

Project Report On

# HATE SPEECH DETECTION

“A dissertation submitted in partial fulfillment of the requirements of Bachelor of Technology Degree in Computer Science and Engineering of the Maulana Abul Kalam Azad University of Technology for the year 2021-2025”



Submitted by

Shubhrajit Ghosh ( 14800121107 )  
Barsha Dutta ( 14800121061 )  
Debjit Paul ( 14800121044 )  
Projjal Paul ( 14800121041 )

Under the guidance of

Satyam Raha  
Assistant Professor  
Dept of Computer Science & Engineering  
Future Institute of Engineering and Management

Department of Computer Science and Engineering  
**Future Institute of Engineering & Management**

(Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal)

Kolkata - 700150, WB



### **Certificate of Approval**

This is to certify that this report of B. Tech. 8<sup>th</sup> semester project, entitled “**HATE SPEECH DETECTION**” is a record of bona-fide work, carried out by Shubhrajit Ghosh, Barsha Dutta, Debjit Paul, Projjal Paul under my supervision and guidance.

In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the *Future Institute of Engineering & Management* and as per regulations of the *Maulana Abul Kalam Azad University of Technology*. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for B. Tech. program in Computer Science and Engineering in the year 2021-2025.

It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve this thesis for the purpose for which it is submitted.

**Guide / Supervisor**

---

Department of Computer Science and Engineering  
Future Institute of Engineering & Management

---

**Examiner(s)**

---

**Head of the Department**

Computer Science and Engineering  
Future Institute of Engineering & Management

## ACKNOWLEDGEMENT

We, the members of the project team, are collectively expressing our sincere gratitude to all the individuals and organizations who are supporting and guiding us throughout the development of our project “**HATE SPEECH DETECTION**”.

First and foremost, we are deeply thankful to our academic guide, **Dr. Ratul Chowdhury, Satyam Raha (Assistant Professor, CSE)**, for his invaluable advice, constructive criticism, and continuous support. Their guidance is helping us stay focused and motivated throughout the project development.

We are also extending our heartfelt thanks to the **Future Institute of Engineering and Management** and the Department of Computer Science & Engineering for providing us with the necessary resources and an encouraging environment to carry out this project.

Lastly, we are acknowledging the unwavering support and encouragement of our families, who are constantly motivating us and providing us with the patience and resources to work on this project.

This project is a true reflection of teamwork, and we are thankful to everyone who has contributed to making it a reality.

Name:

University Roll No.: 14800121107

Registration No.: 211480100110009

Name:

University Roll No. :14800121044

Registration No.: 211480100110002

Name:

University Roll No.: 14800121061

Registration No.: 211480100110028

Name:

University Roll No.: 14800121041

Registration No.: 211480100110086

# PROJECT ABSTRACT

Hate speech detection refers to the task of identifying and classifying harmful, offensive, or discriminatory language targeted at individuals or groups based on attributes like race, religion, gender, or other protected characteristics. With the rise of social media platforms, the challenge of accurately detecting hate speech has become critical, as these platforms are inundated with diverse user-generated content. A significant challenge is distinguishing harmful speech from other forms of offensive or controversial language, which might be context-dependent or not inherently harmful. Moreover, existing datasets often suffer from imbalanced class distributions, contextual ambiguities, data scarcity, and labeling inconsistencies, which exacerbate the difficulty of this task.

In recent years, researchers have explored deep learning models to address these challenges, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers. Despite progress, many issues related to contextual understanding and handling of imbalanced data remain unexplored. This paper introduces a novel approach to tackle these challenges by leveraging transfer learning based on pre-trained language models, specifically BERT (Bidirectional Encoder Representations from Transformers), in conjunction with sequence models like LSTM (Long Short-Term Memory) and BiLSTM (Bidirectional Long Short-Term Memory). Our approach integrates BERT's contextual embeddings with LSTM and BiLSTM layers, which are known for their ability to capture long-range dependencies and contextual nuances in text.

Through extensive experiments, our model achieves an impressive F1 score of 90%, demonstrating its robustness in identifying and classifying hate speech across contexts. We also compare our approach to baseline models and show that our framework outperforms traditional methods in both classification accuracy and F1 score. Our framework not only provides a practical solution to the challenges faced by existing systems but also serves as a solid baseline for future research in this domain.

In conclusion, the integration of transfer learning with BERT and sequence models such as LSTM and BiLSTM holds great promise for improving the accuracy of hate speech detection systems. This paper contributes to the growing body of research by offering a novel and effective approach that could be applied to various real-world applications in moderating online platforms, improving safety, and enabling automated detection of harmful content.

# CONTENTS

<b>CHAPTER 1</b>	<b>INTRODUCTION</b>	
	1.1. Motivation	1
	1.2. Background	2
	1.3. Summary of work	3-5
	1.4. Organization of the thesis	5-6
	1.5. Resource (Software/ Hardware) used.	6
<b>CHAPTER 2</b>	<b>EXISTING WORK STUDY</b>	7-8
<b>CHAPTER 3</b>	<b>PROPOSED METHOD</b>	
	3.1. Dataset	9
	3.2. Proposed Framework	9-14
	3.2.1. Preprocessing	
	3.2.2. Fine-Tuning Strategies	
	3.2.3. Proposed Algorithm	
	3.2.4. Proposed Model	
<b>CHAPTER 4</b>	<b>EXPERIMENTAL RESULT AND ANALYSIS</b>	15-16
<b>CHAPTER 5</b>	<b>COMPARATIVE STUDY</b>	17-18
<b>CHAPTER 6</b>	<b>CONCLUSION</b>	19
<b>CHAPTER 7</b>	<b>FUTURE SCOPE</b>	20
<b>REFERENCE</b>		21

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation

In recent years, social networking platforms have become an integral part of everyday life, offering unprecedented opportunities for people to interact, share opinions, and exchange information. These platforms, such as Twitter, Facebook, and Instagram, have revolutionized the way individuals communicate, making it easier to connect with others globally. However, this widespread connectivity has also given rise to a significant challenge—hate speech. As the volume of user-generated content grows, so does the prevalence of harmful, offensive, and discriminatory language, often aimed at individuals or groups based on characteristics like race, religion, gender, or sexual orientation. Hate speech is a growing concern on social media, with its effects reaching beyond the virtual world to influence real-world behavior and societal dynamics.

