CYBERBULLYING DETECTION IN EDUCATIONAL PLATFORMS USING MACHINE LEARNING

SOCIALLY RELEVANT MINI PROJECT REPORT

Submitted by

JANAVI SREE R [REGISTER NO:211423104238]

HEMASEN S [REGISTER NO:211423104223]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE,

CHENNAI- 600123.

(An Autonomous Institution Affiliated to Anna University, Chennai)

OCTOBER 2025

BONAFIDE CERTIFICATE

Certified that this project report **“ CYBERBULLYING DETECTION IN EDUCATIONAL PLATFORMS USING MACHINE LEARNING ”**is the bonafide work of “**JANAVI SREE R (211423104238), HEMASEN S (211423104223)”**who carried out the project work under my supervision.

SIGNATURE

DR.T.JACKULIN,M.E.,Ph.d.,
PROFESSOR SUPERVISOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
COLLEGE,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

DR.L.JABASHEELA,M.E.,Ph.d.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
COLLEGE,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the 23CS1512-Socially relevant mini project Viva-Voce Examination held on..........................

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **JANAVI SREE R** (211423104238), **HEMASEN S** (211423104223) hereby Declare that this project report titled "**CYBERBULLYING DETECTION IN EDUCATIONAL PLATFORMS USING MACHINE LEARNING** ", under the guidance of DR.T.JACKULIN, M.E.,Ph.D., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

1. JANAVI SREE R

2. HEMASEN S

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere and hearty thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA M.E.,Ph.D.,**
for the support extended throughout the project.

We would like to thank my Project Guide **DR.T.JACKULIN,M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

JANAVI SREE R

HEMASEN S

ABSTRACT

Nowadays, online educational platforms have become essential tools for communication among students and teachers. However, the rise of such platforms has also increased the risk of cyberbullying, affecting students' mental health and academic performance. A Cyberbullying Detection system aims to automatically identify and prevent harmful or abusive behavior in digital learning environments. This system utilizes **Behavioral-Based Machine Learning** to analyze user text, sentiment, and activity patterns. It combines **Natural Language Processing (NLP)** and **Machine Learning algorithms** to detect offensive or bullying messages effectively. The model employs techniques such as **text preprocessing, sentiment analysis, and behavioral feature extraction** to understand the context of interactions. Classification algorithms like **SVM and Random Forest** are used to categorize messages as bullying or non-bullying. The project is implemented using **Python, Scikit-learn, and NLP libraries**, with datasets collected from educational discussion forums. This work helps to ensure safer online communication, early detection of bullying incidents, and promotes a positive digital learning environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

The pervasive integration of online educational platforms into modern learning environments has introduced unprecedented flexibility and accessibility. However, this digital transformation also brings forth new vulnerabilities, notably the rise of cyberbullying, which poses a significant threat to student safety and well-being. A critical challenge lies in the often-subtle nature of these harmful online interactions, which frequently go undetected by conventional monitoring systems. This project proposes a sophisticated Hybrid Detection Framework specifically designed to address this growing concern. This framework represents an advanced approach by synergistically blending state-of-the-art Deep Learning models, such as RoBERTa for contextual text understanding and BiLSTM for sequential pattern analysis, with established traditional Machine Learning algorithms including Support Vector Machine (SVM), Random Forest, and Logistic Regression. Furthermore, the system innovates by expanding its analytical scope beyond mere textual content to incorporate crucial behavioral indicators—such as excessive capitalization, unusual session patterns, login irregularities, and message frequency—thereby capturing more nuanced forms of harassment. For practical and immediate utility, the framework integrates with a secure ngrok-based interface, furnishing administrators with specialized access links for real-time monitoring of alerts and the capability for prompt manual intervention, ultimately fostering a more secure digital learning environment.

1.2 Problem Definition

 Inadequate Detection of Subtle Cyberbullying: Existing systems often fail to identify non-explicit or nuanced harmful interactions, leaving many incidents undetected.

- **Over-Reliance on Keywords and Text Analysis:** Most platforms depend solely on keyword matching or basic sentiment analysis, which cannot capture context-based or disguised bullying.
- **Vulnerability to Behavioural Circumvention:** Users can bypass text-centric filters by using context-dependent language or behavioural tactics that appear harmless in isolation.
- **Need for a Robust Multi-Faceted Approach:** The project aims to develop a system that detects both explicit textual threats and covert behavioural harassment patterns to enhance student safety and create a secure learning environment.

