training progresses. This indicates that the model is overfitting; it is performing well on the training data but poorly on the validation set.

Training Loss Curve: The training loss decreases steadily and approaches zero over 500 epochs, showing that the model is learning the patterns in the training data. The model is fitting very well to the training data, which is typically good, but when paired with the increasing validation loss, it points to overfitting.

Validation Accuracy Curve: Validation accuracy initially increases, peaks at around epoch 2 or 3, and then fluctuates or decreases slightly in subsequent epochs. This trend is another indication of overfitting, the model's ability to generalize is highest early in training, but as training continues, performance on unseen data worsens.

The training dynamics suggest that while the model learns effectively from the training data, it struggles to generalize to the validation data, resulting in overfitting.



(a) Training Loss

(b) Validation Loss

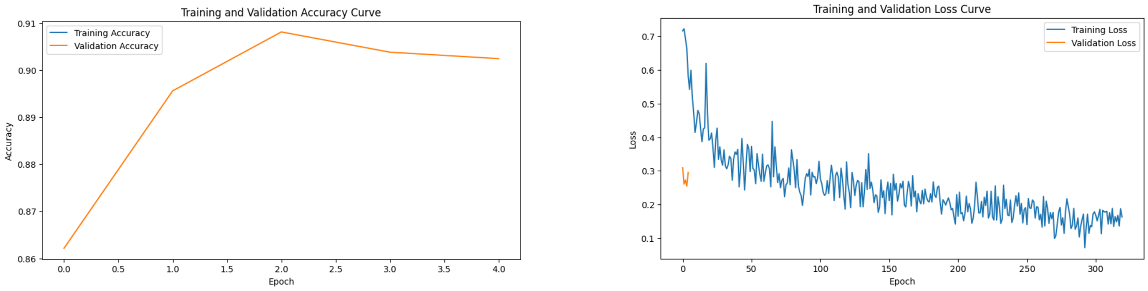
(c) Validation Accuracy

Figure 4.5: Visualization of Training and Validation Loss, and Validation Accuracy of DistilBERT

4.4.2.4 RoBERTa

The graph shows the training and validation accuracy of a machine learning model over multiple epochs. Initially, both accuracies increase, indicating successful learning. However, the training accuracy continues to rise while the validation accuracy plateaus or declines. This suggests overfitting, where the model performs well on the training data but poorly on unseen data. To address this, techniques like regularization or early stopping can be used to prevent the model from memorizing noise and improve its generalization ability.

The graph illustrates the training and validation loss of a machine learning model during its training process. Initially, both training and validation loss decrease, indicating successful learning. However, as training progresses, the training loss continues to decrease while the validation loss starts to plateau or even increase. This gap suggests overfitting, where the model performs well on the training data but poorly on unseen data. To address this, techniques like regularization or early stopping can be implemented to prevent the model from memorizing noise and improve its generalization ability.



(a) Training Vs. Validation Accuracy

(b) Training Vs. Validation Loss

Figure 4.6: Training Vs. Validation Accuracy and Loss Visualization for RoBERTa

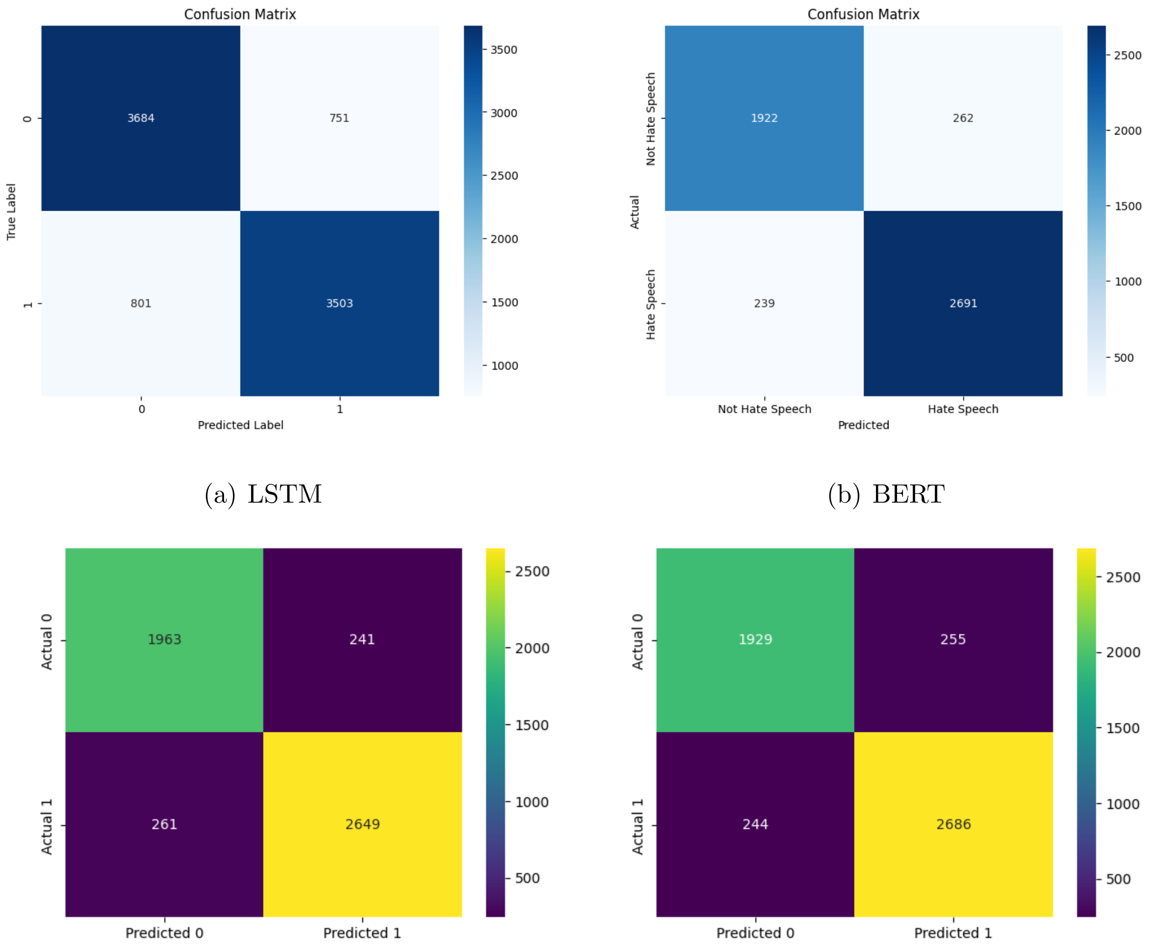
4.4.3 Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a binary classification model by comparing predicted and actual labels. It consists of four components: True Positives (TP), where the model correctly predicts the positive class; True Negatives (TN), where it correctly predicts the negative class; False Positives (FP), where it incorrectly predicts positive for a negative instance (Type I error); and False Negatives (FN), where it misses a positive instance (Type II error). Key metrics derived from the confusion matrix include accuracy, which measures overall correctness; precision, the proportion of correct positive predictions; recall (sensitivity), the proportion of actual positives correctly identified; specificity, the ability to identify negatives; and the F1 score, which balances precision and recall. This matrix provides valuable insights into the model's performance and the trade-offs between different types of errors.

4.4.3.1 Explanation of Each Confusion Matrix

For the LSTM model, it correctly predicted 3684 instances of non-hate speech (True Negatives) and 3503 instances of hate speech (True Positives). However, it misclassified 751 non-hate speech as hate speech (False Positives) and 801 hate speech instances as non-hate speech (False Negatives). This indicates that LSTM struggles with both overpredicting hate speech and missing actual hate speech, making it less reliable.

BERT, on the other hand, showed better performance. It correctly predicted 1922 non-hate speech and 2691 hate speech instances. The number of misclassifications was significantly lower, with only 262 non-hate speech instances classified as hate speech and 239 hate speech instances classified as non-hate speech. This balanced performance reflects BERT's strong ability to distinguish between the two classes.



(a) LSTM

(b) BERT

(c) DistilBERT

(d) RoBERTa

Figure 4.7: Visualization of the Confusion Matrix's

DistilBERT, a lightweight version of BERT, performed comparably to its larger counterpart. It correctly identified 1963 non-hate speech and 2649 hate speech instances, with only 241 false positives and 261 false negatives. While its results are close to BERT's, DistilBERT had slightly more false negatives, indicating it misses a few more hate speech instances compared to BERT.

RoBERTa demonstrated the best performance among all models. It accurately predicted 1929 non-hate speech and 2686 hate speech instances. The number of false positives (255) and false negatives (244) were the lowest among the models, making it highly effective at identifying both hate speech and non-hate speech with minimal errors.