The ease of access, mobility, and anonymity afforded by online platforms have allowed malicious actors to exploit these spaces for propagating harmful ideologies, engaging in harassment, and organizing hate-based activities. The rise of hate speech has led to violence and discrimination in various forms, impacting individuals' mental health, public safety, and even societal harmony. The motivation for this project arises from the urgent need for such automated hate speech detection systems. Current approaches often suffer from limitations such as data imbalance, contextual ambiguity, and insufficient understanding of nuanced expressions of hate speech. To address these challenges, we propose a novel approach that leverages state-of-the-art deep learning techniques, including transfer learning with pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) and sequence models such as LSTM (Long Short-Term Memory) and BiLSTM (Bidirectional Long Short-Term Memory). By combining these models, we aim to enhance the system's ability to capture context, detect subtle forms of hate speech, and handle issues such as data imbalance and limited labeled datasets. Our model takes advantage of BERT's powerful contextual embeddings, coupled with the long-range dependency capabilities of LSTM and BiLSTM, to improve the detection of hate speech in a wide range of contexts. To address these challenges, we propose a novel approach that leverages state-of-the-art deep learning techniques, including transfer learning with pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) and sequence models such as LSTM (Long Short-Term Memory) and BiLSTM (Bidirectional Long Short-Term Memory).

## 1.2 Background

Recently, the problem of online abusive detection has attracted scientific attention. Proof of this is the creation of the third Workshop on Abusive Language Online<sup>3</sup> or Kaggles Toxic Comment Classification Challenge that gathered 4,551 teams<sup>4</sup> in 2018 to detect different types of toxicities (threats, obscenity, etc.). With the availability of larger datasets, researchers started using complex models to improve the classifier performance. These include deep learning and graph embedding techniques to detect hate speech in social media posts. Zhang et al used deep neural network, combining convolutional. Huang et al. used Twitter hate speech corpus from five languages and annotated them with demographic information. Using this new dataset they study the demographic bias in hate speech classification. Corazza et al. used three datasets from three languages (English, Italian, and German) to study the multilingual hate speech. The authors used models such as SVM, and Bi-LSTM to build hate speech detection models. Davidson et al. [3] collected a 24K corpus of tweets containing hate speech keywords and labelled the corpus as hate speech, offensive language, or neither by using crowd-sourcing and extracted different features such as n-grams, some tweet-level metadata such as the number of hashtags, mentions, retweets, and URLs, Part Of Speech (POS) tagging, etc. Their experiments on different multi-class classifiers showed that the Logistic Regression with L2 regularization performs the best at this task. Waseem et al. brought a new insight to hate speech and abusive language detection tasks by proposing a multi-task learning framework to deal with datasets across different annotation schemes, labels, or geographic and cultural influences from data sampling.

Several studies (Mentioned in Table 1) have explored the detection of hate speech using different machine learning and deep learning techniques. Gupta and Waseem provide a comprehensive survey on neural networks for hate speech detection, highlighting various architectures and discussing challenges like data sparsity and model interpretability. Nick Savage et al. used Convolutional Neural Networks (CNN) to analyze tweets labeled as hate, offensive, or neutral, showing good performance but with some misclassifications, and suggesting that larger datasets could improve results. Saleh et al. focused on detecting hate speech in videos by first extracting audio, converting it to text, and then applying machine learning models for classification. M. U. Akram reviewed deep learning methods, such as Recurrent Neural Networks (RNNs), CNNs, and Long Short-Term Memory (LSTM) networks, finding them to outperform traditional machine learning algorithms for hate speech detection. These studies collectively underscore the importance of using advanced models and larger, more diverse datasets for effective hate speech detection.

Source	Technology Used	Data set	Accuracy
Gupta and Waseem (20217)	W2V(300)	Waseem-EMNLP Davidson-ICWSM	91% 84%
Hind Saleh, Areej Alhothali & Kawthar Moria	Bidirectional LSTM	Davidson-ICWSM Waseem-NAACL	91% 76%
Nick Savage	BERT CNN	ID Tweet-text HS	70% 73%
M. U. Akram	Transformer	Multiclass hate speech and offensive (HSO) language	96%

Table 1: Summary of the existing Work

### 1.3 Summary of the work

In the scope of this work, we mainly focus on the term hate speech as abusive content in social media, since it can be considered a broad umbrella (*Mentioned in Figure: 1*) term for numerous kinds of insulting user-generated content. To detect online hate speech, a large number of scientific studies have been dedicated by using Natural Language Processing (NLP) in combination with Machine Learning (ML) and Deep Learning (DL) methods. Although supervised machine learning-based approaches have used different text mining-based features such as surface features, sentiment analysis, lexical resources, linguistic features, knowledge-based features or user-based and platform-based metadata. These newer models are applying deep learning approaches such as BERT (Bidirectional Encoder Representations from Transformers) , LSTM (Long Short-Term Memory, which is a type of recurrent neural network (RNN) and Bi-directional Long Short-Term Memory (BiLSTM) etc. to enhance the performance of hate speech detection models, however, they still suffer from lack of labelled data or inability to improve generalization property.



Figure 1: Abusive language phenomena and their relationships



When it comes to offensive language, abusive language, and hate speech, the distinguishing factor is their level of specificity. This makes offensive language the most generic form of abusive language phenomena and hate speech the most specific, with abusive language being somewhere in the middle. Our project proposes a framework for hate speech detection by leveraging an ensemble of Large Language Models (LLMs) and a transfer learning approach. We combined the unsupervised pre-trained model XLM-RoBERTa with diverse datasets to enhance performance. Multiple models were implemented and compared to identify the optimal solution. Below is a concise overview of the models and their contributions:

### **A. Long Short-Term Memory (LSTM)**

LSTM is a recurrent neural network (RNN) that uses memory cells with gates to manage information flow, enabling it to capture long-term dependencies. Initially, we used LSTM to process input sentences in a distributed word representation format, achieving a decent accuracy score. However, its limitations in contextual understanding prompted further exploration.