1.3 Scope

The scope of this project encompasses a comprehensive approach to cyberbullying detection, from data utilization and model development to secure deployment. In terms of data and feature scope, the project will leverage both the rich textual content of communications (e.g., messages, forum posts) and a set of carefully selected behavioural metadata. These behavioural indicators include, but are not limited to, excessive capitalization, unusual session login/logout patterns, irregularities in login frequency, and anomalous message submission rates, which collectively provide a holistic view of user interaction. The model scope involves the design, development, and integration of a sophisticated hybrid classification model. This model will combine the deep contextual understanding capabilities of Deep Learning models (specifically RoBERTa for transformer-based text embeddings and BiLSTM for sequential pattern recognition) with the robust predictive power of traditional Machine Learning algorithms (SVM, Random Forest, Logistic Regression) to process the diverse feature set. Finally, the deployment scope focuses on practical and secure operationalization. This includes implementing a secure ngrok-based administrative interface that provides special, authenticated access links. Through this portal, administrators will gain capabilities for real-time monitoring of detected alerts, access to detailed activity logs, and the crucial ability to manually review and remove offenders, ensuring prompt and decisive action within the online learning environment. The entire framework is specifically tailored for deployment within online digital learning platforms, ensuring its relevance and effectiveness in the target domain.

1.4 Objectives

□ **To develop a robust hybrid detection model:** Design, implement, and evaluate a framework that integrates **Deep Learning models** (RoBERTa for contextual understanding and BiLSTM for sequential analysis) with **traditional Machine Learning algorithms** (SVM, Random Forest, Logistic Regression) for effective cyberbullying detection.

□ **To enhance detection accuracy via behavioural indicators:** Incorporate diverse behavioural features (e.g., excessive capitalization, session patterns, login irregularities, message frequency) to identify subtle and indirect forms of cyberbullying that text-only methods often miss.

□ **To ensure practical and secure deployment:** Provide a secure, user-friendly administrative interface using tools like **ngrok**, enabling real-time alerts, monitoring, and manual review of detected offenders by educational platform administrators.

□ **To promote a safer digital learning environment:** Enable early, accurate, and resource-efficient detection of cyberbullying to proactively protect student well-being and foster a positive, regulated online educational experience.

CHAPTER 2

LITERATURE SURVEY

In recent years, automated cyberbullying detection has become an active research area due to the rise of online interactions in educational platforms. Numerous studies have explored text-based, behaviour-based, and hybrid approaches to improve the accuracy and timeliness of detection, each contributing valuable insights to the literature.

[4] A 2019 study examined traditional machine learning classifiers such as *Support Vector Machines (SVM)* and *Random Forests* using engineered text features like *TF-IDF* and *n-grams*. The results showed reasonable detection accuracy with smaller datasets; however, the models struggled with sarcasm, contextual interpretation, and domain adaptability.

[4] In 2020, researchers shifted toward deep learning architectures including *BiLSTM* and *CNN* models for sequential and sentiment-based text analysis. These models captured contextual cues more effectively than classical methods but required substantial computational resources and large annotated datasets for optimal performance.

[8] A **2021 comparative study** evaluated **transformer-based models** (such as *BERT* and *RoBERTa*) against RNN variants on both social media and educational chat datasets. Pretrained transformers outperformed traditional models on several

benchmarks due to contextual embeddings. Nevertheless, *domain shift* remained a challenge, as models pretrained on general web data did not always generalize well to educational settings.

[10] Another **2021 work** proposed **hybrid models** that combined behavioural features such as posting frequency, time-of-day spikes, and sudden changes in message volume—with textual sentiment features. This integration improved *early-warning capabilities* and reduced false negatives. However, the study emphasized the challenges of collecting labelled behavioural data while maintaining user privacy.

[5] In **2022**, research attention turned to **data imbalance** and **augmentation methods** like *SMOTE* and *back-translation* to enhance the detection of minority bullying instances. While these methods improved recall rates, they sometimes introduced synthetic noise, slightly reducing precision and highlighting the trade-offs inherent in oversampling.

[7] A **2023 study** incorporated **explainable AI (XAI)** frameworks such as *attention visualization* and *SHAP* to make model predictions more interpretable for educators. Though explainability increased trust in automated systems, results showed that explanations could be too technical or misleading when based on spurious correlations.