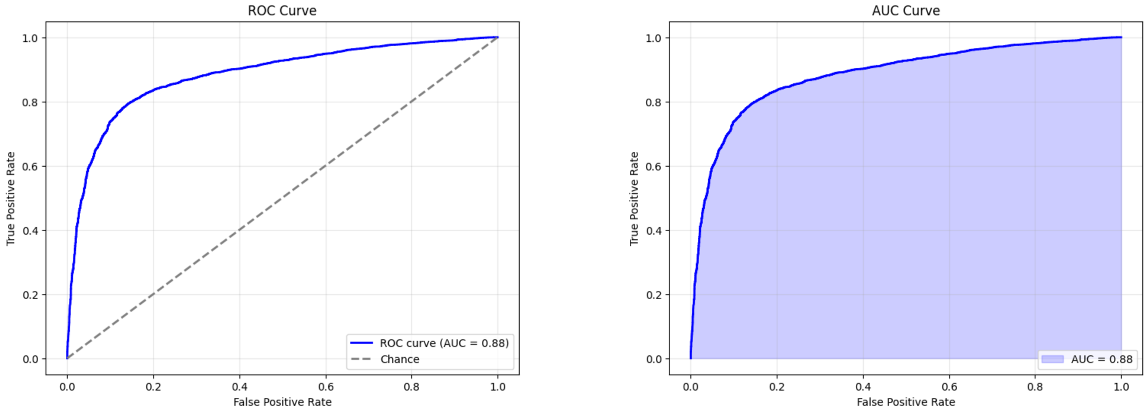
Among the models, RoBERTa is the best-performing, with the lowest false negatives and false positives, highlighting its superior ability to identify hate speech and non-hate speech with precision. BERT follows closely, with slightly higher misclassification rates but still demonstrating balanced and reliable performance. DistilBERT, though slightly less accurate, serves as a good lightweight alternative to BERT for applications requiring faster inference. LSTM, however, is the worst performer, with significantly higher misclassification rates, making it less suitable for hate speech detection tasks compared to transformer-based models. RoBERTa's superior performance can be attributed to its robust pretraining and ability to understand nuanced language effectively.

4.4.4 ROC and AUC Curve

The ROC curve (Receiver Operating Characteristic curve) compares the True Positive Rate (TPR) and the False Positive Rate (FPR) at various classification thresholds. Figure presents the Receiver Operating Characteristic (ROC) curves for four machine learning models that are utilized for binary class classification. These models includes LSTM(Long Short-Term Memory), BERT(Bidirectional Encoder Representations from Transformers), DistilBERT(Distilled Bidirectional Encoder Representations from Transformers) derived from DynaBERT(Dynamic Bidirectional Encoder Representations from Transformers), and RoBERTa(Robustly Optimized Bidirectional Encoder Representations from Transformers Approach).

An important metric in these plots is the AUC (Area Under the Curve), which shows how well the models perform overall in differentiating between true positives and false positives across all classes. Greater model performance is indicated by an AUC score that is closer to 1, which means the model can distinguish between classes more effectively with fewer false positives. The overall AUC for "BERT", "DistilBERT", and ”RoBERTa” is nearly 1, although it is 0.96. For ”LSTM” consistently achieves AUC scores between 0.88 demonstrating a drop in performance. The best performing algorithms are ”BERT”, ”DistilBERT”, and ”RoBERTa” which achieve an AUC of 0.96 in average, indicating perfect class distinction.

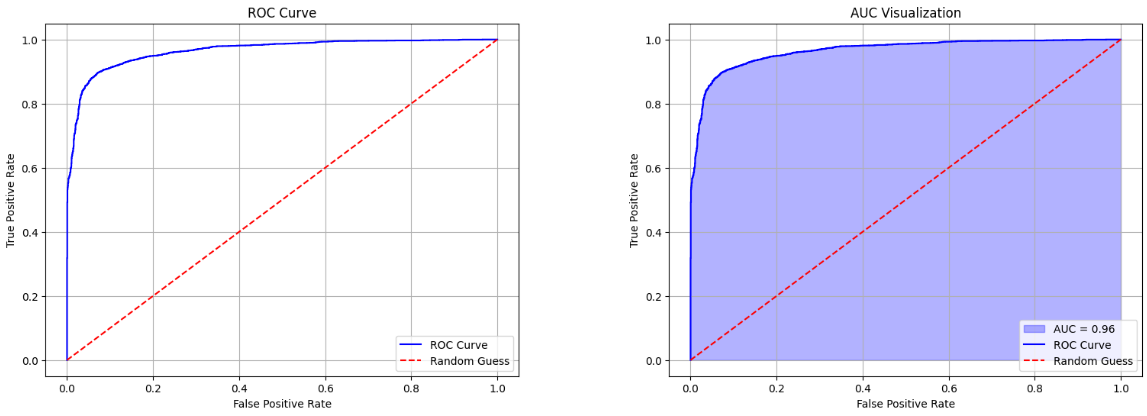
”LSTM” has slightly lower AUC scores, ranging from 0.86 to 0.89. They vary slightly in, ”BERT”, ”DistilBERT”, and ”RoBERTa” models demonstrating almost perfect classification abilities, with AUCs typically ranging from 0.96 to 1.00. ”DistilBERT” beat the other models when assessing overall AUC performance across all models.



(a) ROC Curve of LSTM

(b) AUC Curve of LSTM

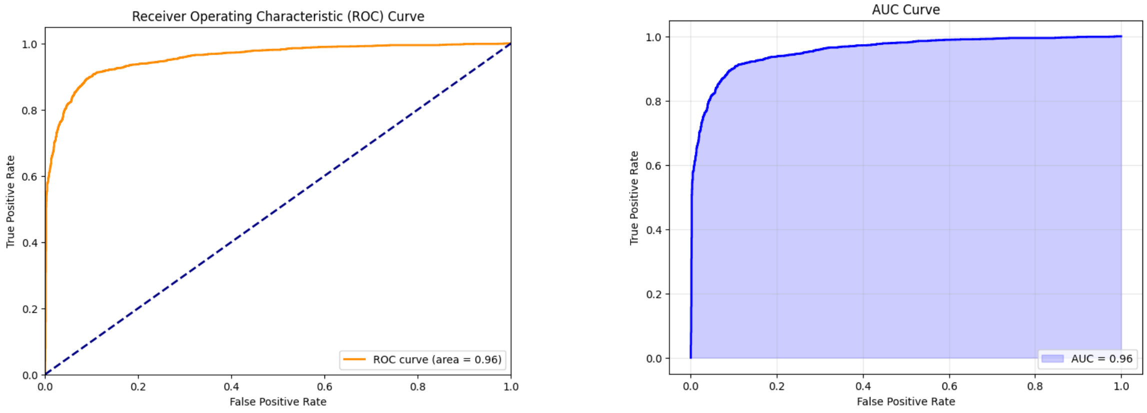
Figure 4.8: Visualized representation of ROC and AUC of LSTM



(a) ROC Curve of BERT

(b) AUC Curve of BERT

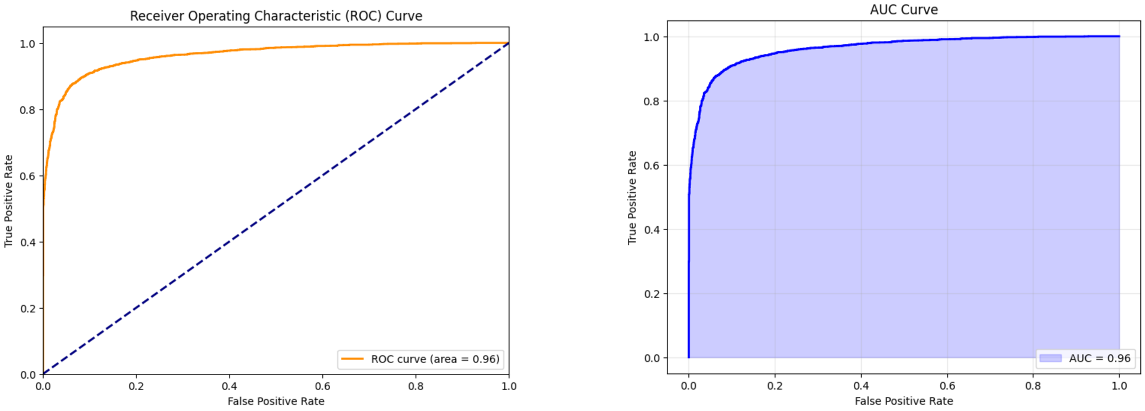
Figure 4.9: Visualized representation of ROC and AUC of BERT



(a) ROC Curve of DistilBERT

(b) AUC Curve of DistilBERT

Figure 4.10: Visualized representation of ROC and AUC of DistilBERT



(a) ROC Curve of RoBERTa

(b) AUC Curve of RoBERTa

Figure 4.11: Visualized representation of ROC and AUC of RoBERTa

4.5 Results and Discussion after Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the best configuration of hyperparameter. Predefined model parameters that manage the learning process to maximize the performance of a machine learning model. The learning rate, number of estimators, batch size and tree depth are examples of hyperparameters that affect the model's training process. In order to optimize model accuracy, precision, recall, and other metrics, tuning determines the best values for these parameters.