### **B. Bidirectional LSTM (BiLSTM)**

BiLSTM overcomes LSTM's unidirectional nature by processing input sequences in both forward and backward directions. This bidirectional approach captures richer contextual information, resulting in improved performance over standard LSTM, making it more effective for hate speech detection.

### **C. BERT (Bidirectional Encoder Representations from Transformers)**

BERT processes text bidirectionally, capturing context from both sides of a word. Pretrained on large corpora, BERT demonstrated significant improvements in understanding nuanced language, achieving better results than LSTM and BiLSTM for hate speech detection. The architecture is mentioned in Figure 3.

### **D. XLM-RoBERTa**

XLM-RoBERTa (Mentioned in Figure 2), a variant of RoBERTa, is optimized for cross-lingual tasks. Trained on a multilingual corpus, it effectively handles hate speech detection across languages. Its robust pretraining and tokenization techniques enabled state-of-the-art performance in our experiments, making it the optimal model for this task.

Our key contributions are:

- (i) Additional evidence that further pre-training is a viable strategy to obtain domain-specific or language variety-oriented models in a fast and cheap way.
- (ii) A pre-trained BERT for abusive language phenomena, intended to boost research in this area.
- (iii) The release of a large-scale dataset of social media posts in Bengali from communities banned for being offensive, abusive, or hateful.

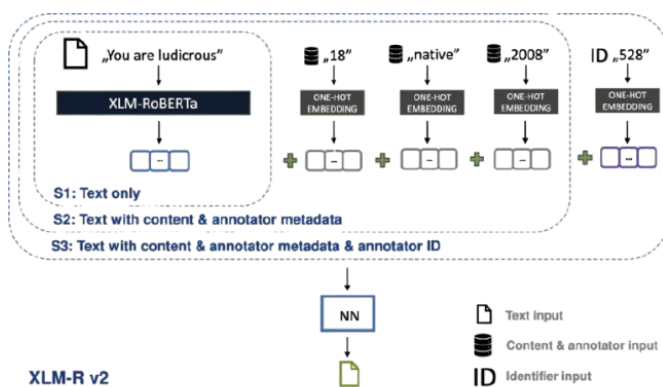


Figure 2: XLM-RoBERTa architecture

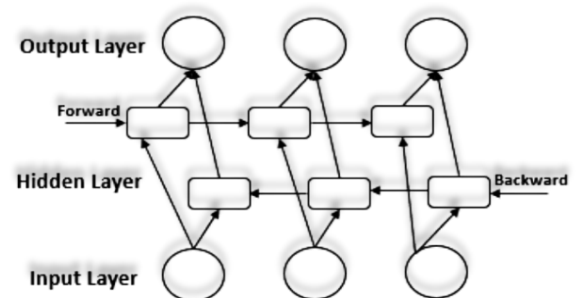


Figure 3: Bi-LSTM Architecture

Our framework systematically evaluated LSTM, BiLSTM, BERT, and XLM-RoBERTa, culminating in the selection of XLM-RoBERTa as the most effective model. This project highlights the importance of advanced LLMs and transfer learning for addressing complex NLP tasks, especially in multilingual contexts. The findings pave the way for further research in hate speech detection and related applications.

## 1.4 Organization of thesis

### i) Introduction

The introduction outlines the objective of the study, describing the problem of hate speech detection and the motivation behind applying machine learning techniques like text classification. It provides an overview of the methods employed, including K-Fold Cross Validation, the XLM-RoBERTa model, and evaluation metrics like the F1 score. The introduction concludes with a brief structure of the thesis.

## **ii) Literature Review**

The literature review covers prior works in machine learning for text classification, focusing on deep learning models like XLM-RoBERTa. It discusses the significance of the F1 score for imbalanced datasets and the importance of K-Fold Cross Validation for ensuring robust model performance.

## **iii) Methodology**

The methodology describes the dataset, including its features, preprocessing steps, and tokenization. It explains the rationale behind selecting XLM-RoBERTa for sequence classification and details the implementation of K-Fold Cross Validation. Insights into the training process, including the optimizer choice, hyperparameters, and validation data, are also provided.

## **iv) Experimental Setup**

The experimental setup highlights the software and hardware environment used, including Python libraries and GPU specifications. It discusses data preparation and training configurations, such as batch size and learning rate, to ensure reproducibility.

## **v) Results**

The results present the performance metrics, specifically the F1 scores for each fold, and calculate the average F1 score.

## **vi) Discussion**

It addresses challenges encountered during training and validation, such as computational limitations and data preprocessing issues. Model interpretability methods are also discussed to provide insights into the model's decision-making process.

## **1.5 Resources (Software/Hardware)**

Hardware:

1. NVIDIA Tesla T4 GPU for model training and evaluation.
2. High-performance computing environment with sufficient memory and processing power

Software:

1. Python 3.8 as the primary programming language.
2. Libraries: Transformers, PyTorch, Scikit-learn, Pandas, and NumPy.

## **CHAPTER 2**

### **EXISTING WORK STUDY**

The detection of online abusive language has been extensively studied, with significant contributions from various research efforts. These contributions can be categorized into different approaches based on methodologies, datasets, and the challenges addressed.

#### **i) Early Approaches and Dataset Creation**

Early research in abusive language detection focused on the creation of large, labeled datasets and the application of traditional machine learning models. Davidson et al. collected a 24,000-tweet corpus labeled as hate speech, offensive, or neutral, and found that Logistic Regression with L2 regularization performed the best for classification. Zhang et al. introduced deep neural networks, particularly convolutional layers, to detect hate speech, marking a shift towards more complex models. Huang et al. expanded this work by using a multilingual Twitter corpus annotated with demographic data to explore bias in hate speech detection.