[8] Further comparisons in **2023–2024** revealed that **RoBERTa** consistently achieved higher *F1 scores* than classical models in benchmark evaluations. The **ETASR 2024** review also highlighted that **hybrid deep learning models** (e.g., *BiGRU with attention*

mechanisms) and **RoBERTa-based architectures** were among the top performers, although dataset heterogeneity and cross-domain adaptation persisted as major issues.

[9] Recent studies from **2023–2024** observed a trend toward **behaviour-aware hybrid systems** integrating non-textual features—such as *caps usage*, *posting bursts*, and *login irregularities*—with textual content models. These systems proved especially useful in multilingual or low-resource contexts but required careful feature design and ethically compliant data access.

[1],[2],[3],[5],[8] Overall, the literature identifies several recurring challenges: *limited and biased datasets*, *poor cross-domain generalization*, *class imbalance*, and *privacy concerns* regarding user behaviour logs. Future directions emphasize the development of **larger and more diverse educational corpora, privacy-preserving learning, robust domain adaptation**, and **human-centred explainable dashboards** for educators to monitor and intervene responsibly

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing Systems

The current primary challenge with existing systems, as highlighted in the topic, is the undetected nature of subtle harmful interactions in online educational platforms. These subtle forms of cyberbullying frequently go unnoticed when relying solely on typical, rule-based, or simpler detection mechanisms. Traditional detection approaches often depend heavily on text-only analysis (content-based), which can be easily circumvented by users employing nuanced or behavioural forms of harassment (e.g., using capitalization, odd session patterns, or excessive messaging to intimidate). Therefore, the implied existing systems are often text-centric, inaccurate in detecting subtle or behavioural bullying, and lack real-time administrative intervention features.

3.2 Proposed System

The proposed solution is a Hybrid Cyberbullying Detection Framework designed for online educational platforms. Its main purpose is to enable early, accurate, and resource-efficient detection of cyberbullying by moving beyond text-only analysis. The framework achieves this by blending both modern and conventional algorithms: Deep Learning models (RoBERTa, BiLSTM) are combined with traditional Machine Learning algorithms (Support Vector Machine (SVM), Random Forest, Logistic Regression). Crucially, the system expands detection capability by incorporating behavioural indicators such as excessive capitalization, session patterns, login irregularities, and message frequency. The system is deployed via a secure ngrok-based interface that provides special access links for administrators, allowing them to monitor alerts, log in, and manually remove offenders, thereby guaranteeing practical and immediate intervention.

3.3 Feasibility Study

The project appears highly feasible across all critical dimensions, leveraging robust technology to address a significant social problem.

Economic Feasibility

The project is economically feasible, as it aims for resource efficiency and utilizes a hybrid model, suggesting optimized computational costs compared to deploying large-scale, standalone Deep Learning models. The use of a platform like ngrok for the secure interface is cost-effective for initial deployment and testing, minimizing infrastructure costs. The primary economic benefit is the reduction of indirect costs associated with undetected cyberbullying, such as decreased student retention, lower academic performance, and institutional reputational damage, making the system a valuable investment for educational platforms.

Technical Feasibility

The framework is technically feasible because it relies on mature and well-supported algorithms. The integration of RoBERTa and BiLSTM provides state-of-the-art capability for complex text and sequence analysis, while the inclusion of established algorithms like SVM and Random Forest ensures robustness and interpretability. Emphasizing behavioural indicators lessens the dependency on pure language modelling, improving resilience against subtle attacks. The proposed ngrok-based interface provides a viable and secure method for practical deployment and administrative access, ensuring the system can be integrated into existing platform management workflows.

Social Feasibility

The system is socially feasible as it addresses an explicit need for student safety in online educational platforms. By focusing on the subtle nature of harmful interactions, the system directly tackles the challenge of undetected cyberbullying, promising to create a safer and more regulated digital learning environment. The administrative interface, which allows for manual offender removal and alert monitoring, ensures human oversight, fostering trust and accountability within the educational community. The system's success is directly measured by its positive impact on the well-being and security of students.

3.4 Development Environment

Software Requirements

OS: Windows 10/11, Linux, or macOS

Programming Language: Python (latest stable version)

Libraries: Pandas, Scikit-learn, NumPy, Matplotlib, RandomforestClassifier.