4.5.1 Model evaluation across different Metrices

Table 4.3 Three tuning techniques are seen in the DistilBERT Hyper Parameter Tuning Result Table 4.3: "Optuna, Randon Search and Grid Search. The optimal values for the three hyperparameters: "n-estimator" (number of combinations), "learningrate" (a measure of the model's adjustment per iteration), and batch size are derived from each method. Excellent recall, precision, and F1-scores of about 0.90 are obtained by all methods, indicating perfect classification. Yet, "Optuna," which has 6395 optimization steps and a moderate learning rate of 0.000006238, yields the accuracy at "90.06%" With twice the more optimization steps and a higher learning rate, "Random Search" attains "90.14%" accuracy, while "Grid Search" yields the highest accuracy of "90.75%". The best tuning method among the three is "Grid Search," so far, which is the most efficient and strikes the best balance between accuracy and computational resources despite having almost similar precision and recall results.

Table 4.3: DistilBERT Hyper Parameter Tuning Results as Best Parameters

Tuning Technique	**Best Parameters**		
	Optimization Steps	**Learning-rate**	**Batch Size**
Optuna	6395	0.000006238	16
Random Search	12785	0.00000643	8
Grid Search	3200	0.000005	32

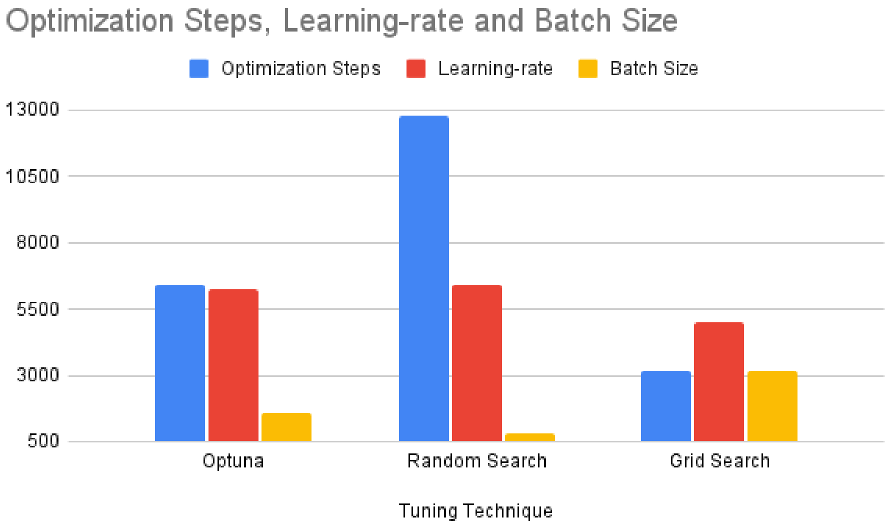


Figure 4.12: Graphical Visualization of Table 4.3

A graphical representation has been provided for the Table 4.3 which visualize the models accuracy, precision, recall and f1-score versus on a scale of 0 to 1.

Table 4.4: Comparison between the accuracies, precision,recall and f1-score of the Models

Methodology	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Optuna	90.06%	91.28%	91.72%	90.85%
Random Search	90.14%	91.31%	91.5%	91.4%
Grid Search	90.75%	93.27%	90.37%	91.8%

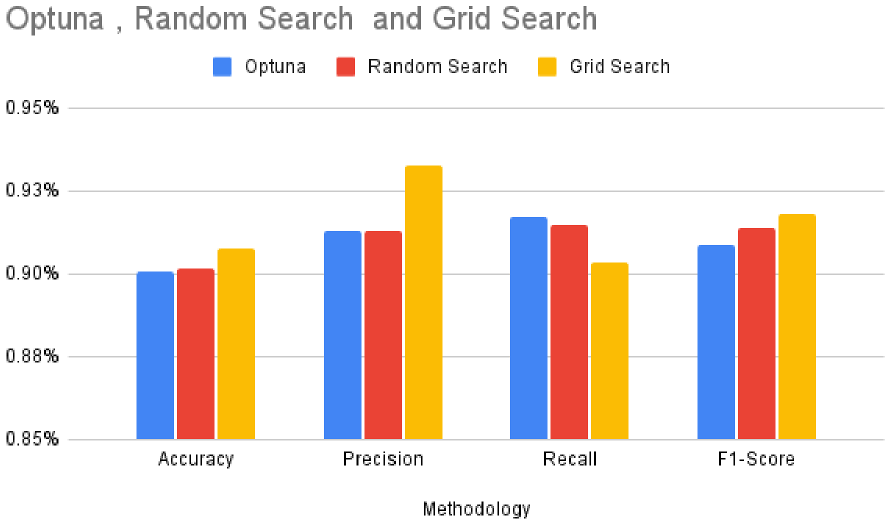


Figure 4.13: Graphical Visualization of Table 4.4

A graphical representation has been provided for the Table 4.4 which visualize the models accuracy, precision, recall and f1-score versus on a scale of 0 to 100%.

4.5.2 Training Loss, Validation Loss, Training Accuracy and Validation Accuracy Curves

Training Loss and Validation Loss are metrics that indicate how well a model is learning. Training Loss is calculated on the training data and represents the model's error in making predictions during training. Validation Loss, on the other hand, is calculated on unseen validation data and helps evaluate how well the model generalizes to new data. Training Accuracy measures the proportion of correct predictions made by the model on the training set, while Validation Accuracy assesses its predictive performance on the validation set. By analyzing the curves of these metrics over training epochs, you can detect overfitting (when validation loss increases despite decreasing training loss) or underfitting (when both losses remain high and accuracies are low).

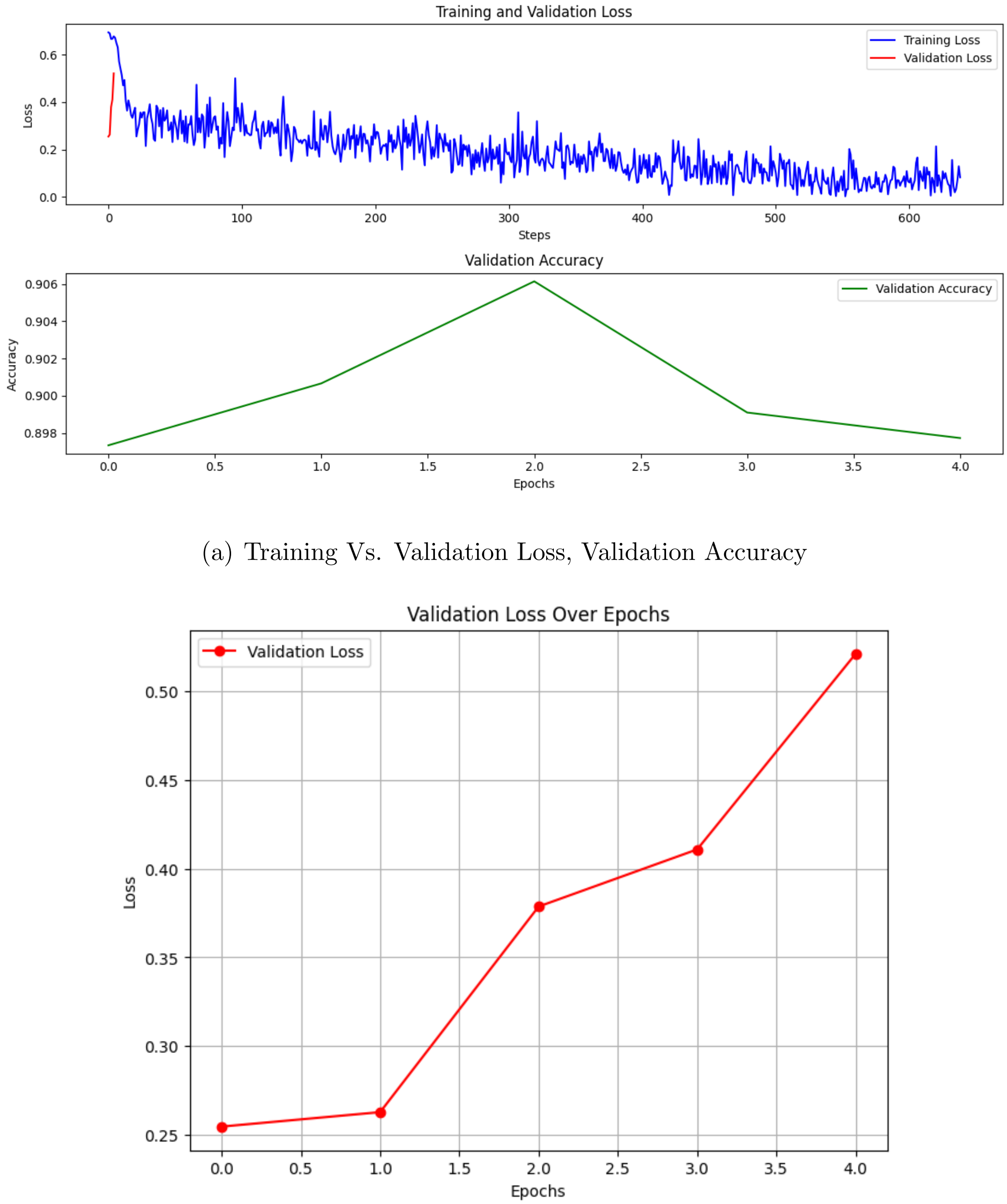
4.5.2.1 Optuna

The two images display the training and validation performance metrics for an DistilBERT model:

Training and Validation Loss: The training loss curve (blue) decreases steadily, indicating that the model is learning and improving its predictions on the training data over epochs. The validation loss curve (red) increases after a few epochs, showing that the model begins to overfit the training data. The increasing validation loss implies poorer generalization to unseen data.