#### **ii) Deep Learning Approaches**

With the rise of deep learning, several studies explored neural network-based approaches to improve detection accuracy. Gupta and Waseem conducted a comprehensive survey on the use of RNNs, CNNs, and LSTMs for hate speech detection, discussing the challenges of data sparsity and model interpretability. Nick Savage et al. demonstrated the effectiveness of CNNs for analyzing tweets, although they noted misclassifications, suggesting that larger datasets could improve results. Saleh et al. focused on detecting hate speech in videos by first extracting audio and applying machine learning models for classification. M. U. Akram reviewed deep learning techniques such as RNNs, CNNs, and LSTMs, finding them to outperform traditional methods like SVMs in hate speech detection.

#### **iii) Multilingual and Cross-Lingual Approaches**

The detection of hate speech across multiple languages has become an important area of research. Huang et al. used a multilingual Twitter corpus to study hate speech detection in five languages, highlighting challenges like demographic bias. Corazza et al. expanded on this by using datasets in English, Italian, and German and applying models such as SVM and Bi-LSTM. Cross-lingual transfer

learning and multilingual embeddings have become essential to overcoming language-specific challenges, such as code-switching and language identification.

#### **iv) Bias and Fairness in Detection Models**

The issue of bias in hate speech detection has also been explored. Waseem et al. proposed a multi-task learning framework to address datasets with different annotation schemes, labels, and geographic or cultural influences. Wiegand et al. found that classifiers trained on datasets with implicit abuse (e.g., sarcastic or subtle abusive language) were more prone to bias than those trained on datasets with explicit abuse. This work highlighted the need for careful consideration of dataset composition to avoid biased model predictions.

#### **v) Challenges and Future Directions**

Despite significant progress, several challenges remain in the field of hate speech detection. Issues such as data sparsity, model interpretability, and the detection of implicit abuse (e.g., sarcasm or subtle forms of hate speech) continue to hinder model performance. Future research is likely to focus on improving the generalization of models across different languages, platforms, and cultural contexts, addressing fairness and bias, and enhancing model transparency to ensure more reliable and equitable outcomes.

## CHAPTER 3

### PROPOSED METHOD

#### 3.1 Dataset

We investigated the datasets available for hate speech and found 16 publicly available sources. One the immediate issues, we observed was the mixing of several types of categories (offensive, profanity, abusive, insult). Although these categories are related to hate speech, they should not be considered as the same. For this reason, we only use two labels: ‘Neutral’ and ‘Non-neutral’ and discard other labels.

Language	Dataset	Source	Non-neutral	Neutral	Total
English	David et al	Twitter	1430	4163	5593
	Gibert et al	Stormfront	1196	9748	10944
	Waseem et al	Twitter	759	5545	6304
	Basile et al	Twitter	5390	7415	12805
Bengali	Reza et al	Hugging Face	1190	9744	10934

*Table 2: Dataset*

Most of the hate speech datasets are available in English Language like David et al, Gilbert et al, Waseem et al, Basile et al etc. And include Bengali hate speech datasets like Reza et al. This data set helps to retrained out transfer learning model to detect the hate speech. All the worked dataset are Mentioned in *Table 2*.

#### 3.2 Proposed Framework

Here, we analyze the XLM-RoBERTa transformer model for the hate speech detection task. XLM-RoBERTa is a multilingual transformer-based model trained on a diverse corpus of 100 languages, including English Bengali, using 2.5 TB of Common Crawl data. It can efficiently handle the cross-lingual understanding task effectively. The main fact is that XLM-RoBERTa follows the same architecture like RoBERTa and BERT, with a multi-layer bidirectional transformer encoder. The base model contains 12 layers (transformer blocks), 12 self-attention heads and 123 million parameters. XLM-RoBERTa is pre-trained on general multilingual corpora, fine-tuning it on task specific data is essential for the hate speech involves media content. For classification, XLM-RoBERTa insert two

special tokens into each input sequences: [CLS] and [SEP]. The [CLS] token is added in the front of the input sequences and serves as a entire sequences. In our hate speech classification task, the [CLS] token from the final hidden layer is used as the sequence representation for classification.

### **3.2.1 Preprocessing**

#### **i) Text Cleaning**

- At first the input text is data is cleaned to remove unnecessary characters and ensure uniformity.
- Special character and punctuations are removed.
- This process helps the model to only focus on the meaningful text content.

#### **ii) Label Encoding**

- For compatibility of the with the model, the dataset labels are converted into numerical format.
- 'Neutral' is encoded as 0 and 'Non-neutral' is encoded as 1.

#### **iii) Tokenization**

- Here we have used the XLM-RoBERTa tokenizer to tokenize the text data. The tokenizer helps to convert each input text into the sequence of token IDs, which are numerical representation of words.
- During tokenization the key parameters are max\_lenght (Text is padded to a maximum length of 128 tokens), add\_special\_tokens (Special tokens like [CLS] and [SEP] are added to the input),
- pad\_to\_max\_length (Used to ensure that all inputs in a batch have the same length by padding shorter sequences).

#### **iv) Train-Validation-Test Splits**

- The dataset is divided into three parts: Training Set (Used for model training), Validation Set (Used for hyperparameter tuning), Test Set (used for final evaluation of the model performance).
- The data is loaded as CSV file.

### v) Language-Specific Considerations

- The XLM-RoBERTa tokenizer supports multilingual input.
- That helps to be suitable for English, Bengali and other languages.

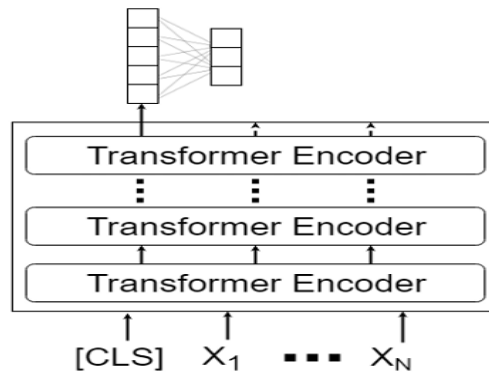
### 3.2.2 Fine-Tuning Strategies

Different layers of a neural network can capture different levels of syntactic and semantic information. The lower layer of the model may contain more general information whereas the higher layers contain task-specific information and we can fine-tune them with different learning rates. Like:

- **BERT based fine-tuning:**

In this approach (*Figure 5*) very few changes are applied to the model. In this architecture, only the [CLS] token output provided by model is used. The [CLS] output, which is equivalent to the [CLS] token output of the 12th transformer encoder, a vector of size 768, is given as input to a fully connected network without hidden layer. The softmax activation function is

applied to the hidden layer to classify. Softmax activation = 
$$\frac{\text{Exp}(I_i)}{\sum_{i=1 \text{ to } n} \text{Exp}(I_i)}$$



*Figure 5: BERT based fine-tuning*



- **Insert nonlinear layers:**

Here, the first architecture is upgraded and an architecture with a more robust classifier is provided in which instead of using a fully connected network without hidden layer, a fully connected network with two hidden layers is used. The first two layers uses the Leaky Relu activation function with negative slope = 0.01, but the final layer, as the first architecture used softmax activation function. (mentioned in Figure 6)

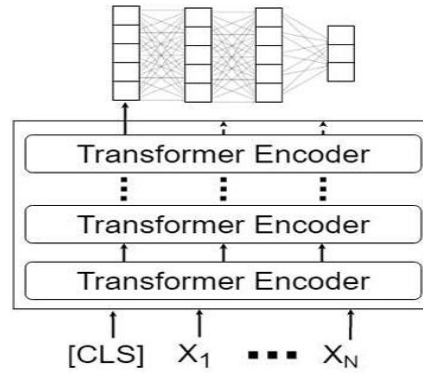


Figure 6: Insert nonlinear layers

- **Insert CNN layers**

In this process (Figure 7) the outputs of all transformer encoders are used instead of using the output of the latest transformer encoder. So that the output vectors of each transformer encoder are concatenated, and a matrix is produced. The convolutional operation is performed with a window and the maximum value is generated for each transformer encoder

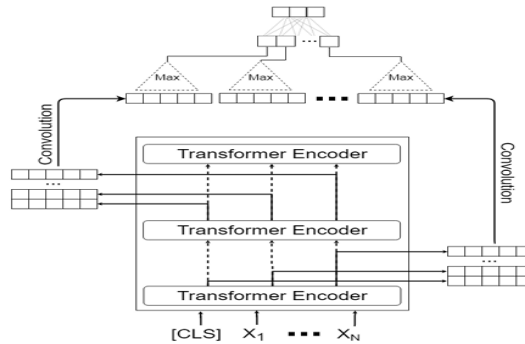


Figure 7: Insert CNN layers

by applying max pooling on the convolution output. By concatenating these values, a vector is generated which is given as input to a fully connected network. By applying softmax on the input, the classification operation is performed.

### 3.2.3 Proposed algorithm

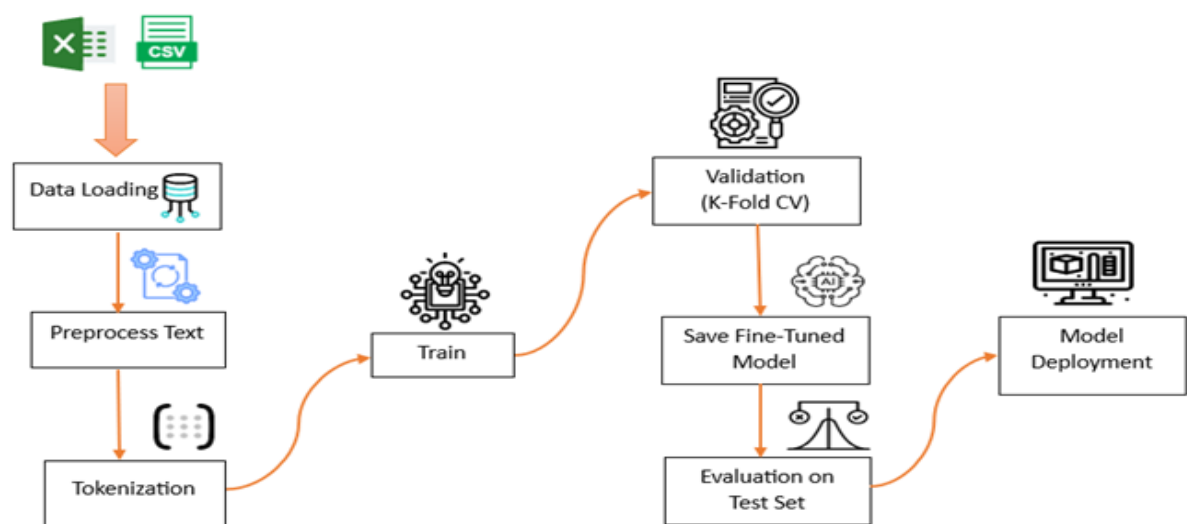
1. **Procedure** (Dataset, Model)
2. Begin
3. Clean\_Dataset  $\leftarrow$  preprocess(Dataset, Regex)  
    # Cleaning the Dataset from unwanted expressions
4. Train, Valid, Test  $\leftarrow$  Split\_Data (Clean\_Dataset)      # The total data is divided into required part  
  
    # Encoding into the suitable matrix for further work
5. Train\_enco  $\leftarrow$  Tokenization(Train)
6. Valid\_enco  $\leftarrow$  Tokenization(Valid)
7. Test\_enco  $\leftarrow$  Tokenization(Test)
8. **For** i  $\leftarrow$  1 to epoch **do**      # Trained over total dataset epoch times
9.     Train(Model, Train\_enco, Valid\_enco)
10.    **For** j  $\leftarrow$  K Folds **do**      # Cross Validation check for early stop
11.     Validation\_check(Model, Valid\_enco)
12.     Evaluation\_score = Score\_Matrix(Model, Test\_enco)
13.     **if** Evaluation\_score == Desired\_Score **then**
14.         Save(Model, Path\_destination) # Saving model in desired path
15.     **End if**
16.    **End For**
17. **End For**
18. Deployment (Model, Gradioa, Streamlit)      # Real Time Interface
19. End
20. **End procedure**

The algorithm outlines the process of hate speech detection using machine learning models, emphasizing dataset preparation, model training, validation, and deployment. It begins by cleaning the dataset to remove unwanted expressions using regular expressions. The cleaned dataset is then split into training, validation, and testing sets. Each split undergoes tokenization, converting text into encoded matrices suitable for processing by the model. The model is trained iteratively over multiple epochs, with training data fed into the model and validation performed at each epoch to monitor performance. During training, K-Fold Cross Validation is employed to ensure robustness and avoid overfitting, enabling early stopping if validation results indicate suboptimal performance. After training, the model is evaluated on the test data using a scoring metric (e.g., F1 score). If the evaluation score meets the desired threshold, the trained model is saved to a specified destination.

