

M.Sc. in Computer Science and Engineering Thesis

A Multilingual NLP Model for Cyberbullying Detection in Bangla, English, and Romanized Bangla

Submitted by

Faria Afroz
ID:M240105034

Supervised by

Professor Dr. Mohammed Nasir Uddin



**Department of Computer Science and Engineering
Jagannath University**

Dhaka, Bangladesh

November 2025

CERTIFICATION

This thesis titled, “**A Multilingual NLP Model for Cyberbullying Detection in Bangla, English, and Romanized Bangla**”, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree M.Sc. in Computer Science and Engineering in November 2025.

Group Members:

Faria Afroz

Supervisor:

Professor Dr. Mohammed Nasir Uddin

Professor

Department of Computer Science and Engineering

Jagannath University

ACKNOWLEDGEMENT

In particular, we express our sincere gratitude to our advisor, Professor Dr. Mohammed Nasir Uddin, for the continuous support of our research and her patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of research and writing of this project. We could not imagine having a better advisor and mentor for my project.

We also thank our honorable teachers for their encouragement, insightful comments, and tough questions.

Last but not least, we also wish to acknowledge the help provided by the CSE department students who helped us by giving their valuable information for our project.

Dhaka

November 2025

Faria Afroz

Contents

<i>CERTIFICATION</i>	i
<i>ACKNOWLEDGEMENT</i>	ii
List of Figures	v
List of Tables	vi
List of Algorithms	vii
ABSTRACT	viii
1 Introduction	1
1.1 Objectives	2
1.2 Contributions	3
1.3 Justification	4
1.4 Background	5
2 Methodology	6
2.1 Feature extraction phase	7
2.2 Evaluation Metrics	8
2.2.1 Classification phase	9
2.2.2 Logistic regression (LR) algorithm	9
2.2.3 Naïve Bayes (NB) algorithm	9
2.2.4 Support vector machine (SVM) algorithm	10
2.2.5 K-Nearest Neighbor (KNN) algorithm	10
2.2.6 Recurrent Neural Network (RNN) algorithm	10
2.2.7 Long Short-Term Memory (LSTM) algorithm	11
3 Related work	13
3.1 Related work	13
3.2 Machine Learning Models	16
3.2.1 Logistic Regression	16
3.2.2 Naive Bayes	17

3.2.3	Linear SVM	18
3.2.4	Passive-Aggressive	19
3.3	Deep Learning Models	20
3.3.1	BiLSTM	20
3.3.2	Simple RNN	21
3.3.3	LSTM Model	22
3.3.4	1D CNN Model	23
3.3.5	Roc Curves-	24
3.3.6	Computing environment	24
3.3.7	Model Accuracy in Deep Learning	26
4	Index Creation	27
5	<i>k</i>-safe Labeling of Petersen Graph	28
References		29
Index		30
A	Algorithms	31
A.1	Sample Algorithm	31
B	Codes	32
B.1	Sample Code	32
B.2	Another Sample Code	33

List of Figures

2.1 model	6
3.1 Logistic regression report	16
3.2 Logistic regression confusion matrix	16
3.3 naive bayes classification report	17
3.4 naive bayes confusion matrix	17
3.5 linear classification report	18
3.6 linear confusion matrix	18
3.7 passive classification report	19
3.8 passive confusion matrix	19
3.9 deep learning classification report	20
3.10 deep learning confusion matrix	20
3.11 Rnn classification report	21
3.12 Rnn confusion matrix	21
3.13 lstm classification report	22
3.14 lstm confusion matrix	22
3.15 cnn classification report	23
3.16 cnn confusion matrix	23
3.17 ROC Curves DL	24
3.18 Accuracy (DL)	26
3.19 Accuracy (ML)	26

List of Tables

2.1 Example of applying NLP techniques to an English cyberbullying tweet.	12
2.2 Example of applying NLP techniques to a Bangla cyberbullying comment.	12
2.3 Example of applying NLP techniques to a Romanized Bangla cyberbullying comment.	12

List of Algorithms

1	Calculate $y = x^n$	31
---	---------------------	----

ABSTRACT

Thesis abstract

Chapter 1

Introduction

Cyberbullying involves online harassment, threats, shame, and embarrassment. Digital communication methods are used intentionally in cyberbullying [1]. Due to the rapid growth of social media platforms, negative behavior can spread faster and reach more people than offline bullying. Facebook, YouTube, Instagram, and Twitter (X) are the most popular platforms for online harassment, especially among young people, according to studies. This is especially true for younger users. More than 80 million Bangladeshis use the Internet and more than 90 percent of them use Facebook. Most of these people are teens and young adults, which makes them vulnerable to online harassment and abuse. Due to Unicode's growing recognition and Bangladesh's high smartphone and internet penetration, Bangla social media communication is rising. Bangla ranks sixth in global language use. Online language patterns are more casual, loud, and mixed, as Bangladeshi users often speak Bangla, English, and Romanized Bangla ("tumi baje kotha bolcho," "onek beshi annoying," and "ami tomake hate kori"). Despite Bangla's rich linguistic legacy, cyberbullying research in Bangla lags behind English. The biggest challenges include a lack of annotated Bangla datasets, lexicons, morphological and sentiment analysis tools, and deep learning resources for the processing of Bangla natural language. Most cyberbullying detection technologies, such as keyword filters, manual moderation, and rule-based algorithms, are ineffective on social networks because of casual spellings, code-mixing, slang, sarcasm, emojis, and dynamic language patterns. A recent study shows that machine learning and sentiment analysis are better at identifying harmful or offensive content than manual methods and keyword-based methods. However, these machine learning models work only for English content and not for Bangla or multilingual content. To effectively identify Bangla cyberbullying, socio-emotional factors, cultural context, and user-specific variables such as age, gender, and location are important. Support Vector Machine (SVM), which performs well on English datasets, may perform poorly on Bangla due to linguistic structure variations. In some circumstances, Naïve Bayes or Bangla-tuned transformers may outperform these algorithms. This project aims to create a cyberbullying detection algorithm for Bangla and Romanized Bangla social media.

content, taking into account these challenges. Several machine learning classifiers have been trained to identify bullying tendencies in mixed-script Bangla social media posts. User-specific demographic data improve detection accuracy. Performance comparisons are given. Section V concludes the study and suggests further investigation.

1.1 Objectives

Objectives The primary goal of this research is to develop an accurate and scalable cyberbullying detection model that can classify harmful content in Bangla, English, and Romanized Bangla with high efficiency. This system will be able to automatically identify and reject inappropriate comments in social media platforms, ensuring that offensive content is flagged before being posted. The research integrates text analytics and machine learning techniques to address the following specific objectives:

- Section I: Develop a robust cyberbullying detection model capable of accurately classifying comments in Bangla, English, and Romanized Bangla, particularly in the context of code-mixed language commonly used on social media.
- Section II: Compare the performance of various machine learning (ML) models (such as Random Forest, Logistic Regression, SVM, Naïve Bayes, and Passive Aggressive) alongside deep learning (DL) models (including RNN, LSTM) and transformer-based models (like BERT and RoBERTa) for cyberbullying detection in multilingual code-mixed datasets. The evaluation will focus on Bangla and Romanized Bangla text, assessing each model's ability to accurately classify cyberbullying content across these languages.
- Section III: Evaluate and improve domain-specific preprocessing techniques, such as morphological normalization and grammatical structure-based stemming, to improve model performance on Bangla and Romanized Bangla datasets.
- Section IV: Assess the scalability, efficiency, and real-time applicability of the proposed detection model, ensuring that it can operate effectively in high-volume, real-world social media environments.