Validation Accuracy: The validation accuracy curve (green) starts high, oscillates, and gradually decreases, showing that the model's generalization worsens due to overfitting.

The graph illustrates the validation loss over multiple training epochs. Ideally, validation loss should decrease as the model learns. However, this graph shows an increasing validation loss, indicating that the model is overfitting to the training data. This means the model is memorizing the training examples instead of learning generalizable patterns, leading to poor performance on new, unseen data.



(a) Training Vs. Validation Loss, Validation Accuracy

(b) Validation Loss over Epochs

Figure 4.14: Training Vs. Validation Loss, Validation Accuracy and Validation Loss over Epochs for DistilDERT(Optuna)

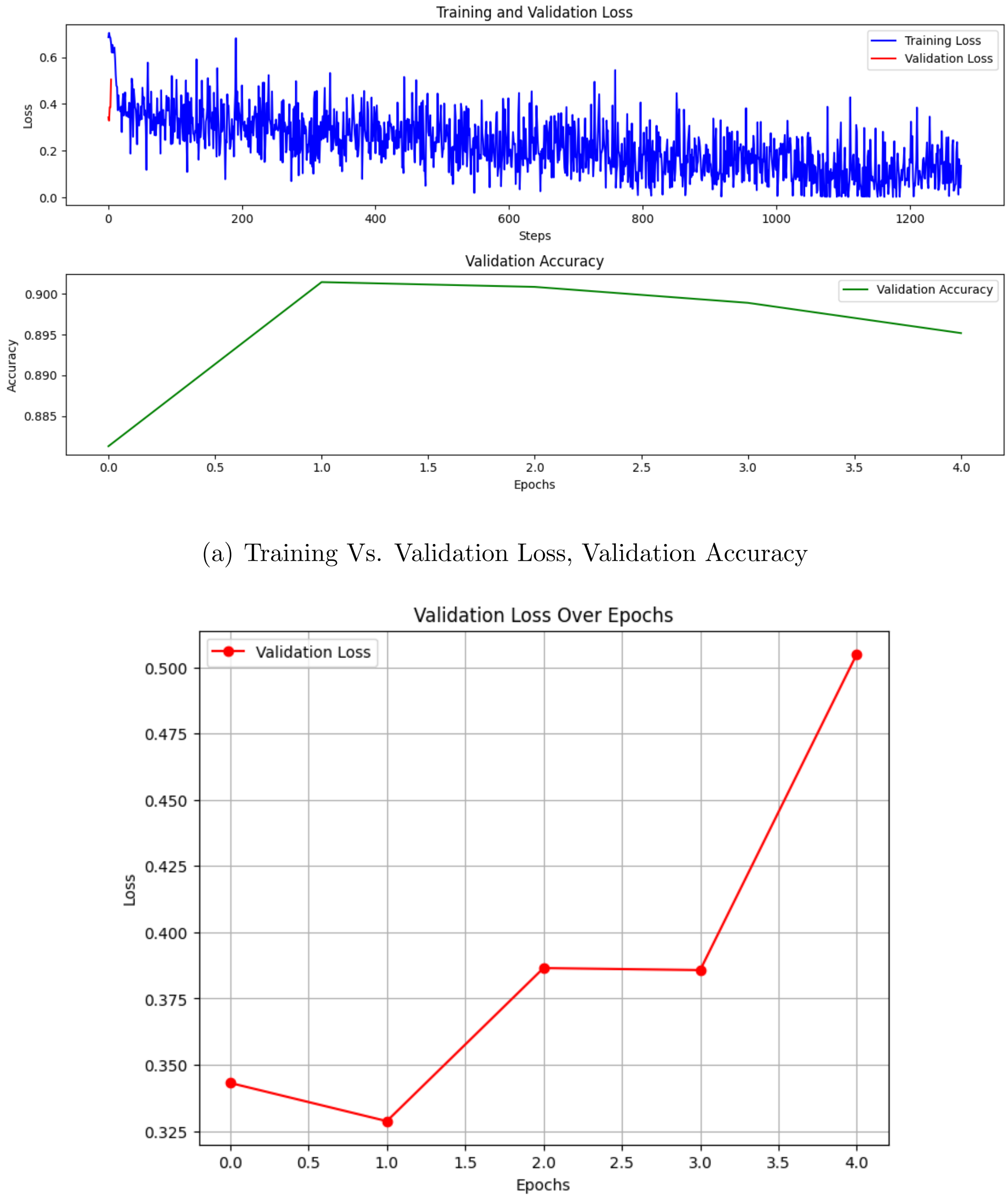
4.5.2.2 Random Search

The two images display the training and validation performance metrics for an DistilBERT model:

Training and Validation Loss: The training loss curve (blue) decreases steadily, indicating that the model is learning and improving its predictions on the training data over epochs. The validation loss curve (red) increases after a few epochs, showing that the model begins to overfit the training data. The increasing validation loss implies poorer generalization to unseen data.

Validation Accuracy: The validation accuracy curve (green) starts high, oscillates, and gradually decreases, showing that the model's generalization worsens due to overfitting.

The graph illustrates the validation loss over multiple training epochs. Ideally, validation loss should decrease as the model learns. However, this graph shows an increasing validation loss, indicating that the model is overfitting to the training data. This means the model is memorizing the training examples instead of learning generalizable patterns, leading to poor performance on new, unseen data.



(a) Training Vs. Validation Loss, Validation Accuracy

(b) Validation Loss over Epochs

Figure 4.15: Visualization of Training and Validation Loss, and Validation Accuracy of DistilBERT(Random Search)

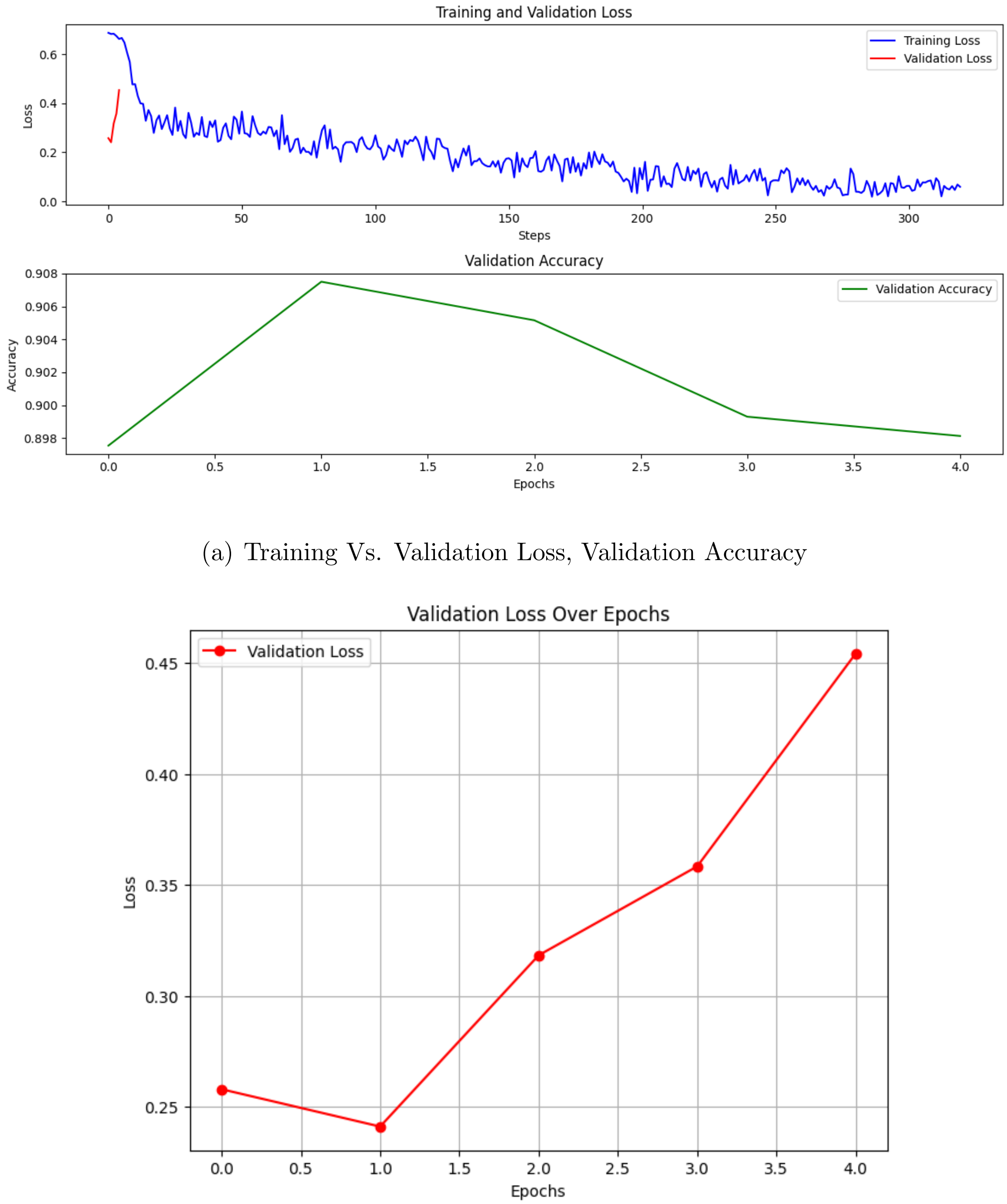
4.5.2.3 Grid Search

The two images display the training and validation performance metrics for an DistilBERT model:

Training and Validation Loss: The training loss curve (blue) decreases steadily, indicating that the model is learning and improving its predictions on the training data over epochs. The validation loss curve (red) increases after a few epochs, showing that the model begins to overfit the training data. The increasing validation loss implies poorer generalization to unseen data.