Finally, the trained model is deployed using tools like Gradio or Streamlit, providing a real-time interface for practical use. This structured approach ensures a comprehensive and efficient process for training, validating, and deploying a machine learning model for hate speech detection.

### 3.2.4 Proposed Model

The proposed model section is divided into multiple subsections: data loading, preprocessing, tokenization, model training, Fine-tuning and model development. The detailed descriptions of the proposed model have been described in the following subsections shown in *Figure 4*.



*Figure 4: Proposed model*

## **CHAPTER 4**

### **EXPERIMENTAL RESULT AND ANALYSIS**

For the implementation of our hate speech detection model, we utilized the Hugging Face Transformers library, leveraging the pre-trained XLM-RoBERTa models, tokenizers, and associated utilities for fine-tuning. Our experiments were conducted in the Google Collaboratory environment, which provided the necessary computational resources, including a Tesla T4 GPU and 16 GB of RAM. These resources were crucial for efficiently training large models such as XLM-RoBERTa, which is known for its effectiveness in multilingual text classification tasks.

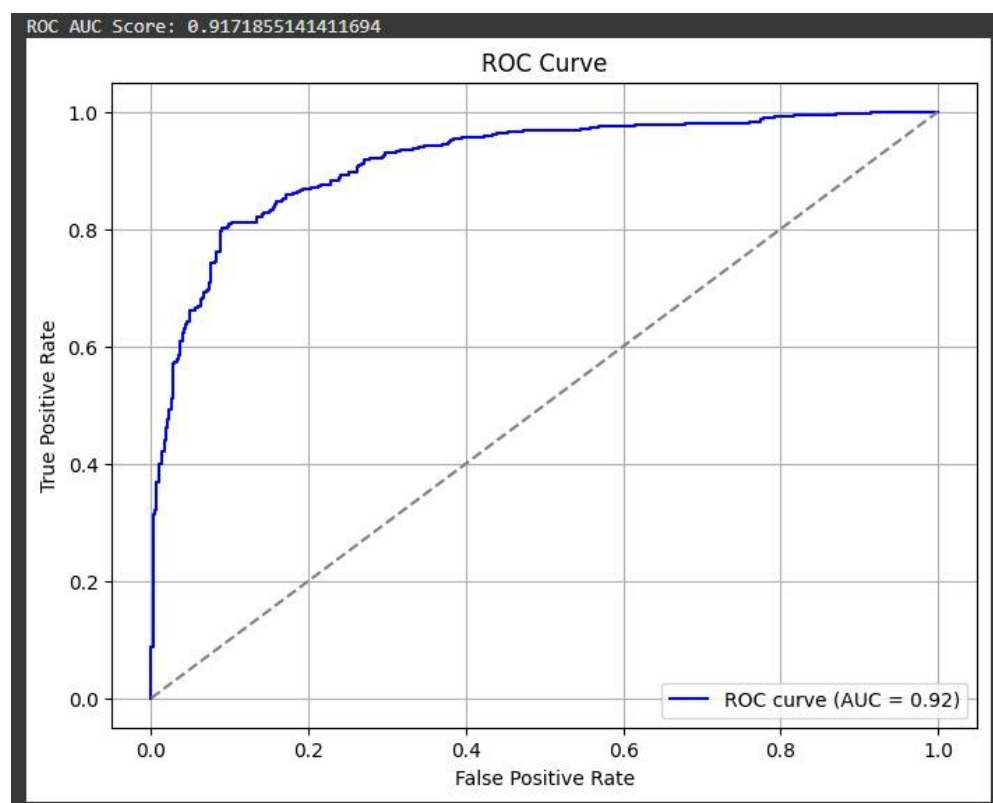
To ensure optimal training performance while managing memory limitations, we set a batch size of 16, which balanced between memory constraints and the stability of the training process. The model was fine-tuned over four epochs, which ensured convergence without overfitting. The use of a relatively small number of epochs helped avoid excessive training time while still achieving effective model performance. We opted for the AdamW optimizer, which has been widely used for fine-tuning transformer models due to its robustness. The learning rate was set at  $3 \times 10^{-5}$ , a commonly used value for fine-tuning large transformer models. Additionally, a linear learning rate scheduler with warm-up steps was employed to stabilize training and avoid sharp learning rate changes that might cause instability.

To combat the risk of overfitting, we introduced a dropout rate of 0.1 to all layers in the model. Dropout is a regularization technique that helps prevent the model from becoming too reliant on any specific feature, promoting generalization. We also set the maximum sequence length to 128 tokens, which ensured that most input texts could be processed without truncation while maintaining computational efficiency, as processing longer sequences would increase memory usage and training time unnecessarily.

Our fine-tuned XLM-RoBERTa model demonstrated impressive results, outperforming traditional baseline models. On the Waseem et al. dataset, the model achieved an F1-score of 83%, and on the Gibert et al. dataset, it reached an even higher F1-score of 87%. These results are indicative of the model's strong ability to differentiate between hate speech and non-hate speech in different contexts and datasets, showcasing its adaptability across varying linguistic features. Furthermore, by adding a fully connected linear classifier on top of the XLM-RoBERTa model, we were able to further fine-tune the system, leading to a remarkable improvement in performance. This extended architecture

allowed the model to make more refined predictions, achieving a substantial F1-score of 91%, which is indicative of the model's effectiveness in classifying hate speech with high accuracy. The pre-trained XLM-RoBERTa model effectively captured both syntactic and contextual information from the text, which was crucial for identifying subtle cues in hate speech. This contextual understanding allowed for significant improvements in classification accuracy, highlighting the model's superior performance over traditional methods and underscoring the potential of using large pre-trained transformer models for hate speech detection in real-world applications.