IDE/Notebook: Jupyter Notebook / Google Colab / VS Code

Database: CSV files (for MVP) or SQLite/MySQL (if extended)

Version Control: Git/GitHub (optional, for team collaboration)

Hardware Requirements

Processor: Minimum Dual-Core CPU (i3 or higher recommended)

RAM: Minimum 4 GB RAM (8 GB recommended for smooth model training)

Storage: Minimum 2 GB free disk space (for dataset & dependencies)

Internet: Required for research, package installation, and cloud notebooks

Optional: GPU not mandatory (CPU is sufficient for small dataset MVP)

Language: Python 3.8+

CHAPTER 4
SYSTEM DESIGN

4.1 FLOW DIAGRAM

This project requires a dataset which have both images and their caption. The dataset should be able to train the image captioning model.

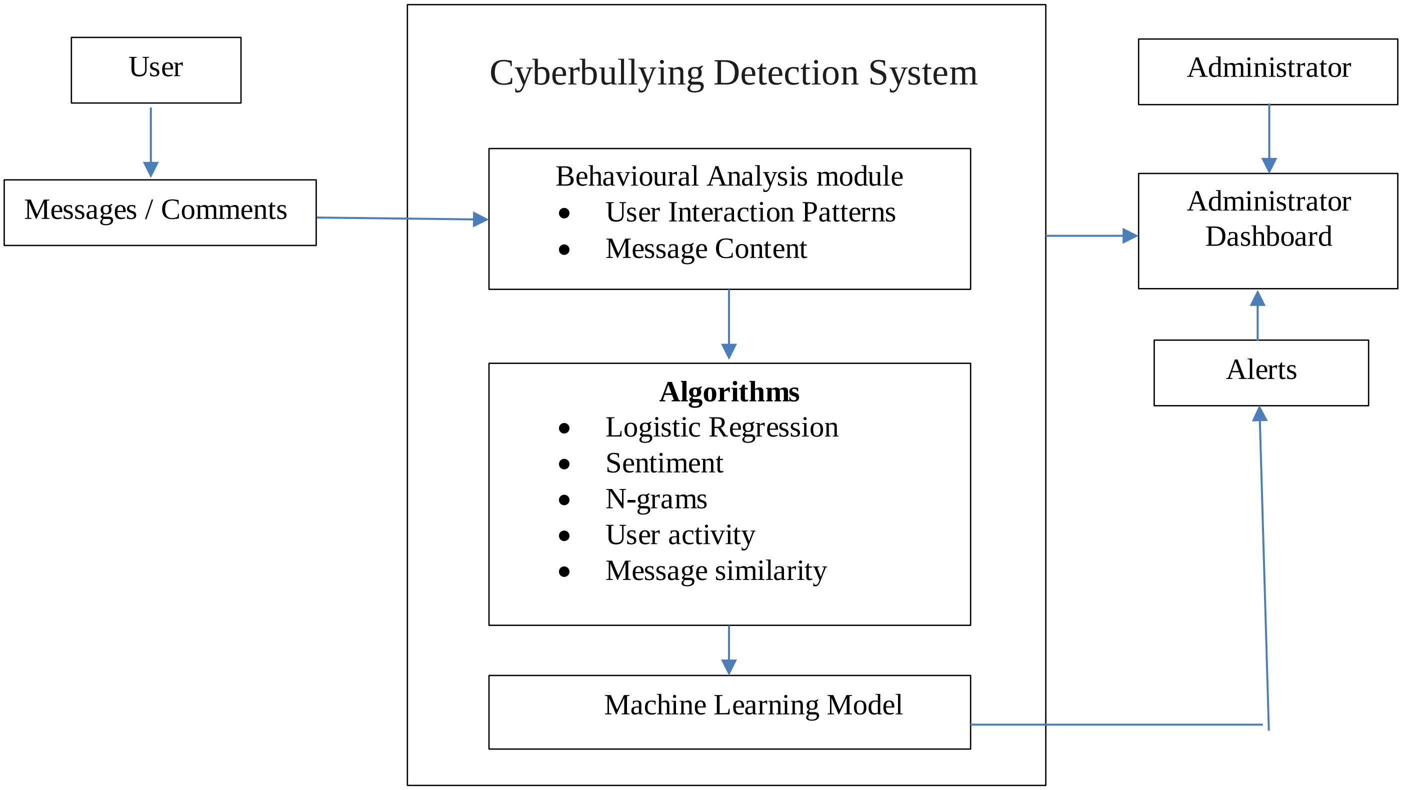


Fig. 4.1 Working flow of the model

4.2 Data Description

The efficacy of the proposed hybrid cyberbullying detection framework hinges upon a rich and multi-modal dataset, meticulously curated from online educational platforms. This dataset comprises two primary categories: **textual data** and

behavioural metadata, each providing unique insights into potentially harmful interactions.