Validation Accuracy: The validation accuracy curve (green) starts high, oscillates, and gradually decreases, showing that the model's generalization worsens due to overfitting.

The graph illustrates the validation loss over multiple training epochs. Ideally, validation loss should decrease as the model learns. However, this graph shows an increasing validation loss, indicating that the model is overfitting to the training data. This means the model is memorizing the training examples instead of learning generalizable patterns, leading to poor performance on new, unseen data.



(a) Training Vs. Validation Loss, Validation Accuracy

(b) Validation Loss over Epochs

Figure 4.16: Visualization of Training and Validation Loss, and Validation Accuracy of DistilBERT(Grid Search)

4.5.3 Confusion Matrix

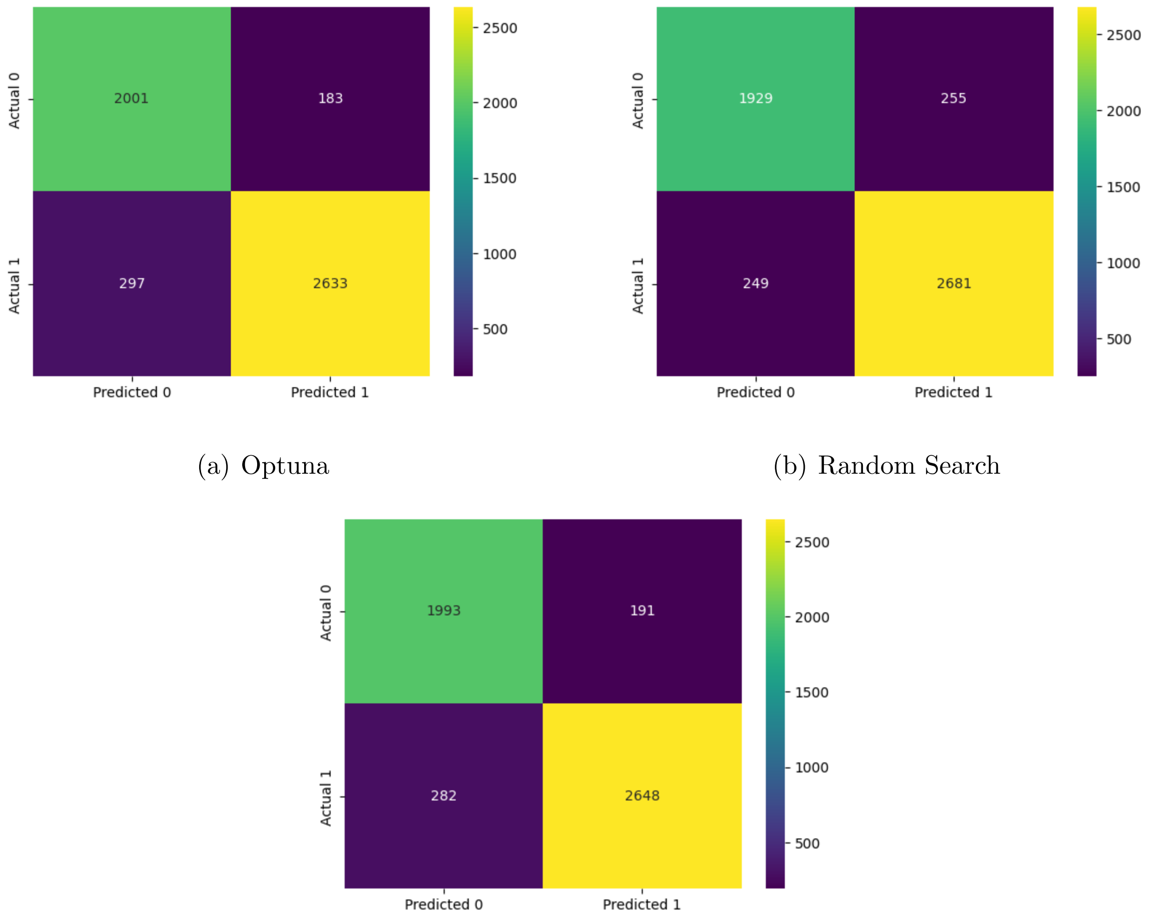
A confusion matrix is a table used to evaluate the performance of a binary classification model by comparing predicted and actual labels. It consists of four components: True Positives (TP), where the model correctly predicts the positive class; True Negatives (TN), where it correctly predicts the negative class; False Positives (FP), where it incorrectly predicts positive for a negative instance (Type I error); and False Negatives (FN), where it misses a positive instance (Type II error). Key metrics derived from the confusion matrix include accuracy, which measures overall correctness; precision, the proportion of correct positive predictions; recall (sensitivity), the proportion of actual positives correctly identified; specificity, the ability to identify negatives; and the F1 score, which balances precision and recall. This matrix provides valuable insights into the model's performance and the trade-offs between different types of errors.

4.5.3.1 Explanation of Each Confusion Matrix

For Optuna, it correctly predicted 2001 instances of non-hate speech (True Negatives) and 2633 instances of hate speech (True Positives). However, it misclassified 297 non-hate speech as hate speech (False Positives) and 183 hate speech instances as non-hate speech (False Negatives).

Random Search, on the other hand, showed, it correctly predicted 1929 non-hate speech and 2681 hate speech instances. The number of misclassifications was significantly lower, with only 249 non-hate speech instances classified as hate speech and 255 hate speech instances classified as non-hate speech. This balanced performance reflects Random Search's slightly strong ability to distinguish between the two classes.

Grid Search demonstrated that, it accurately predicted 1993 non-hate speech and 2648



(a) Optuna

(b) Random Search

(c) Grid Search

Figure 4.17: Visualization of the Confusion Matrix's of the Hyper Parameter tuned DistilBERT

hate speech instances. The number of false positives (282) and false negatives (191) were the lowest among the models, making it highly effective at identifying both hate speech and non-hate speech with minimal errors.

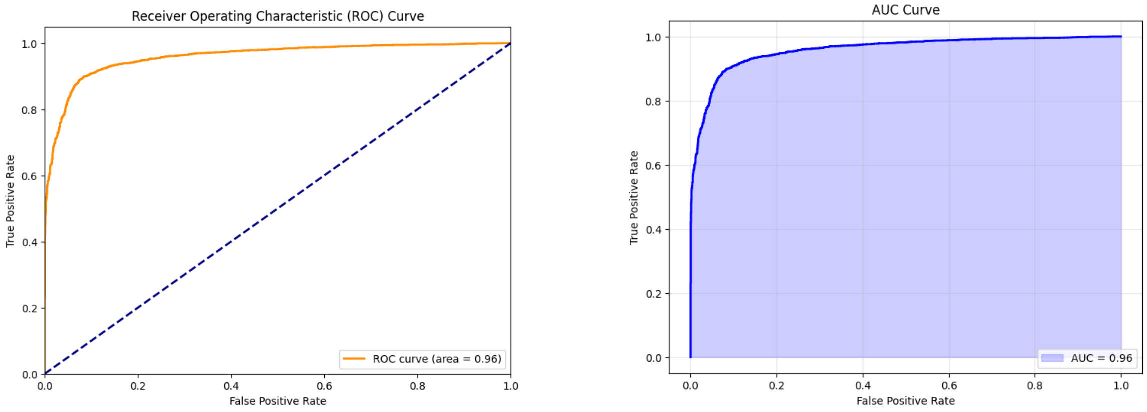
When comparing the models, Grid Search demonstrated the most balanced performance, with the lowest number of false positives (282) and false negatives (191), accurately predicting 1993 non-hate speech and 2648 hate speech instances. This makes it the most effective model at minimizing errors in both classes. Random Search, while slightly less effective, correctly classified 1929 non-hate speech and 2681 hate speech instances, with 249 false positives and 255 false negatives, reflecting a strong but less balanced ability. Optuna achieved high true positive (2633) and true negative (2001) counts but struggled with more misclassifications, yielding 297 false positives and 183 false negatives, making it less consistent compared to the other two approaches. Overall, Grid Search excelled in precision and balance, outperforming the others in minimizing classification errors.