1.2 Contributions

This research contributes to the field of cyberbullying detection in several key ways:

Comprehensive Evaluation of Language Models: This study rigorously evaluates the performance of various machine learning (ML) and deep learning (DL) models for cyberbullying detection across Bangla, English, and Romanized Bangla, addressing the challenges posed by code-mixed and informal language commonly used on social media platforms.

Benchmarking and Dataset Evaluation: A detailed analysis of a labeled Bangla social media dataset is provided, evaluating factors such as class balance, linguistic diversity, and annotation consistency, and how these properties affect model performance and generalization.

Improvement in Preprocessing Techniques: The study proposes innovative preprocessing techniques, including stemming based on grammatical structure and morphological normalization tailored to Bangla and Romanized Bangla, to improve the performance of the model in real-world applications.

Real-Time Feasibility and Efficiency: The study provides an in-depth evaluation of the scalability, processing time, and resource usage of various models, offering practical insights into which models balance accuracy and efficiency for deployment on social media platforms.

Practical Cyberbullying Prevention Framework: The research contributes a modern multilingual evaluation framework for cyberbullying detection, integrating transformer-based models and language normalization techniques tailored for Bangla and Romanized Bangla.

Future research on developing inclusive, resource-efficient, and explainable AI systems that can comprehend the linguistic and cultural subtleties found in diverse online communities is guided by the findings.

1.3 Justification

The increasing prevalence of cyberbullying on social media platforms is a pressing issue, particularly in multilingual environments like Bangla and Romanized Bangla, where informal language, slang, and code-mixing complicate the detection process. While significant strides have been made in detecting harmful content in English, most research has not focused on Bangla or Romanized Bangla, which introduces unique challenges due to linguistic, cultural, and structural differences. Current approaches—relying on keyword matching, manual moderation, and sentiment analysis—often fail to identify subtle forms of bullying, such as sarcasm or indirect insults, especially in code-mixed and informal language.

The gap in Bangla-specific resources, such as labeled datasets and preprocessing tools, makes it difficult to apply existing machine learning or deep learning models to Bangla and Romanized Bangla. Additionally, Romanized Bangla introduces further complexity, as it does not follow standardized transliteration rules, leading to inconsistency in text representation. This study is crucial as it addresses these gaps by developing a tailored detection model that works effectively with Bangla, Romanized Bangla, and English.

Moreover, the effectiveness of deep learning models like BERT and RoBERTa for detecting cyberbullying in low-resource languages like Bangla is still underexplored. By evaluating transformer-based models alongside traditional machine learning models, this study seeks to determine the most accurate and computationally efficient models for cyberbullying detection in multilingual and code-mixed content.

The practical applications of this research are significant, as it provides tools for improving content moderation on social media platforms, enhancing safety and preventing harassment, especially for vulnerable groups like adolescents

1.4 Background

A lot of research has been done on English text categorization and cyberbullying detection using text mining and NLP [1,13]. Initial study focused on linguistic patterns and contextual cues to classify user posts and chats. Using N-gram annotation and TF-IDF weighting, Yin, Xue, and Hong [7] classified texts using supervised learning. Dinakar, Reichart, and Lieberman [5] examined many supervised learning algorithms for offensive and bullying content. They employed binary and multi-class categorization on manually categorized YouTube comments. Kelly Reynolds [6] studied the Decision Tree (J48) and k-Nearest Neighbor (KNN) methods, but other research has found the Support Vector Machine (SVM) to be one of the most efficient text categorization classifiers due to its excellent management of high-dimensional feature spaces. As stated in [11], SVM has a solid theoretical foundation and solves many text categorization difficulties. Zhijie et al. [10] found that SVM outperformed Naïve Bayes (NB) and KNN in English-language datasets.

Despite these achievements, linguistic, syntactic, and morphological differences cause algorithms to perform differently with non-English data. Numerous studies have adapted text classification methods to South Asian and regional languages, highlighting these challenges. A recent study found Naïve Bayes (NB) effective at classifying Indian language text [12]. In [19], a hybrid Naive Bayes-Ontology-based classification technique improved text classification, but [12] showed that SVM outperformed Naive Bayes in English language classification. ANNs outperformed Vector Space Models (VSM) . This emphasizes the need to represent languages uniquely. Decision Trees, Neural Networks, and N-gram-based representations might help grasp non-English languages with many forms.

Despite NLP's global popularity, little research has been done on Bangla text or Romanized Bangla's difficulties. Bengalis often write Bangla words in English letters on social media ("tui bhalo na," "amar mon kharap"). Transliteration-based communication has spelling, grammar, and word construction issues that make it harder for computers to process than Bengali writing.[07]

Due to these limitations, this study compares multiple Machine Learning (ML) and Deep Learning (DL) algorithms for detecting cyberbullying in Bangla, English, and Romanized Bangla social media content. This study addresses morphological variation, spelling normalization, and contextual ambiguity to find the best algorithm for detecting multilingual and code-mixed cyberbullying in real-world online contexts.

Chapter 2

Methodology

The proposed cyberbullying detection model seeks to identify instances of cyberbullying in social comments and postings by utilizing natural language processing techniques in conjunction with machine learning classifiers. The dataset was preprocessed employing NLP techniques prior to its utilization in training the machine learning classifiers. Finally, the performance assessment employs metrics such as accuracy and precision.

recall and F1-score metrics. Fig. 1 illustrates the proposed cyberbullying detection model, which consists of four phases: the preprocessing phase, the feature extraction phase, the classification phase, and the evaluation phase.[2.1](#)