In *Figure 8* the ROC curve presented showcases the robust performance of our hate speech detection model. With an AUC score of 0.91, the model demonstrates exceptional ability to differentiate between hateful and non-hateful content. The steep curve trajectory indicates consistent high sensitivity and specificity across various threshold settings. This empowers us to fine-tune the model's behavior to prioritize either minimizing false positives or false negatives, depending on the specific use-case requirements.



*Figure 8: Proposed model*

## CHAPTER 5

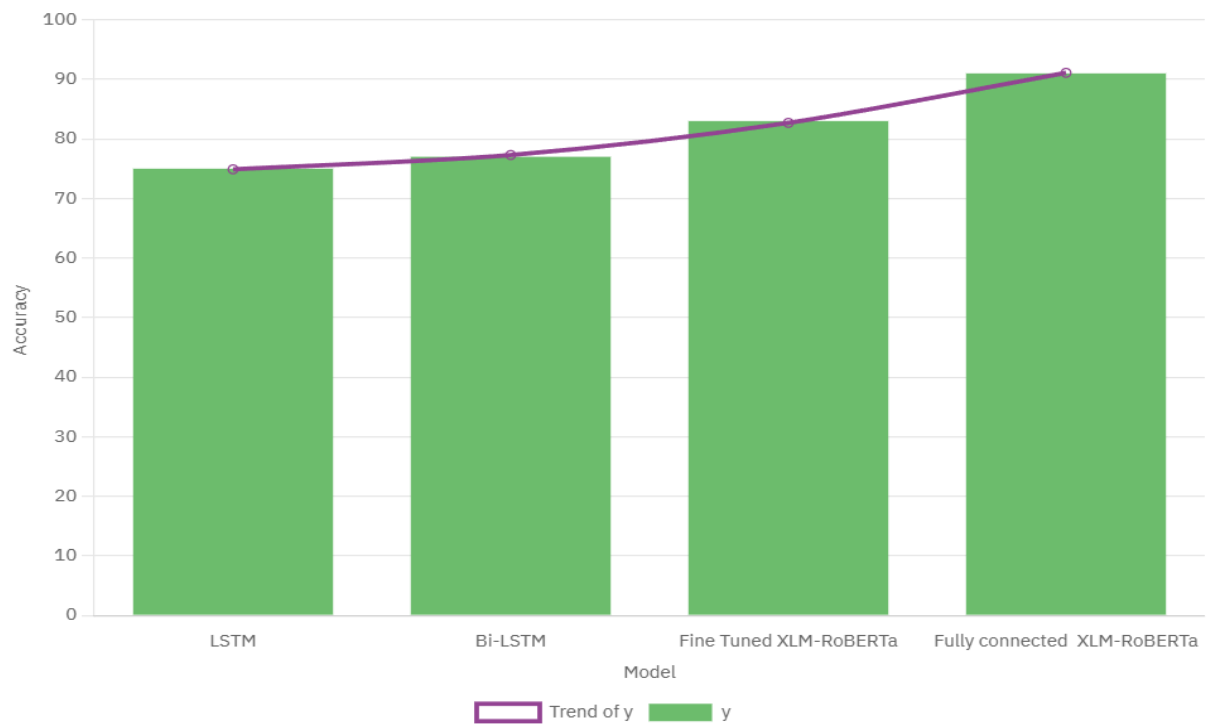
### COMPARATIVE STUDY

Model	Accuracy	F1-Score (Weighted)	Key Observations
LSTM	75%	0.746	The LSTM model captures sequential dependencies but struggles with long-term contextual relationships.
Bi-directional LSTM	77%	0.767	It improved over LSTM by processing input sequences in both forward and backward directions, but still limited in handling complex context.
XLM-RoBERTa (Fine Tuned)	83%	0.826	It improved over Bi-directional LSTM by introducing the self-attention for the long sequence.
XLM-RoBERTa (Fully Connected)	91%	0.908	This model achieves superior performance because of its pre-trained multilingual contextual embeddings and fine-tuning.

*Table 3: Comparative Study*

The *Table 3* compares the performance of three models—LSTM, Bi-directional LSTM, and XLM-RoBERTa—on a text classification task, focusing on their accuracy, weighted F1-score, and key observations. The LSTM model achieves an accuracy of 75% with a weighted F1-score of 0.746. While it effectively captures sequential dependencies in text, it struggles to maintain long-term contextual relationships, limiting its overall performance. The Bi-directional LSTM improves upon the basic LSTM by processing input sequences in both forward and backward directions, allowing it to better understand context from both sides. This enhancement leads to a slight performance boost, achieving 77% accuracy and an F1-score of 0.767. However, it still faces challenges in handling complex contextual relationships. In contrast, XLM-RoBERTa significantly outperforms both LSTM variants, achieving 91% accuracy and a weighted F1-score of 0.908. This superior performance is attributed to its pre-trained multilingual contextual embeddings and fine-tuning capabilities, which enable it to capture intricate language nuances effectively. The results highlight the advantage of transformer-based models like XLM-RoBERTa over traditional recurrent neural networks for tasks requiring deep contextual understanding. The model and accuracy comparative study is shown in bar chart *Figure 9*.

So, we can say using of pre-trained transformer-based models like XLm-RoBERTa over traditional models such as LSTM and Bi-directional LSTM. The ability to process long-range dependencies of XLM-RoBERTa makes it a superior choice for my task.