4.5.4 ROC and AUC Curve

The ROC curve (Receiver Operating Characteristic curve) compares the True Positive Rate (TPR) and the False Positive Rate (FPR) at various classification thresholds. Figure presents the Receiver Operating Characteristic (ROC) curves for four machine learning models that are utilized for binary class classification. These models includes LSTM(Long Short-Term Memory), BERT(Bidirectional Encoder Representations from Transformers), DistilBERT(Distilled Bidirectional Encoder Representations from Transformers) derived from DynaBERT(Dynamic Bidirectional Encoder Representations from Transformers), and RoBERTa(Robustly Optimized Bidirectional Encoder Representations from Transformers Approach).

An important metric in these plots is the AUC (Area Under the Curve), which shows how well the models perform overall in differentiating between true positives and false positives across all classes. Greater model performance is indicated by an AUC score that is closer to 1, which means the model can distinguish between classes more effectively with fewer false positives. The overall AUC for ”BERT”, ”DistilBERT”, and ”RoBERTa” is nearly 1, although it is 0.96. For ”LSTM” consistently achieves AUC scores between 0.88 demonstrating a drop in performance. The best performing algorithms are ”BERT”, ”DistilBERT”, and ”RoBERTa” which achieve an AUC of 0.96 in average, indicating perfect class distinction.

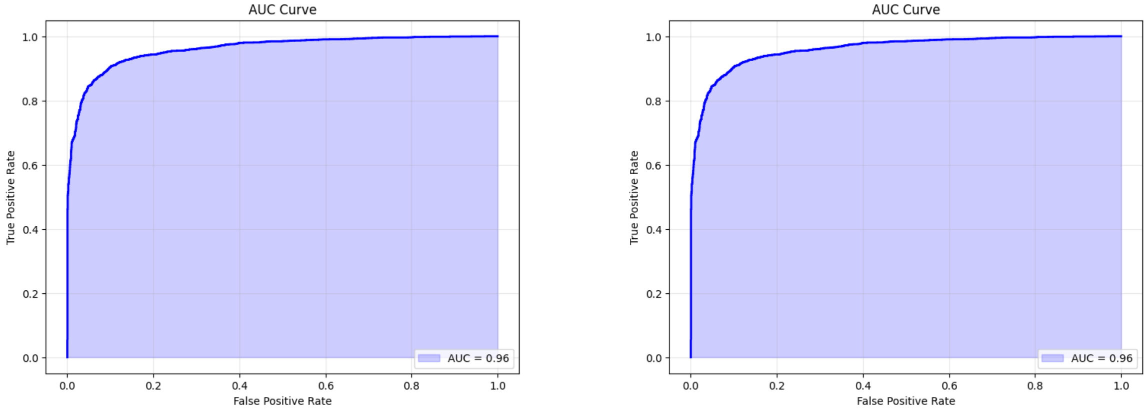
”LSTM” has slightly lower AUC scores, ranging from 0.86 to 0.89. They vary slightly in, ”BERT”, ”DistilBERT”, and ”RoBERTa” models demonstrating almost perfect classification abilities, with AUCs typically ranging from 0.96 to 1.00. ”DistilBERT” beat the other models when assessing overall AUC performance across all models.



(a) ROC Curve

(b) AUC Curve

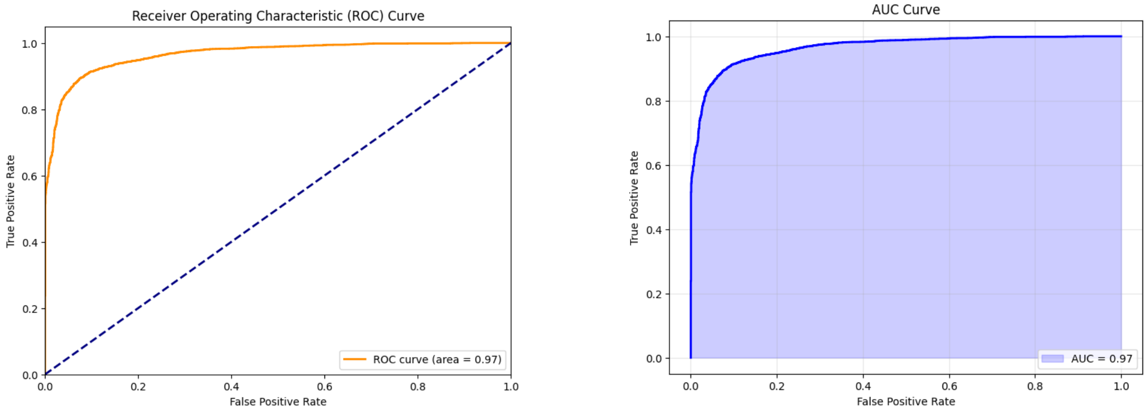
Figure 4.18: Visualized representation of ROC and AUC of DistiilBERT after appying Optuna)



(a) ROC Curve

(b) AUC Curve

Figure 4.19: Visualized representation of ROC and AUC of DistilBERT after applying Random Search



(a) ROC Curve

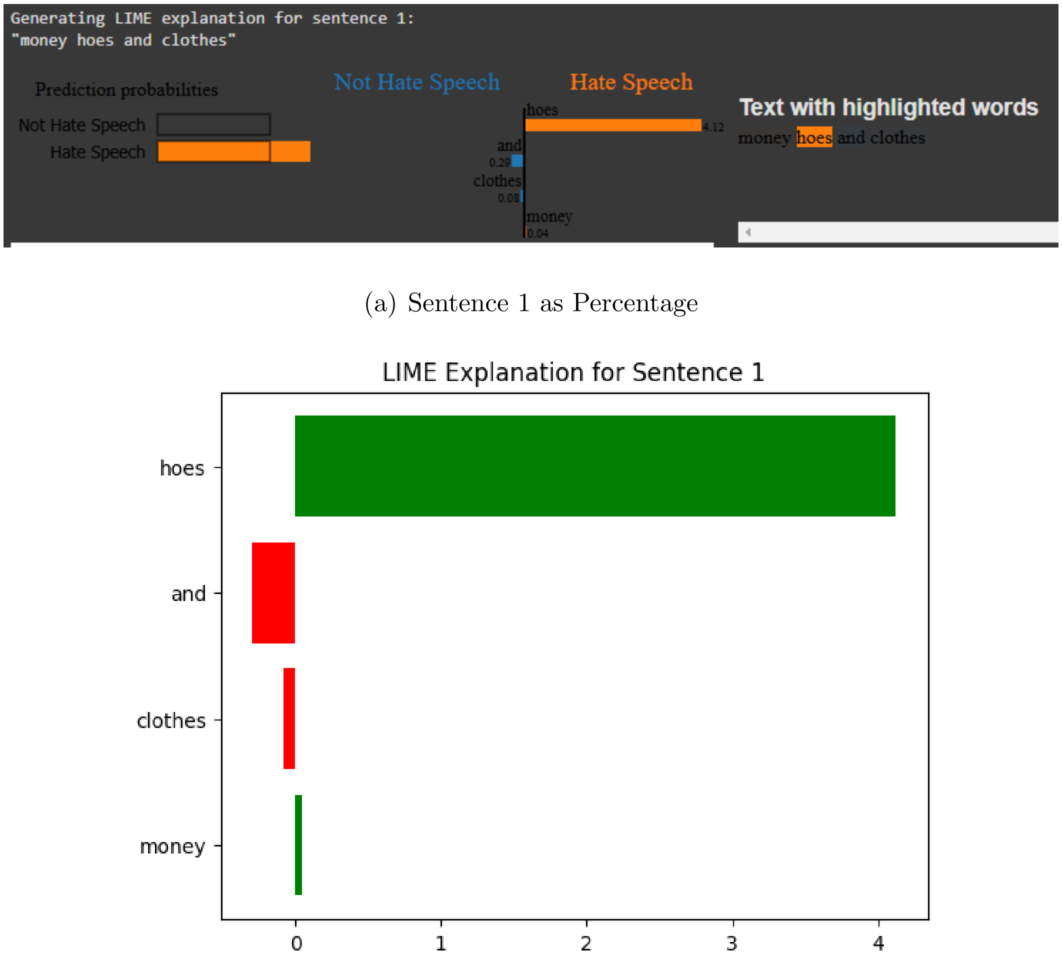
(b) AUC Curve

Figure 4.20: Visualized representation of ROC and AUC of DistilBERT after applying Grid Search

4.6 XAI(Explainable Artificial Intelligence)

Explainable Artificial Intelligence (XAI) for Natural Language Processing (NLP) focuses on making NLP models, such as deep learning architectures, interpretable and transparent. It provides insights into how models process language and make decisions, helping users understand the importance of specific words, phrases, or structures in tasks like sentiment analysis or text classification. Techniques such as attention visualization, saliency maps, SHAP, LIME, and model-agnostic methods are commonly used to explain model behavior. XAI is crucial for addressing ethical concerns, debugging models, and improving trust, particularly in high-stakes applications like healthcare, legal, and finance. By making NLP systems more interpretable, XAI promotes accountability and enables better alignment with human values. In the second figure of explanation the green bar shows hate speech and redone indicates non hate speech.