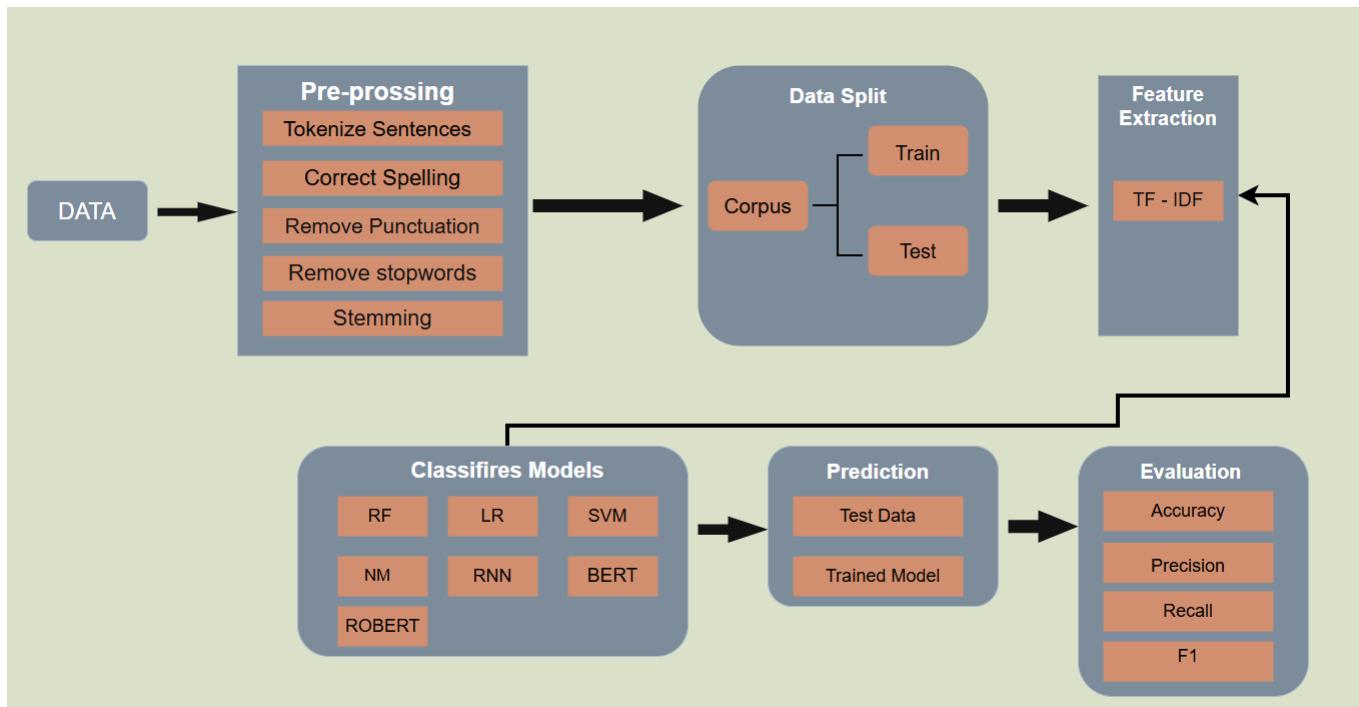


Figure 2.1: model

Prior to being entered into the models, data must be preprocessed. The steps involved in data pretreatment are described in this section:

- **Data purification:** uses regular expressions to eliminate punctuation, stopwords, low-length data, and unnecessary symbols.
- **Tokenization:** Uses the Keras tokenizer function to transform preprocessed text into a vector of unique numbers.
- **Padding:** For model input, it generates vectors of equal length and a vectorized padded text sequence.
- **Normalization:** Converts data onto a standard scale to expedite training and lower the likelihood of becoming trapped in local optima.
- **TF-IDF:** Each term in a document is weighted based on its frequency within that document and its inverse frequency across the entire corpus. In our experiment, we established a vector size of 1065 based on the unique words present in our corpus.

2.1 Feature extraction phase

Similar characteristics have been used to detect cyberbullying occurrences and fine-grained types of text [?]. To summarize, several frequently used word representation techniques have been shown to increase classification accuracy [?], such as Term Frequency (TF) [?] and Term Frequency–Inverse Document Frequency (TF-IDF) techniques [?]. The text is converted into numbers for machine learning classifiers to use them. To calculate term frequency (TF), a text's total number of words is divided by how frequently each term appears. TF is computed using Eq. (2.1) [?].

$$TF = \frac{\text{Frequency of a particular word in the document}}{\text{Total number of words in the document}} \quad (2.1)$$

The *IDF* estimates the weights of essential terms inside a document, as defined by Eq. (2.2) [?].

$$IDF = \log_2 \left(\frac{\text{Total documents}}{\text{Documents with a particular term}} \right) \quad (2.2)$$

Finally, the *TF-IDF* may be determined by multiplying the term and inverse document frequencies, yielding normalized weights. This calculation is performed using Eq. (2.3) [?].

$$TF-IDF = IDF \times TF \quad (2.3)$$

2.2 Evaluation Metrics

- Accuracy:** The accuracy score is determined by the percentage of texts in the dataset that were correctly classified. This is how the accuracy is calculated:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.4)$$

- Precision:** Precision represents the ratio of correct positive predictions to the total number of positive predictions made. The formula for calculating precision is as follows:

$$Precision = \frac{TP}{TP + FP} \quad (2.5)$$

- Recall:** Recall is the ratio of actual positive instances that are correctly identified as positive. The formula for calculating recall is as follows:

$$Recall = \frac{TP}{TP + FN} \quad (2.6)$$

- F1 Score:** The F1 score is a metric derived by calculating the harmonic mean of precision and recall. The formula for the F1 score is as follows:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

- Error Rate (E):** This is the percentage of misclassified instances. It can be calculated as:

$$Error\ Rate(E) = \frac{FP + FN}{N} \quad (2.8)$$

where N represents the total number of instances used during prediction, which is split from the original dataset. A lower error rate indicates an overperforming and more effective method for detecting cyberbullying on social media platforms.

- Inference Time ($t_{instance}$):** The inference time is the average time taken by the model to make predictions on a single cyberbullying instance. The formula for the inference time is as follows:

$$t_{instance} = \frac{T_{end} - T_{start}}{N} \quad (2.9)$$

where N is the total number of examples utilized for prediction, which comes from the original dataset. The prediction procedure starts at T_{start} and ends at T_{end} . A reduced

inference time means that the method for finding cyberbullying on social media sites is faster and more effective.

2.2.1 Classification phase

In this phase, we utilize Machine Learning (ML), a subset of computer science that intersects with artificial intelligence (AI), which simulates human learning using algorithms and data with progressively higher accuracy [?].

According to the proposed model, several ML algorithms have been implemented to classify cyberbullying. These algorithms include Random Forest, Support Vector Machine, Logistic Regression, Naïve Bayes, K-Nearest Neighbor, Recurrent Neural Network, and Long Short-Term Memory (LSTM).

2.2.2 Logistic regression (LR) algorithm

Among the supervised machine learning algorithms, the LR algorithm is one of the most widely used. It forecasts a categorical dependent variable from a given set of independent components [?]. The LR algorithm classifies data utilizing a sigmoid function between 0 and 1 and converts any real value [?]. The function of the sigmoid is presented in Eq. (2.10) [?].

$$S(Z) = \frac{1}{1 + e^{-z}} \quad (2.10)$$

Where ‘ z ’ is the linear combination of input features and their respective weights.

2.2.3 Naïve Bayes (NB) algorithm

The NB algorithm is an essential and effective classification algorithm that enables rapid machine-learning models and predictions. It is a probabilistic classifier based on an object’s probability [?]. It delivers the most significant probability value after computing the likelihood for a given sample using Eq. (2.11) [?].

$$P(h|D) = \frac{P(D|h) \times P(h)}{P(D)} \quad (2.11)$$

This $P(h|D)$ represents the probability of hypothesis h being true given the data D . This $P(D|h)$ is the probability of observing data D given hypothesis h .

2.2.4 Support vector machine (SVM) algorithm

The SVM algorithm is an extensively used supervised learning algorithm for regression and classification tasks. It is most commonly applied to solve machine learning classification problems. The SVM algorithm aims to quickly assign a new data point to the appropriate category by constructing an optimal hyperplane that separates different classes with the maximum possible margin [?]. The decision boundary can be mathematically represented as in Eq. (2.12).

$$f(x) = w^T x + b \quad (2.12)$$

Where w denotes the weight vector and b is the bias term. The SVM seeks to maximize the margin between classes by optimizing w and b .

2.2.5 K-Nearest Neighbor (KNN) algorithm

The KNN algorithm is a simple yet effective non-parametric method used for classification and regression tasks [?]. It classifies a sample based on the majority label among its k nearest neighbors in the feature space. The distance between two samples x_i and x_j is often calculated using the Euclidean distance as shown in Eq. (2.13).

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (2.13)$$

Where n represents the total number of features and $d(x_i, x_j)$ is the distance between samples x_i and x_j .

2.2.6 Recurrent Neural Network (RNN) algorithm

The RNN algorithm is a deep learning model specifically designed for sequential data such as text [?]. It maintains a hidden state that captures information about previous inputs, making it suitable for context-dependent tasks like cyberbullying detection. The hidden state at time t is updated based on the previous state and current input as shown in Eq. (2.14).

$$h_t = f(W_h h_{t-1} + W_x x_t + b) \quad (2.14)$$

Where h_t is the hidden state at time t , W_h and W_x are the weight matrices for the hidden and input layers, respectively, x_t the current input, and b the bias term? The activation function f is

typically a nonlinear function such as tanh ReLU.