Textual Data: This category encompasses all forms of user-generated communicative content within the online educational platform. It includes, but is not limited to, messages exchanged in chat rooms, forum posts, comments, assignment submissions, and any other textual interactions. The data is typically in raw string format, reflecting the diverse linguistic styles of users, which can range from formal academic discourse to informal conversations, incorporating slang, emojis, and platform-specific jargon. Critically, this textual corpus may contain both overt and subtle instances of harassment, making its nuanced analysis paramount. The volume of such data is expected to be substantial, necessitating robust processing capabilities.

Behavioural Metadata: This crucial category captures quantitative and categorical information derived from system logs, user activity records, and session management databases within the platform. Unlike textual content, behavioural data provides insights into *how* users interact with the platform and each other, often revealing patterns indicative of bullying. Specific indicators include, but are not limited to:

Excessive Capitalization: Quantifying the prevalence or proportion of all-caps text within messages.

Session Patterns: Analysis of login/logout timestamps, frequency of logins over specific intervals, and duration of active sessions.

Login Irregularities: Detection of unusual login locations (e.g., disparate IP addresses), frequent failed login attempts, or atypical login times.

Message Frequency: Measuring the rate at which a user sends messages within defined time windows (e.g., messages per minute or hour). These indicators, often structured or semi-structured, offer a complementary perspective to textual analysis.

Ground Truth/Labels: For the purpose of supervised learning, both textual and behavioural data require associated labels. These labels, typically acquired through rigorous manual annotation, categorize interactions as 'bullying,' 'not bullying,' or potentially more granular categories like 'harassment,' 'insult,' or 'normal.' The accuracy and consistency of these labels are foundational to training effective detection models.

4.3 Data Preprocessing

Data preprocessing is a pivotal stage that transforms raw, heterogeneous data into a clean, consistent, and model-ready format, optimizing it for both Deep Learning and traditional Machine Learning algorithms.

For Textual Data: The preprocessing pipeline for textual content involves several critical steps:

Cleaning: Initial cleaning removes irrelevant elements such as URLs, HTML tags, special symbols, and excessive whitespace. The handling of emojis is context-dependent; they may be removed or replaced with a descriptive token if deemed semantically relevant.

Normalization: All text is typically converted to lowercase to standardize vocabulary and treat word variations uniformly, reducing sparsity.

Tokenization: Text is broken down into smaller units—words or sub-words. For models like RoBERTa, which leverage sub-word tokenization (e.g., Byte-Pair Encoding), this step aligns the text with the pre-trained model's vocabulary.

Stop Word Removal and Lemmatization/Stemming (Conditional): While often used in traditional NLP, these steps are typically applied conditionally or less aggressively for modern Deep Learning models like RoBERTa, which derive meaning from word context and morphology.

Handling Imbalance: As cyberbullying instances are typically rare, strategies such as oversampling the minority class, under sampling the majority class, or employing weighted loss functions during training are essential to prevent model bias towards the dominant 'not bullying' class.

For Behavioural Metadata: Preprocessing for behavioural data focuses on standardization and feature engineering:

Normalization/Scaling: Numerical behavioural features (e.g., message frequency, session duration) are scaled using techniques like Min-Max Scaling or Z-score Standardization. This ensures that features with larger numerical ranges do not disproportionately influence the model.

Handling Missing Values: Missing data points (e.g., incomplete session logs) are addressed through imputation strategies (e.g., mean, median, mode imputation) or by marking them appropriately.

Categorical Encoding: Any categorical behavioural features (e.g., type of login irregularity) are converted into numerical representations using one-hot encoding or label encoding.