4.6.1 Optuna



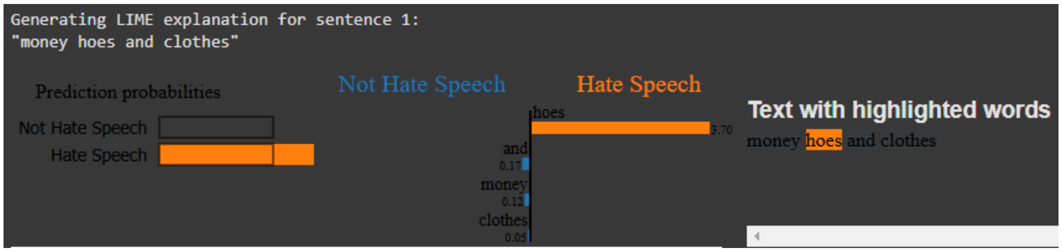
```
Generating LIME explanation for sentence 1:
"money hoes and clothes"
```

(a) Sentence 1 as Percentage

(b) Sentence 1 as bar graph

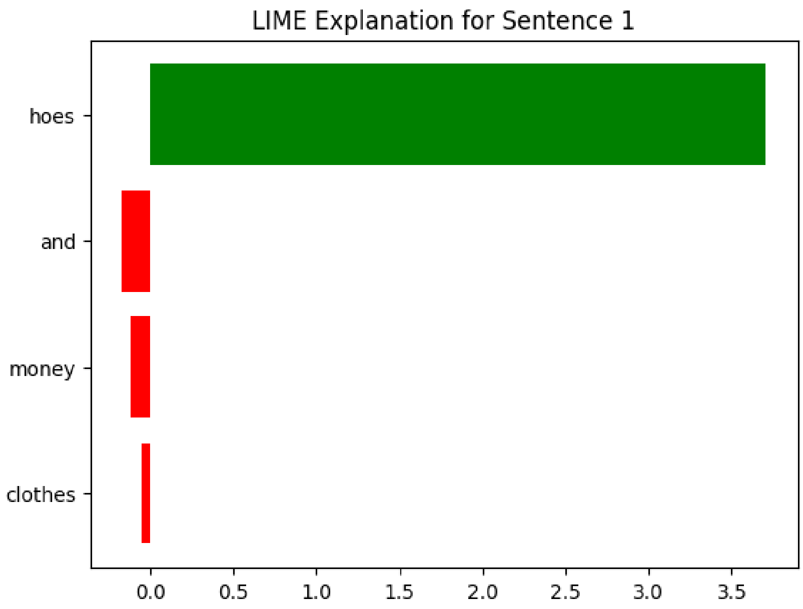
Figure 4.21: LIME explanation for Optuna

4.6.2 Random Search



```
Generating LIME explanation for sentence 1:
"money hoes and clothes"
```

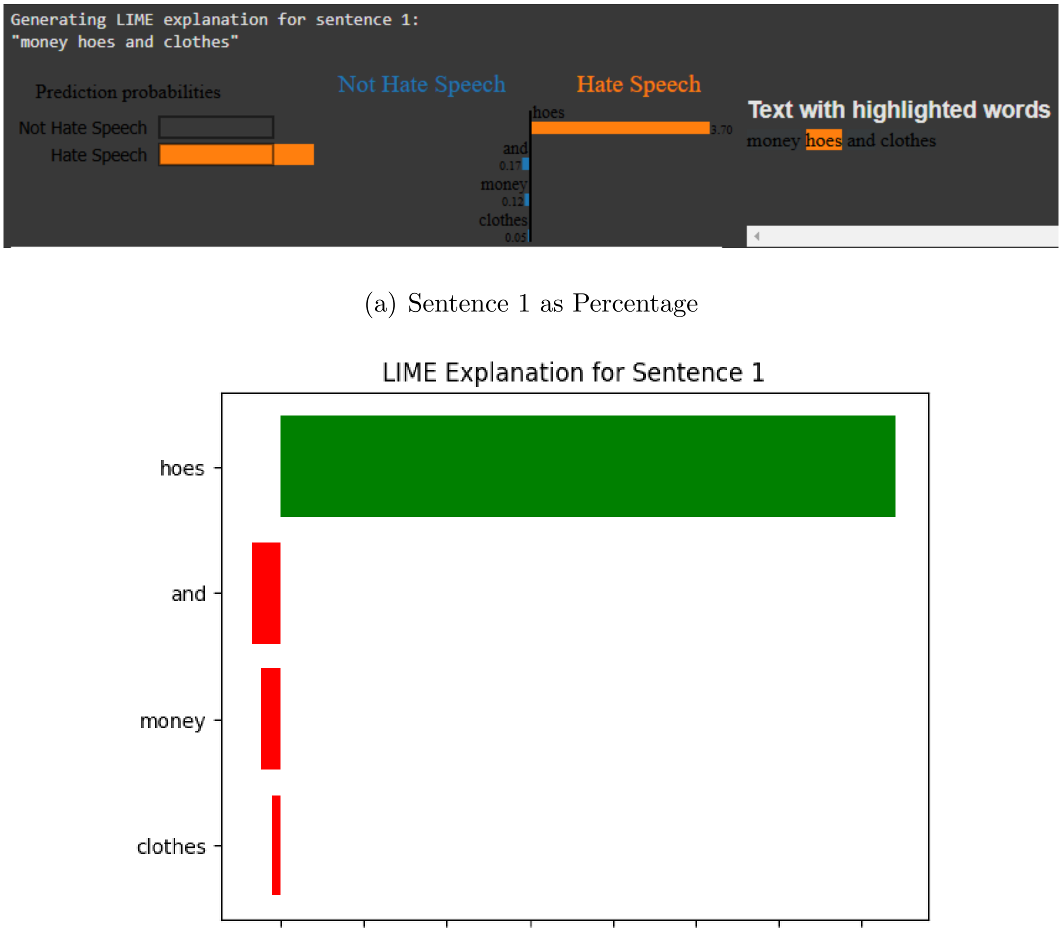
(a) Sentence 1 as Percentage



(b) Sentence 1 as bar graph

Figure 4.22: LIME explanation for Randon Search

4.6.3 Grid Search



```
Generating LIME explanation for sentence 1:
"money hoes and clothes"
```

(a) Sentence 1 as Percentage

(b) Sentence 1 as bar graph

Figure 4.23: LIME explanation for Grid Search

4.7 Summary

This chapter provided a detailed implementation, evaluation, and analysis of machine learning models and deep learning models for Hate Speech detection in Bangali and English. Significant improvements in model performance were observed after applying Hyper parameter tuners, with Optuna and Grid Search achieving the highest accuracy.

Chapter 5

Standards, Constraints and Milestones

5.1 Introduction

The standards, impacts, ethics, and challenges of the thesis work are highlighted in this section. Next, the Alternatives and Constraints are displayed. Lastly, the tasks, milestones, and schedules for the planned work are shown.

5.2 Standards

We have given careful consideration to sustainability in the design of our Hate Speech Detection Model, ensuring both long-term effectiveness and minimal environmental impact. The model reduces energy consumption and operating costs by optimizing the use of computational resources through efficient machine learning algorithms. It incorporates procedures for regular maintenance and updates, ensuring it remains accurate and reliable as linguistic trends and online behaviors evolve. Additionally, the model's architecture is highly scalable, allowing for future enhancements without requiring significant resource investment. This forward-thinking approach promotes continuous improvement while aligning with ethical principles and supporting efforts to foster safer online communities.

5.3 Impacts on Society

Our Hate Speech Detection Model plays a significant role in society by enabling the early and accurate identification of hate speech across various platforms. By detecting harmful content promptly, the model helps mitigate the spread of toxic language, fostering a safer and more inclusive online environment. Its implementation reduces the burden on moderators and support teams by providing precise and efficient detection capabilities. Additionally, it supports public welfare by discouraging harmful behaviors and promoting respectful communication. Ultimately, the model contributes to healthier online interactions, empowering communities to engage in more constructive and informed discussions.

5.4 Ethics

Data security will be our top priority in operating our Hate Speech Detection Model in accordance with ethical standards. The model employs robust data anonymization techniques and adheres to all relevant regulations to protect user privacy and sensitive information. Its development and deployment are grounded in transparency and fairness, ensuring unbiased results and equitable access to its capabilities across diverse communities.

5.5 Challenges

One of the key challenges that machine learning models for hate speech detection must address is ensuring accuracy and reliability when dealing with diverse linguistic and cultural contexts. Overfitting to specific datasets or languages can hinder the model's ability to generalize effectively across different communities. Balancing interpretability with model complexity is critical for fostering trust and facilitating practical applications. Additionally, handling user-generated content requires strict measures to protect data privacy and security. Finally, overcoming logistical and technological barriers is crucial for seamlessly integrating the model into existing platforms and moderation workflows.