2.2.7 Long Short-Term Memory (LSTM) algorithm

The LSTM network is an advanced type of RNN that effectively overcomes the vanishing gradient problem by introducing a memory cell structure and gating mechanisms [?]. It learns long-term dependencies within sequential data and is highly effective for detecting cyberbullying in text where contextual information spans multiple words. The LSTM cell updates are defined as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.15)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.16)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.17)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.18)$$

$$h_t = o_t * \tanh(C_t) \quad (2.19)$$

Where f_t , i_t , and o_t denote the forget, input, and output gates respectively; C_t is the cell state, and h_t is the hidden output at time t . The σ function represents the sigmoid activation.

Table 2.1: Example of applying NLP techniques to an English cyberbullying tweet.

Original Text	@john123 You are so BLACK and white,... Pahahahaha . Common is dumb as fuck! Visit http://example.com . BTW, I have 2 cats.
Tokenizing Sentences	['@john123', 'You', 'are', 'so', 'BLACK', 'and', 'white', ',', '...', 'Pahahahaha', ' ', '.', 'Common', 'is', 'dumb', 'as', 'fuck', '!', 'Visit', ' http://example.com ', '.', 'BTW', ' ', 'I', 'have', '2', 'cats', '.']
Removing Numbers	['@john', 'You', 'are', 'so', 'BLACK', 'and', 'white', 'Pahahahaha', ' ', 'Common', 'is', 'dumb', 'as', 'fuck', 'Visit', ' http://example.com ', 'BTW', 'I', 'have', 'cats']
Removing Punctuation	['@john', 'You', 'are', 'so', 'BLACK', 'and', 'white', 'Pahahahaha', ' ', 'Common', 'is', 'dumb', 'as', 'fuck', 'Visit', ' http://example.com ', 'BTW', 'I', 'have', 'cats']
Removing Stop Words	['@john', 'BLACK', 'white', 'Pahahahaha', ' ', 'Common', 'dumb', 'fuck', 'Visit', ' http://example.com ', 'BTW', 'cats']
Processed Text	['black', 'white', 'pahahahaha', 'common', 'dumb', 'fuck', 'btw', 'cats']

Table 2.2: Example of applying NLP techniques to a Bangla cyberbullying comment.

(Original Text)	@rahim ! ! http://example.com
(Tokenizing)	['@rahim', '', '', '', '', '', '' , '' , '' , 'http://example.com']
(Removing Numbers)	['@rahim', '' , '' , '' , '' , '' , '' , '' , '' , 'httpexamplecom']
(Removing Punctuation)	['rahim' , '' , '' , '' , '' , '' , '' , '' , '' , 'htppexamplecom']
(Removing Stop Words)	['' , '' , '' , '']
(Processed Text)	['' , '' , '' , '']

Table 2.3: Example of applying NLP techniques to a Romanized Bangla cyberbullying comment.

Original Text	@rahim tumi ekdom boka! sobai tomake ghina kore! dekh http://example.com
Tokenizing Sentences	['@rahim', 'tumi', 'ekdom', 'boka', ' ', 'sobai', 'tomake', 'ghina', 'kore', 'dekh', 'http://example.com']
Removing Numbers	['@rahim', 'tumi', 'ekdom', 'boka', ' ', 'sobai', 'tomake', 'ghina', 'kore', 'dekh', 'httpexamplecom']
Removing Punctuation	['rahim', 'tumi', 'ekdom', 'boka', ' ', 'sobai', 'tomake', 'ghina', 'kore', 'dekh', 'httpexamplecom']
Removing Stop Words	['boka', 'sobai', 'ghina', 'dekh']
Processed Text	['boka', 'ghina', 'sobai', 'dekh']

Chapter 3

Related work

3.1 Related work

This section provides an overview of existing research and foundational insights into cyberbullying detection methods in the context of text classification. It outlines the challenges and prerequisites for applying detection techniques to cyberbullying scenarios while examining various approaches. Current advancements, strengths, limitations, and research gaps are highlighted, offering a thorough perspective on the field. Below are key studies conducted on cyberbullying detection across diverse online platforms.

Khan et al. [17] proposed a Bidirectional Long Short-Term Memory (BiLSTM) model for detecting inclusive language in Roman Pashto. This model was compared to traditional machine learning models, including NB, LR, SVM, and RF. The BiLSTM model was trained and tested using three feature-extraction methods: BoW, TF-IDF, and sequence integer encoding. Results indicate that BiLSTM outperformed the other machine-learning models, achieving an accuracy of 97.21%.

Rahman et al. [18] developed a Bangla cyberbullying detection model using a hybrid deep-learning approach. The system utilized CNN and LSTM architectures trained on Facebook and YouTube comments written in Bangla and Romanized Bangla. Their model achieved an F1-score of 93.4%, demonstrating the importance of contextual embedding in low-resource languages.

Obaida et al. [19] proposed a deep-learning method for detecting cyberbullying on social-media platforms using datasets from Twitter, Instagram and Facebook. The LSTM model achieved accuracies of approximately 96.64 %, 94.49 %, and 91.26percent respectively.

Alabdulwahab et al. [20] evaluated several machine-learning and deep-learning algorithms, including KNN, NB, DT, SVM, RF and LSTM, to identify cyberbullying in English social-media posts. Among all models, LSTM achieved the highest accuracy of 96 percent, followed by

SVM at 92 %.

Mehendale et al. [21] designed a hybrid technique for detecting online abusive and bullying messages by combining NLP and machine learning. The model handled both English and Hinglish (Hindi–English code-mixed) datasets using TF-IDF and CountVectorizer for feature extraction. The Random Forest model achieved 97.1% accuracy, outperforming other machine-learning models.

Haque and Akter [22] introduced a Bangla–English multilingual CNN–BiLSTM framework for real-time detection of cyberbullying across social platforms. The model effectively identified offensive and harmful content in both Bangla and Romanized Bangla scripts, achieving 95% accuracy and demonstrating robust cross-lingual performance.

Roy and Mali [23] proposed a deep transfer-learning (2D CNN) model to address image–text-based cyberbullying on platforms such as Facebook and Twitter. Their approach leveraged transfer learning for visual–textual fusion and achieved 89 percent accuracy, showing potential in detecting multimodal cyberbullying content.

Kumar D. [24] presented a hybrid architecture combining a modified DeBERTa transformer model with a Gated Broad Learning System (GBLS) classifier for cyberbullying detection in English text. Evaluated on four benchmark English-language datasets, the model achieved accuracies of 79.3%, 95.41%, 91.37%, and 94.67% respectively. The framework also incorporates explainability mechanisms, including token-level attribution analysis and LIME-based local interpretations, addressing transparency in automated moderation.

Title	Year	Strengths	Weaknesses	Research Gap
Offensive Language Detection for Low-Resource Language Using Deep Sequence Model	2024	ABiLSTM model achieved an accuracy of for offensive language detection in the Roman Bangla language.	Computationally expensive	how to design and develop a deep sequence model with higher performance and lower computational complexity
Offensiveness, Hate, Emotion and GPT: Benchmarking GPT-3.5 and GPT-4 as Classifiers on Twitter-specific Datasets	2024	GPT models can be used as classifiers for hate speech, offensive language, and emotion classification with F1 scores of	Expensive; unable to verify test set contamination; GPT models don't always outperform fine-tuned ones	How to enhance GPT performance to reach or exceed fine-tuned models
Deep Learning Algorithms for Cyberbullying Detection in Social Media Platforms	2024	LSTM achieved accuracy for Instagram, and Facebook datasets	Small dataset size.	How to collect a more extensive dataset.
Cyberbullying Detection using Machine Learning and Deep Learning	2023	CNN+LSTM achieved 96% accuracy in detecting cyberbullying, outperforming KNN and SVM	KNN requires large memory, and SVM is inefficient for datasets.	How to design a lightweight model for cyberbullying tweets.
Cyberbullying detection for Hindi-English language using machine learning	2022	Random Forest achieved 97.1% accuracy in detecting cyberbullying in Hindi-English datasets	Scarcity of labeled data, manual feature extraction, and class imbalance.	How to gather larger balanced data and automate feature extraction.
An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques	2022	CNN+BiLSTM achieved 95% accuracy for multilingual detection.	Language limitations, narrow scope, and lack of contextual understanding.	How to enhance multilingual contextual comprehension.