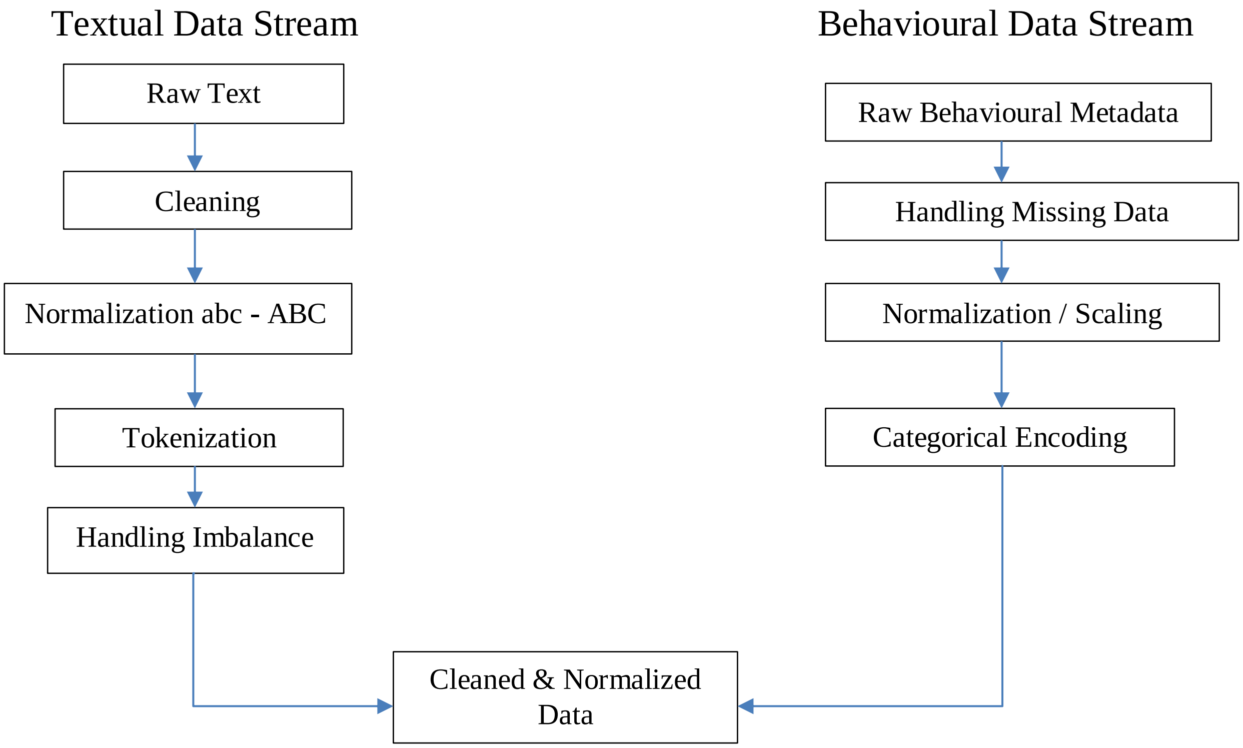


Fig.4.1. Textual and Behavioral Data Preprocessing Flow

4.4 Feature Extraction

Feature extraction is the process of deriving meaningful numerical representations from the pre-processed data, suitable for input into the various models within the hybrid framework. This project employs distinct strategies for textual and behavioural features.

From Textual Data (Deep Learning Features):

Robert Embeddings: After tokenization, the pre-processed text is fed into the fine-tuned Robert model. Robert, a large transformer model, generates rich, contextualized **word embeddings** or a single **sequence embedding** (e.g., the output of the [CLS] token) that encapsulate the semantic and syntactic meaning of the text. These embeddings serve as high-dimensional, dense feature vectors that capture the nuances of language.

BiLSTM Output Features: The contextual embeddings from RoBERTa (or other dense word embeddings) can then be fed into the BiLSTM. The BiLSTM further processes these sequential embeddings, capturing long-range dependencies and temporal context within the message. The final hidden state or an attention-weighted combination of hidden states from the BiLSTM provides another set of powerful, context-aware features.

From Behavioural Metadata (Traditional ML Features):

Direct Features: The pre-processed and scaled behavioural indicators themselves serve as direct numerical features for the traditional ML models. Examples include:

- Normalized count of capitalized words.
- Standardized login frequency.
- Binary flags for unusual login locations.
- Rate of messages per time unit.

Engineered Features: Further features can be engineered from raw behavioural data, such as:

Rate of change: Derivative of message frequency or session duration.

Anomaly scores: Using clustering or isolation forest on login patterns to generate an "anomaly score."

Interaction frequency: How often a user interacts with a specific target.