5.6 Constraints

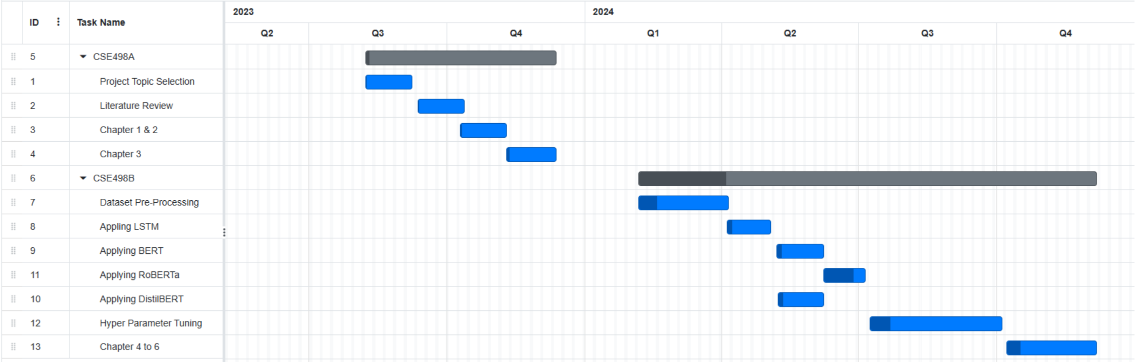
Budget, component, and design constraints are just a few of the limitations covered in this section. An overall structure based on the categorical dataset is presented. Our modal will not function properly without a medium-sized processor to handle this volume of data. Nevertheless, a graphics processing unit (GPU) is not needed. We use this component to train our model:

- Minimum processor: AMD Ryzen 5 5600g with Vega 7(5th Gen)
- Minimum memory: 16GB (DDR4, 3200MHz)

However, because the price of the product component varies, budgets might fluctuate in the market.

5.7 Timeline and Gantt Chart

Our thesis work is divided into two parts because we have two semesters to complete it. We followed our supervisor's instructions and finished the task. In the first semester, we evaluated the relevant thesis work and submitted a proposal. We also used existing systems for analysis and planning to create a prototype of the planned systems. During the second semester, we processed the dataset, designed the complete layout, tested it with the dataset, and then reported on the workflow as a whole. The Gantt Chart (Figure 5.20) shows how the work execution process for finishing this thesis work is organized. The thesis work is performed in two semesters, with each semester lasting six months, for a total of twelve months.



ID	Task Name
5	CSE498A
1	Project Topic Selection
2	Literature Review
3	Chapter 1 & 2
4	Chapter 3
6	CSE498B
7	Dataset Pre-Processing
8	Appling LSTM
9	Applying BERT
11	Applying RoBERTa
10	Applying DistilBERT
12	Hyper Parameter Tuning
13	Chapter 4 to 6

Figure 5.1: Visualization of the Gantt Chart

5.8 Summary

In conclusion, the standards, implications, ethics, and difficulties of the thesis work are briefly discussed in this chapter. The limitations, options, schedules, assignments, and completion dates of the suggested work are also displayed.

Chapter 6

Conclusion

6.1 Introduction

Hate speech is a significant societal concern that requires prompt detection to mitigate its harmful effects and promote safer online interactions. This study explores the application of machine learning to improve hate speech detection processes. The model development begins with a thorough review of research literature to identify advancements and gain insights into current methodologies. Subsequently, a diverse dataset containing labeled examples of hate speech and non-hate speech is compiled. Key preprocessing steps, such as tokenization, stop-word removal, and handling imbalanced data, are meticulously performed. The model is then designed and optimized using machine learning algorithms to effectively differentiate hate speech from non-hate speech. Its performance is rigorously evaluated through iterative training and testing cycles to ensure reliability and robustness.

6.2 Future Works and Limitations

In the future, the HS detection model will be improved even more by expanding the dataset, addressing data imbalance, and researching complex algorithms to boost generalizability and accuracy. An intuitive website that makes results more accessible and enables the model to be incorporated into the workflows is another crucial goal. This website would facilitate the input and output of real-time data, thereby enhancing the model's suitability and facilitating correspondence with social media users. The model will need to be continuously validated and updated in order to remain relevant and adapt to new data. Using categorical data only in the model may make it more challenging to spot subtle patterns, and there may be issues with generalizability when applying the model to other populations. Besides, the model's performance is also affected by the hardware limitation and access to the required resources and time.

References

[1] S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, *A Deep Dive into Multilingual Hate Speech Classification*, 02 2021, pp. 423–439.

[2] M. Jobair, D. Das, B. Islam, and M. Dhar, “Bengali hate speech detection with bert and deep learning models,” 08 2023.

[3] T. Moy, M. Raheem, and R. Logeswaran, “Multilingual hate speech detection,” vol. 4, pp. 19–28, 03 2022.

[4] I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas, “Ethos: a multi-label hate speech detection dataset,” *Complex & Intelligent Systems*, vol. 8, no. 6, pp. 4663–4678, 2022.

[5] M. Mozafari, R. Farahbakhsh, and N. Crespi, “Cross-lingual few-shot hate speech and offensive language detection using meta learning,” *IEEE Access*, vol. 10, pp. 14 880–14 896, 2022.

[6] N. Romim, M. Ahmed, M. S. Islam, A. S. Sharma, H. Talukder, and M. R. Amin, “Hs-ban: A benchmark dataset of social media comments for hate speech detection in bangla,” *arXiv preprint arXiv:2112.01902*, 2021.

[7] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, “Hate speech detection: Challenges and solutions,” *PloS one*, vol. 14, no. 8, p. e0221152, 2019.

[8] A. K. Das, A. Al Asif, A. Paul, and M. N. Hossain, “Bangla hate speech detection on

social media using attention-based recurrent neural network," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578–591, 2021.

[9] M. Das, S. Banerjee, P. Saha, and A. Mukherjee, "Hate speech and offensive language detection in bengali," *arXiv preprint arXiv:2210.03479*, 2022.

[10] N. Romim, M. Ahmed, H. Talukder, and M. Saiful Islam, "Hate speech detection in the bengali language: A dataset and its baseline evaluation," in *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*. Springer, 2021, pp. 457–468.

[11] A. A. Ayele, S. Dinter, S. M. Yimam, and C. Biemann, "Multilingual racial hate speech detection using transfer learning," in *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, 2023, pp. 41–48.

[12] R. Manuvie and S. Chatterjee, "Automated sentiment and hate speech analysis of facebook data by employing multilingual transformer models," *arXiv preprint arXiv:2301.13668*, 2023.

[13] A. Yadav, S. Chandel, S. Chatufale, and A. Bandhakavi, "Lahm: Large annotated dataset for multi-domain and multilingual hate speech identification," *arXiv preprint arXiv:2304.00913*, 2023.

[14] T. Srikissoon and V. Marivate, "Combating hate: How multilingual transformers can help detect topical hate speech," *EPiC Series in Computing*, vol. 93, pp. 203–215, 2023.

[15] V. Hangya and A. Fraserl, "Lmu at haspeede3: Multidataset training for cross-domain hate speech detection," 2023.

[16] D. Nozza, F. Bianchi, G. Attanasio *et al.*, "Hate-ita: hate speech detection in

italian social media text," in *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH).* Association for Computational Linguistics, 2022.

[17] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, "A multi-task learning approach to hate speech detection leveraging sentiment analysis," *IEEE Access*, vol. 9, pp. 112 478–112 489, 2021.

[18] M. Almaliki, A. M. Almars, I. Gad, and E.-S. Atlam, "Abmm: Arabic bert-mini model for hate-speech detection on social media," *Electronics*, vol. 12, no. 4, p. 1048, 2023.

[19] P. Chiril, E. W. Pamungkas, F. Benamara, V. Moriceau, and V. Patti, "Emotionally informed hate speech detection: a multi-target perspective," *Cognitive Computation*, pp. 1–31, 2022.

[20] S. Biere, S. Bhulai, and M. B. Analytics, "Hate speech detection using natural language processing techniques," *Master Business AnalyticsDepartment of Mathematics Faculty of Science*, 2018.

[21] M. Lai, F. Celli, A. Ramponi, S. Tonelli, C. Bosco, and V. Patti, "Haspeede3 at evalita 2023: Overview of the political and religious hate speech detection task," in *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR. org, Parma, Italy*, 2023, pp. 1–8.

[22] A. J. Keya, M. M. Kabir, N. J. Shammey, M. Mridha, M. R. Islam, and Y. Watanobe, "G-bert: An efficient method for identifying hate speech in bengali texts on social media," *IEEE Access*, 2023.

[23] M. Jobair, D. Das, N. B. Islam, and M. Dhar, "Bengali hate speech detection with bert and deep learning models."

[24] J. M. Pérez, F. M. Luque, D. Zayat, M. Kondratzky, A. Moro, P. S. Serrati, J. Zajac, P. Miguel, N. Debandi, A. Gravano *et al.*, “Assessing the impact of contextual information in hate speech detection,” *IEEE Access*, vol. 11, pp. 30 575–30 590, 2023.

[25] A. K. Das, A. Al Asif, A. Paul, and M. N. Hossain, “Bangla hate speech detection on social media using attention-based recurrent neural network,” *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578–591, 2021.