3.2 Machine Learning Models

3.2.1 Logistic Regression

Accuracy: 0.7811

Classification Report:

	precision	recall	f1-score	support
Approved	0.55	0.81	0.65	4612
Not Approved	0.92	0.77	0.84	13514
accuracy			0.78	18126
macro avg	0.73	0.79	0.75	18126
weighted avg	0.83	0.78	0.79	18126

Figure 3.1: Logistic regression report

Confusion Matrix:

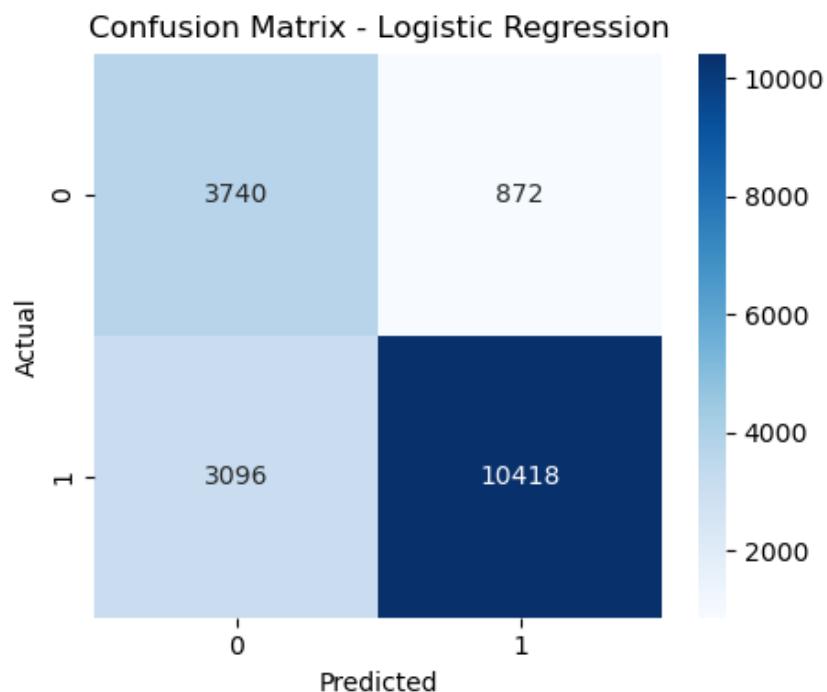


Figure 3.2: Logistic regression confusion matrix

3.2.2 Naive Bayes

Accuracy: 0.8069

Classification Report:

	precision	recall	f1-score	support
Approved	0.68	0.45	0.54	4612
Not Approved	0.83	0.93	0.88	13514
accuracy			0.81	18126
macro avg	0.76	0.69	0.71	18126
weighted avg	0.79	0.81	0.79	18126

Figure 3.3: naive bayes classification report

Confusion Matrix:

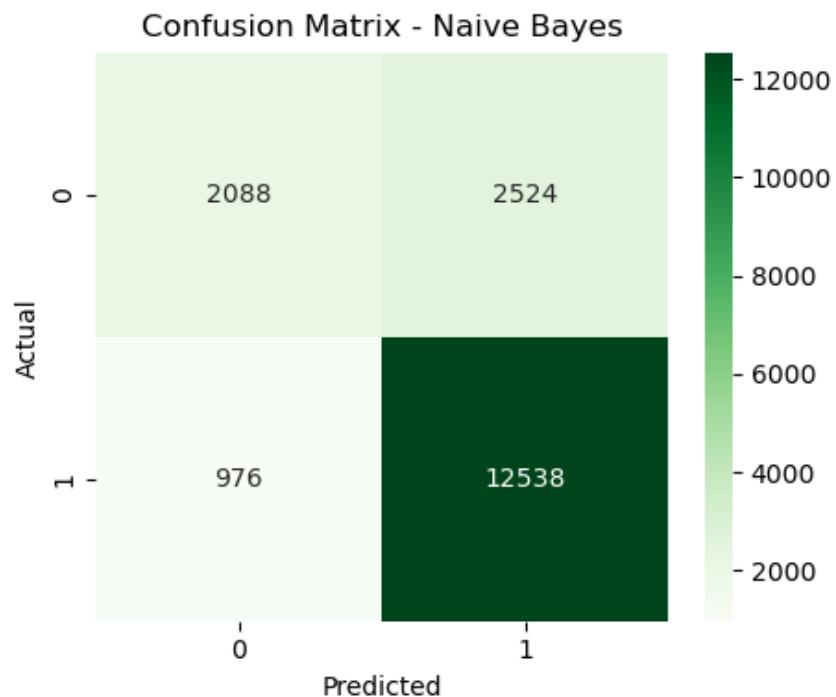


Figure 3.4: naive bayes confusion matrix

3.2.3 Linear SVM

Accuracy:0.7814

Classification Report:

	precision	recall	f1-score	support
0	0.55	0.77	0.64	4612
1	0.91	0.78	0.84	13514
accuracy			0.78	18126
macro avg	0.73	0.78	0.74	18126
weighted avg	0.82	0.78	0.79	18126

Figure 3.5: linear classification report

Confusion Matrix:

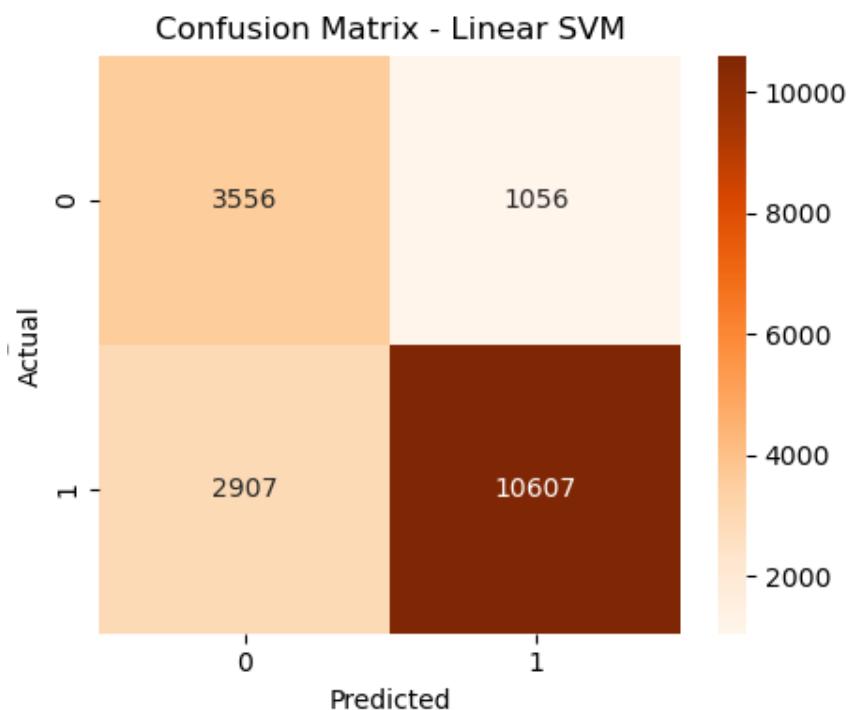


Figure 3.6: linear confusion matrix

3.2.4 Passive-Aggressive

Accuracy:: 0.7987

Classification Report:

	precision	recall	f1-score	support
0	0.61	0.59	0.60	4612
1	0.86	0.87	0.87	13514
accuracy			0.80	18126
macro avg	0.73	0.73	0.73	18126
weighted avg	0.80	0.80	0.80	18126