Hybrid Feature Combination: The final step in feature extraction involves combining these disparate feature sets. The high-dimensional Deep Learning features (from RoBERTa and BiLSTM) are concatenated with the lower-dimensional, engineered behavioural features. This combined feature vector then serves as the comprehensive input for the final classification layer or for the ensemble of traditional ML classifiers, allowing the system to leverage both content and context for highly accurate cyberbullying detection.

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

The proposed cyberbullying detection framework employs a sophisticated, hybrid architecture that synergistically integrates cutting-edge Deep Learning with established Machine Learning methodologies. This design is specifically engineered to address the complex and often subtle nature of online harassment within educational platforms, by meticulously processing both textual content and behavioural metadata. Logically, the system is structured into three interconnected operational phases: the Data Ingestion & Feature Extraction Layer, responsible for transforming raw platform data into actionable feature vectors; the Hybrid Detection Engine, which serves as the core analytical unit for combining diverse predictive signals; and the Administrative Deployment Layer, ensuring secure, real-time administrative oversight and intervention. This multi-layered approach guarantees a comprehensive solution, from initial data capture and intelligent analysis to practical, human-in-the-loop management of detected threats, ultimately enhancing student safety and maintaining a regulated digital learning environment.

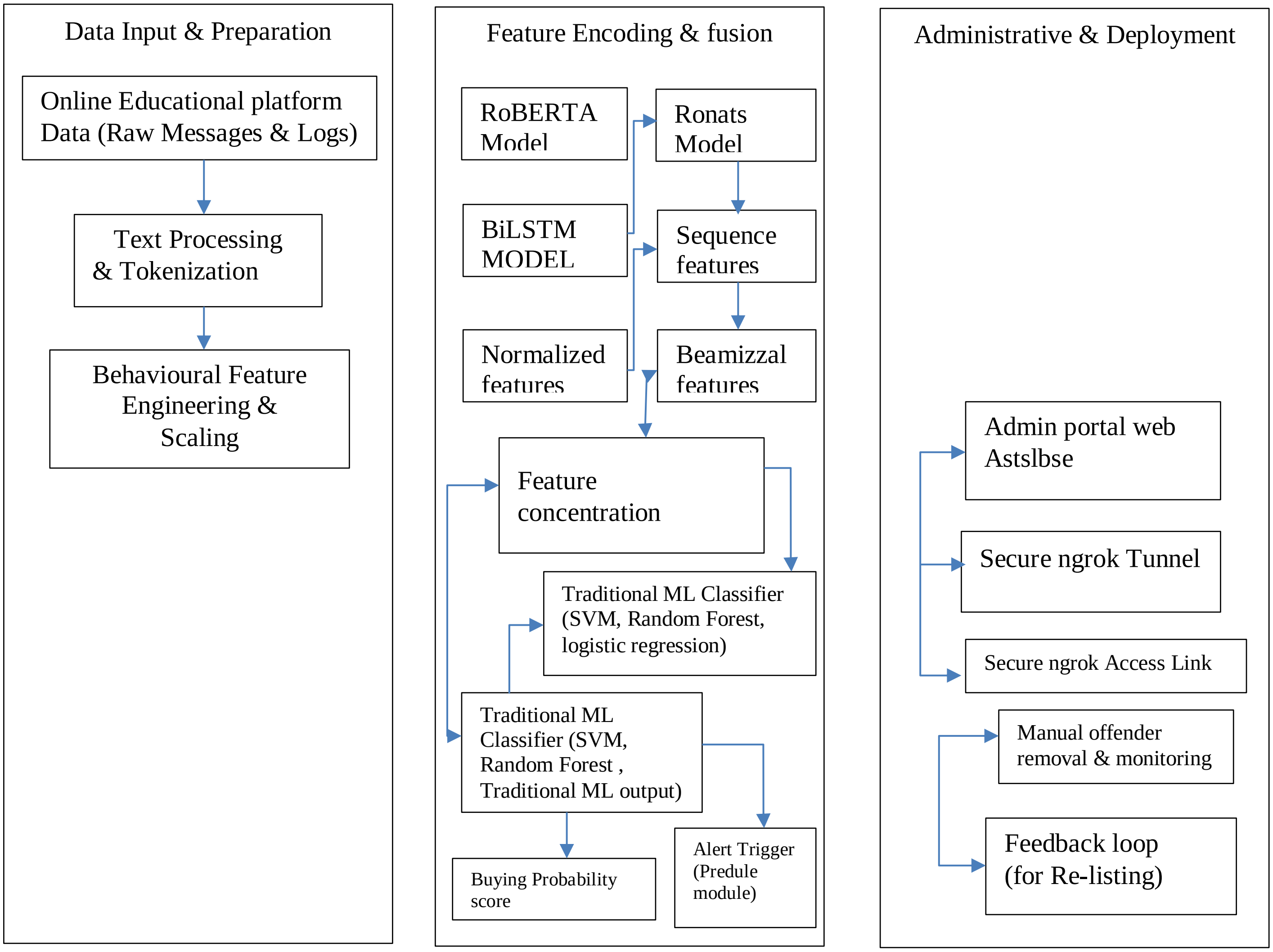

Data Input & Preparation
Online Educational platform Data (Raw Messages & Logs)
Text Processing & Tokenization
Behavioural Feature Engineering & Scaling
Feature Encoding & fusion
RoBERTA Model
Ronats Model
BiLSTM MODEL
Sequence features
Normalized features
Beamizzal features
Feature concentration
Traditional ML Classifier (SVM, Random Forest, logistic regression)
Traditional ML Classifier (SVM, Random Forest , Traditional ML output)
Buying Probability score
Alert Trigger (Predule module)
Administrative & Deployment
Admin portal web Astslbse
Secure ngrok Tunnel
Secure ngrok Access Link
Manual offender removal & monitoring
Feedback loop (for Re-listing)

Fig.5.1. System Architecture.

5.2 Modules

The architecture is composed of several distinct modules, each responsible for a specific function within the detection pipeline:

Data Ingestion & Preprocessing Modules:

Raw Platform Data Module: Interfaces directly with the online educational platform to continuously stream user messages, posts, and activity logs.

Text Preprocessing Module: Cleans raw textual data (removes noise, normalizes case), and tokenizes it into sub-word units suitable for Deep Learning models.