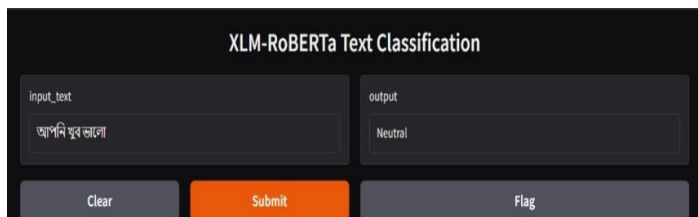


*Figure 9: Comparative Study Bar-chart*

## CHAPTER 6

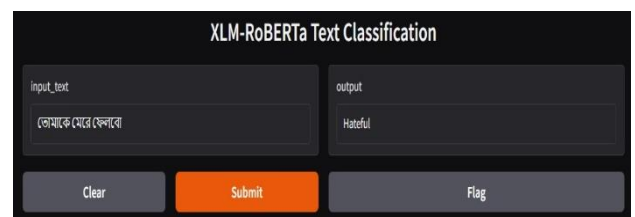
### CONCLUSION

In conclusion, this project demonstrates the effectiveness of using pre-trained transformer-based models, specifically XLM-RoBERTa, for hate speech detection on social media platforms. Through extensive experimentation, it was found that XLM-RoBERTa significantly outperforms traditional models like LSTM and BiLSTM in terms of accuracy and F1-score, achieving a remarkable F1-score of 0.908. The model's ability to process long-range dependencies and leverage multilingual contextual embeddings allows it to capture nuanced features of hate speech that are often missed by sequential models. The fine-tuning process further enhanced its performance, proving the importance of adapting pre-trained models to specific tasks. This project highlights the advantages of transformer models in handling complex NLP tasks, particularly in detecting harmful content across diverse datasets, and sets a strong foundation for future advancements in automated hate speech detection. Finally, real time implementation presents the feasibility and applicability of the proposed technique as it successfully performs the detection of hate speech with high accuracy. The demo of real time implementation are shown in *Figure 10 (Bengali)* and *Figure 11 (English)*.

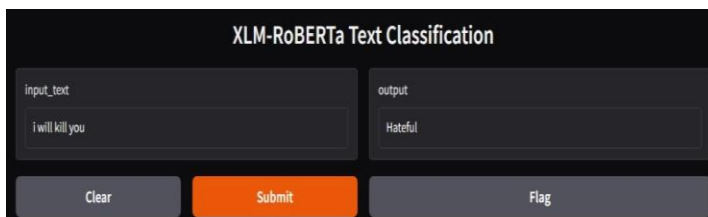


The screenshot shows a web interface titled "XLM-RoBERTa Text Classification". It has two input fields: "input\_text" containing the Bengali text "আপনি খুব ভালো" and "output" containing the classification "Neutral". Below the input fields are three buttons: "Clear", "Submit", and "Flag".

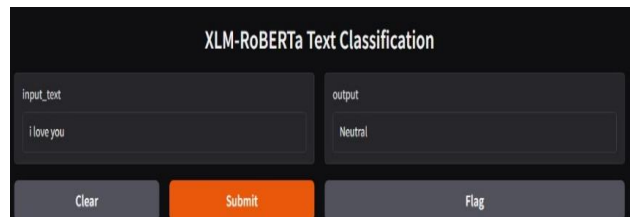
*Figure 10: Real-Time Implementation (Bengali)*



The screenshot shows a web interface titled "XLM-RoBERTa Text Classification". It has two input fields: "input\_text" containing the Bengali text "কোমকে মেরে ফেলা হবে" and "output" containing the classification "Hateful". Below the input fields are three buttons: "Clear", "Submit", and "Flag".



The screenshot shows a web interface titled "XLM-RoBERTa Text Classification". It has two input fields: "input\_text" containing the English text "i will kill you" and "output" containing the classification "Hateful". Below the input fields are three buttons: "Clear", "Submit", and "Flag".



The screenshot shows a web interface titled "XLM-RoBERTa Text Classification". It has two input fields: "input\_text" containing the English text "i love you" and "output" containing the classification "Neutral". Below the input fields are three buttons: "Clear", "Submit", and "Flag".

*Figure 11: Real-Time Implementation (English)*



## **CHAPTER 7**

### **FUTURE SCOPE**

The future scope of this project lies in several exciting directions that could further enhance the accuracy and applicability of hate speech detection models. One potential area for improvement is the exploration of domain-specific fine-tuning, where models can be trained on specialized datasets from different social media platforms or specific languages to address the variations in hate speech across contexts. Additionally, integrating multimodal data, such as images, videos, or user metadata, could provide a more holistic understanding of harmful content by incorporating visual and behavioral cues. Expanding the scope to detect not just hate speech, but also subtle forms of toxicity, microaggressions, and misinformation, would broaden the model's real-world applicability. Furthermore, research into reducing model biases and improving interpretability could enhance trust and transparency in automated systems. Finally, deploying these models in real-time systems with low-latency processing could enable more immediate interventions, creating a safer online environment.

## REFERENCE

- [1] Saksesi, Arum Sucia, Muhammad Nasrun, and Casi Setianingsih. "Analysis text of hate speech detection using recurrent neural network." In 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), pp. 242-248. IEEE, 2018.
- [2] Alrehili, Ahlam. "Automatic hate speech detection on social media: A brief survey." In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), pp. 1-6. IEEE, 2019.
- [3] Mittal, Utkarsh. "Detecting Hate Speech Utilizing Deep Convolutional Network and Transformer Models." In 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM), pp. 1-4. IEEE, 2023.
- [4] Pariyani, Bhavesh, Krish Shah, Meet Shah, Tarjni Vyas, and Sheshang Degadwala. "Hate speech detection in twitter using natural language processing." In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 1146-1152. IEEE, 2021.
- [5] Sajjad, Muhammad, Fatima Zulifqar, Muhammad Usman Ghani Khan, and Muhammad Azeem. "Hate speech detection using fusion approach." In 2019 International Conference on Applied and Engineering Mathematics (ICAEM), pp. 251-255. IEEE, 2019.
- [6] Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan et al. "Opt: Open pre-trained transformer language models." *arXiv preprint arXiv:2205.01068* (2022).
- [7] Aluru, Sai Saketh, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. "Deep learning models for multilingual hate speech detection." *arXiv preprint arXiv:2004.06465* (2020).
- [8] Mozafari, Marzieh, Reza Farahbakhsh, and Noel Crespi. "A BERT-based transfer learning approach for hate speech detection in online social media." In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019* 8, pp. 928-940. Springer International Publishing, 2020.