Figure 3.7: passive classification report

Confusion Matrix:

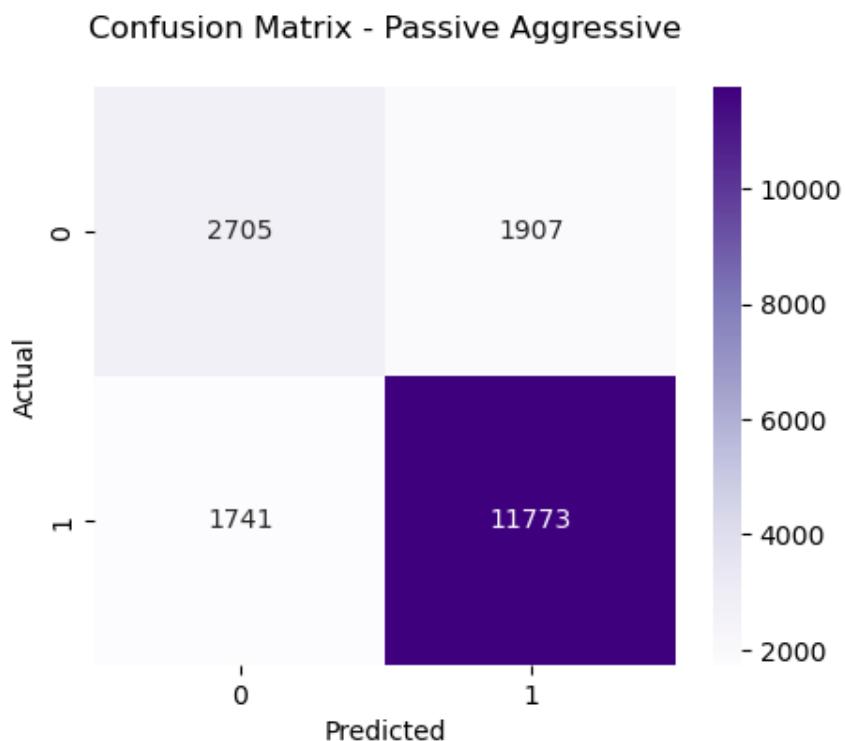


Figure 3.8: passive confusion matrix

3.3 Deep Learning Models

3.3.1 BiLSTM

Accuracy: 0.8711

Classification Report:

Classification Report (BiLSTM):

	precision	recall	f1-score	support
Approved	0.76	0.72	0.74	4612
Not Approved	0.91	0.92	0.91	13514
accuracy			0.87	18126
macro avg	0.83	0.82	0.83	18126
weighted avg	0.87	0.87	0.87	18126

Figure 3.9: deep learning classification report

Confusion Matrix:

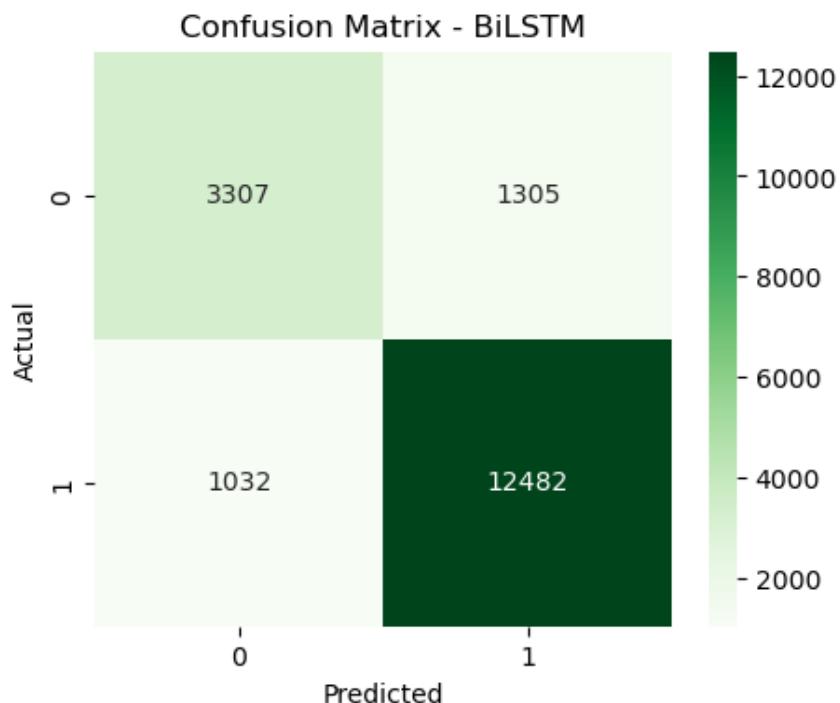


Figure 3.10: deep learning confusion matrix

3.3.2 Simple RNN

Accuracy:: 0.7456

Classification Report:

Classification Report (Simple RNN):

	precision	recall	f1-score	support
Approved	0.00	0.00	0.00	4612
Not Approved	0.75	1.00	0.85	13514
accuracy			0.75	18126
macro avg	0.37	0.50	0.43	18126
weighted avg	0.56	0.75	0.64	18126

Figure 3.11: Rnn classification report

Confusion Matrix:

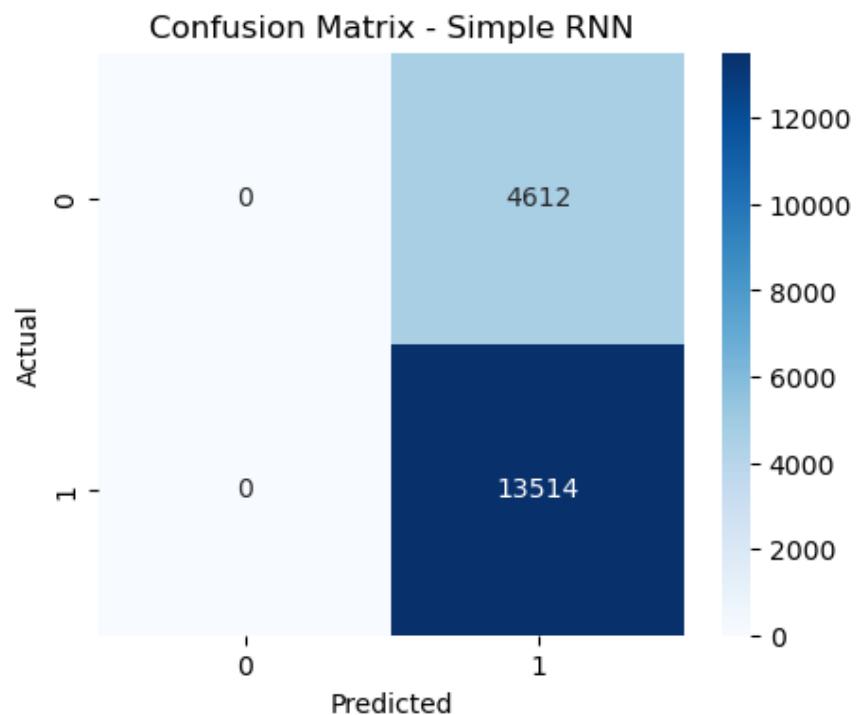


Figure 3.12: Rnn confusion matrix

3.3.3 LSTM Model

Accuracy: 0.7456

Classification Report:

	precision	recall	f1-score	support
Approved	0.00	0.00	0.00	4612
Not Approved	0.75	1.00	0.85	13514
accuracy			0.75	18126
macro avg	0.37	0.50	0.43	18126
weighted avg	0.56	0.75	0.64	18126

Figure 3.13: lstm classification report

Confusion Matrix:

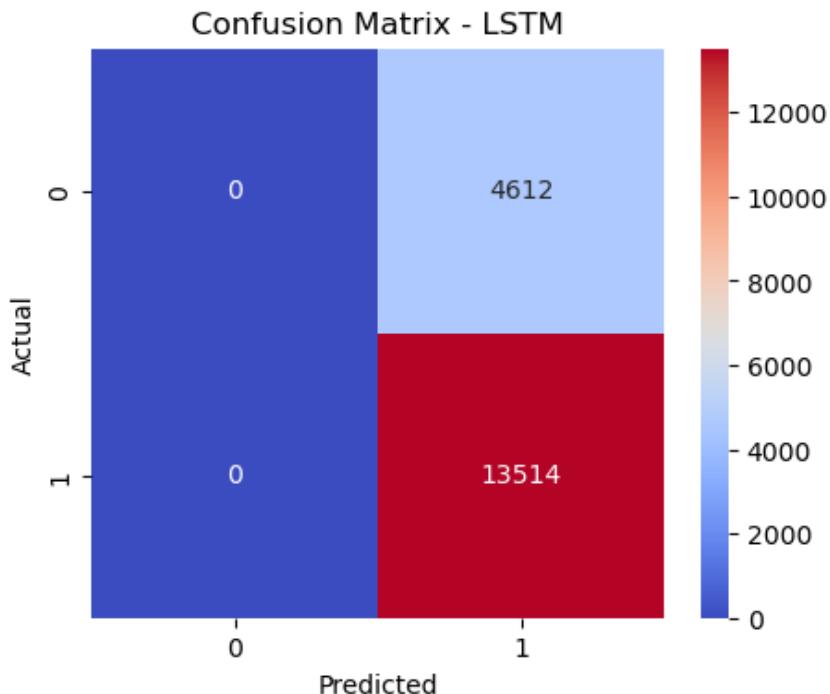


Figure 3.14: lstm confusion matrix

3.3.4 1D CNN Model

Accuracy: 0.8713

Classification Report:

Classification Report (1D CNN):

	precision	recall	f1-score	support
Approved	0.77	0.70	0.73	4612
Not Approved	0.90	0.93	0.92	13514
accuracy			0.87	18126
macro avg	0.84	0.81	0.82	18126
weighted avg	0.87	0.87	0.87	18126

Figure 3.15: cnn classification report

Confusion Matrix:

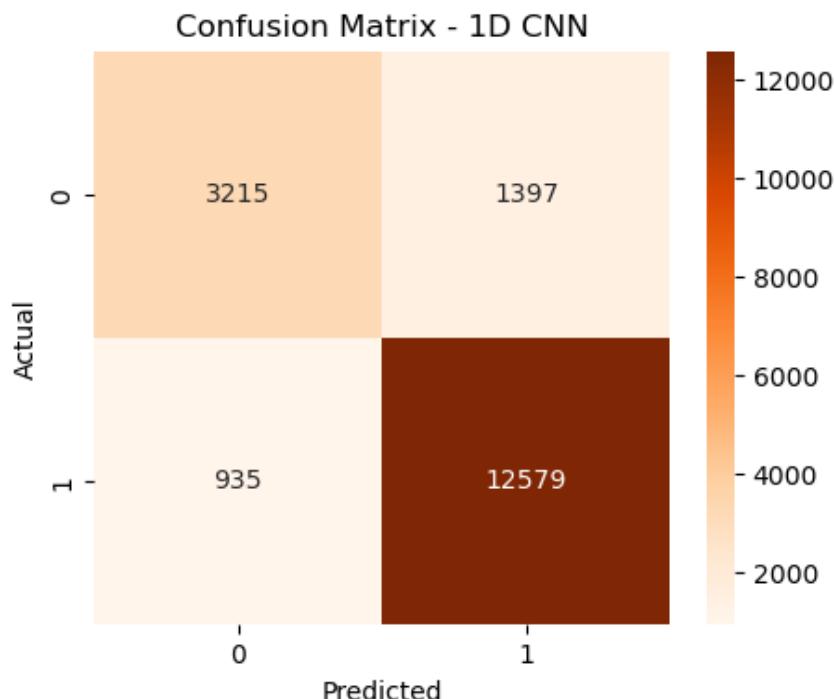
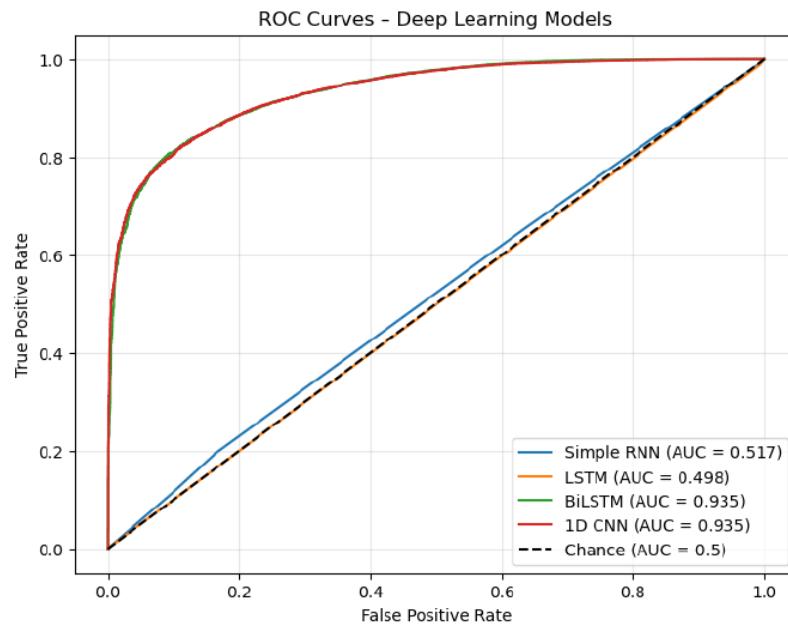


Figure 3.16: cnn confusion matrix

3.3.5 Roc Curves-



BiLSTM and 1D CNN show the strongest performance (AUC ~0.935), clearly separating classes. Simple RNN performs near random (AUC ~0.52), while LSTM underperforms (AUC ~0.50). Overall, BiLSTM and 1D CNN are the best deep models for this task.

Figure 3.17: ROC Curves DL

3.3.6 Computing environment

The experiments described in the proposed model implementation are conducted in a computing environment with the following specifications:

Hardware:

- **Processor:** 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz
- **Memory:** 16 GB RAM
- **Storage:** 1 TB SSD
- **Graphics:** NVIDIA GeForce MX330

Software:

- **Operating System:** Windows 11 Pro 23H2
- **Programming Languages:** Python 3.10.11
- **Libraries and Frameworks:** numpy, pandas, matplotlib, seaborn, sklearn, nltk, contractions.
- **Development Environment:** Jupyter Notebook for interactive code development and visualization.

The implementation results

The experimental results for the proposed model are detailed in this section. The proposed model incorporates feature extraction algorithms into the five Machine Learning classifiers that have been selected. Higher accuracy is the experimentally determined goal of these algorithms. Based on the implementation results, it was found that Random Forest Classifier (RF) achieved higher accuracy by 94.2%, while Support Vector Machine (SVM) classifier achieved accuracy of 93.3%, Logistic Regression (LR) Cla.

3.3.7 Model Accuracy in Deep Learning

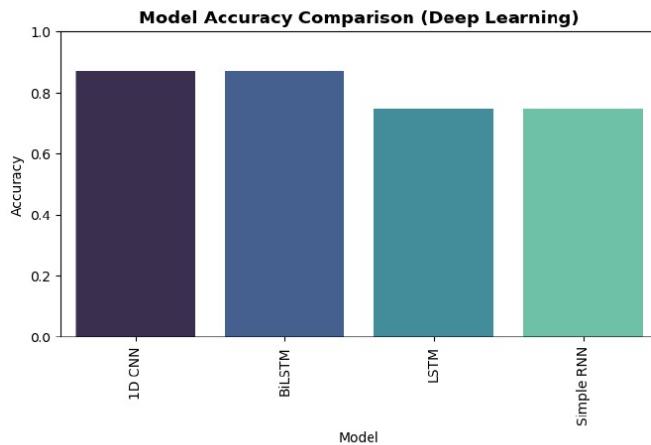


Figure 3.18: Accuracy (DL)

Model Accuracy (ML):

ML Model Comparison:

Model	Accuracy
Naive Bayes	0.8069
Passive-Aggressive	0.7987
Linear SVM	0.7814
Logistic Regression	0.7811

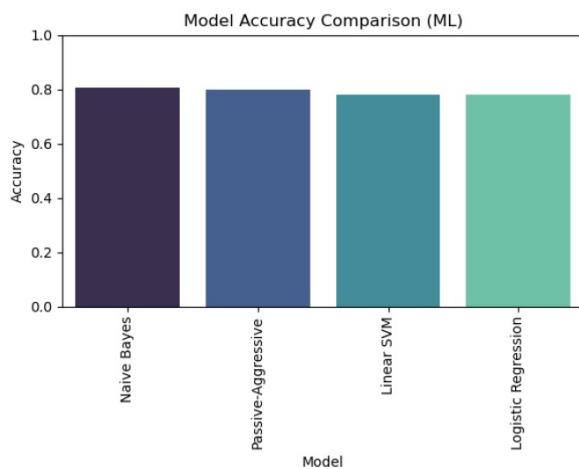


Figure 3.19: Accuracy (ML)

Chapter 4

Index Creation

Here is an example of index creation:

Albert Einstein was a German-born theoretical physicist who developed the theory of relativity, one of the two pillars of modern physics. His work is also known for its influence on the philosophy of science. He is best known to the general public for his mass–energy equivalence formula $E = mc^2$, which has been dubbed the world's most famous equation. He received the 1921 Nobel Prize in Physics for his services to theoretical physics, and especially for his discovery of the law of the photoelectric effect, a pivotal step in the development of quantum theory.

Sir Isaac Newton was an English mathematician, physicist, astronomer, theologian, and author (described in his own day as a natural philosopher) who is widely recognised as one of the most influential scientists of all time, and a key figure in the scientific revolution. His book *Philosophiaæ Naturalis Principia Mathematica* (Mathematical Principles of Natural Philosophy), first published in 1687, laid the foundations of classical mechanics. Newton also made seminal contributions to optics, and shares credit with Gottfried Wilhelm Leibniz for developing the infinitesimal calculus.

Chapter 5

k -safe Labeling of Petersen Graph

In 1898, Petersen produced a trivalent graph with no leaves, now called the Petersen graph [?]. In this chapter we study k -safe labeling for the Petersen graph. We also give upper bound for the span of the Petersen graph. We provide necessary proof for the upper bound.

References

- [1] M. N. Hoque and M. H. Seddiqui, “Detecting cyberbullying text using the approaches with machine learning models for the low-resource bengali language,” *Int J Artif Intell ISSN*, vol. 2252, no. 8938, pp. 358–367, 2024.

Index

1687, *see* Philosophiæ Naturalis Principia Mathematica

Einstein, 27

German-born, 27

Mass-energy equivalence formula , 27

Gottfried Wilhelm Leibniz, 27

Modern physics, 27

Newton, 27

Philosophiæ Naturalis Principia Mathematica, 27

Nobel Prize, 27

Appendix A

Algorithms

A.1 Sample Algorithm

In Algorithm 1 we show how to calculate $y = x^n$.

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

if $n < 0$ **then**

$X \leftarrow 1/x$

$N \leftarrow -n$

else

$X \leftarrow x$

$N \leftarrow n$

end if

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow N/2$

else { N is odd}

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

Appendix B

Codes

B.1 Sample Code

We use this code to find out...

```
1 #include <stdio.h>
2 int Fibonacci(int);
3
4 main()
5 {
6     int n, i = 0, c;
7
8     printf("Enter the value of n: ");
9     scanf("%d", &n);
10
11    printf("\nFibonacci series\n");
12
13    for (c = 1 ; c <= n ; c++)
14    {
15        printf("%d\n", Fibonacci(i));
16        i++;
17    }
18
19    return 0;
20 }
21
22 int Fibonacci(int n)
23 {
```

```

24   if (n == 0)
25     return 0;
26   else if (n == 1)
27     return 1;
28   else
29     return (Fibonacci(n-1) + Fibonacci(n-2));
30 }
```

B.2 Another Sample Code

```

1 SELECT associations2.object_id, associations2.term_id,
2       associations2.cat_ID, associations2.term_taxonomy_id
3 FROM (SELECT objects_tags.object_id, objects_tags.term_id,
4       wp_cb_tags2cats.cat_ID, categories.term_taxonomy_id
5 FROM (SELECT wp_term_relationships.object_id,
6       wp_term_taxonomy.term_id, wp_term_taxonomy.term_taxonomy_id
7 FROM wp_term_relationships
8 LEFT JOIN wp_term_taxonomy ON
9       wp_term_relationships.term_taxonomy_id =
10      wp_term_taxonomy.term_taxonomy_id
11 ORDER BY object_id ASC, term_id ASC)
12 AS objects_tags
13 LEFT JOIN wp_cb_tags2cats ON objects_tags.term_id =
14      wp_cb_tags2cats.tag_ID
15 LEFT JOIN (SELECT wp_term_relationships.object_id,
16       wp_term_taxonomy.term_id as cat_ID,
17       wp_term_taxonomy.term_taxonomy_id
18 FROM wp_term_relationships
19 LEFT JOIN wp_term_taxonomy ON
20       wp_term_relationships.term_taxonomy_id =
21       wp_term_taxonomy.term_taxonomy_id
22 WHERE wp_term_taxonomy.taxonomy = 'category'
23 GROUP BY object_id, cat_ID, term_taxonomy_id
24 ORDER BY object_id, cat_ID, term_taxonomy_id)
25 AS categories on wp_cb_tags2cats.cat_ID = categories.term_id
26 WHERE objects_tags.term_id = wp_cb_tags2cats.tag_ID
27 GROUP BY object_id, term_id, cat_ID, term_taxonomy_id
28 ORDER BY object_id ASC, term_id ASC, cat_ID ASC)
29 AS associations2
30 LEFT JOIN categories ON associations2.object_id =
```

```
31      categories.object_id
32 WHERE associations2.cat_ID <> categories.cat_ID
33 GROUP BY object_id, term_id, cat_ID, term_taxonomy_id
34 ORDER BY object_id, term_id, cat_ID, term_taxonomy_id
```