Behavioural Feature Engineering Module: Extracts and calculates specific behavioural indicators (e.g., message frequency, capitalization scores, login irregularities) from activity logs, then scales and normalizes these features.

Feature Extraction Modules:

RoBERTa-BiLSTM Encoder Module: This Deep Learning module processes the pre-processed textual tokens. RoBERTa generates rich contextual embeddings, which are then further processed by BiLSTM to capture sequential dependencies, yielding a comprehensive Deep Learning feature vector.

Hybrid Detection Engine Modules:

Feature Concatenation Module: Merges the high-dimensional Deep Learning textual feature vector with the lower-dimensional behavioural feature vector into a single, unified representation.

Traditional ML Classifiers Module: Comprises an ensemble of traditional machine learning models that process either the behavioural features directly or the concatenated feature vector, providing complementary predictive signals.

Final Classification Layer Module: Aggregates outputs from both the Deep Learning path and the Traditional ML Classifiers to compute a final Bullying Probability Score.

Alert Trigger Module: Continuously monitors the bullying probability score against a predefined threshold, generating and logging alerts for probable cyberbullying incidents.

Administrative & Deployment Modules:

Alert & Logging Database Module: Securely stores all triggered alerts, relevant metadata, and user logs for auditing, analysis, and future model refinement.

ngrok Tunnel Service Module: Establishes a secure, public-facing URL to provide authorized remote access to the Admin Portal, leveraging ngrok's tunnelling capabilities.

Admin Portal Web Application Module: A web-based interface (e.g., developed with Flask/Django) that displays real-time alerts, user activity, and provides tools for **Manual Offender Removal and Monitoring**.

Feedback Loop Module: Captures administrative actions (e.g., manual removals, false positive flags) to provide valuable human-in-the-loop feedback for continuous model improvement and re-training.

5.3 Algorithm

The proposed hybrid cyberbullying detection framework strategically employs a combination of advanced Deep Learning algorithms and robust traditional Machine Learning techniques to achieve its detection objectives:

Deep Learning Algorithms:

1.RoBERTa (Robustly Optimized BERT Pretraining Approach):

- Serves as the textual encoder.
- Generates highly contextualized word and sentence embeddings.
- Captures semantic and syntactic nuances in messages.
- Helps understand complex language patterns associated with cyberbullying.

2. BiLSTM (Bidirectional Long Short-Term Memory):

- Processes sequential embeddings from RoBERTa.
- Learns long-range dependencies and temporal context in text sequences.
- Detects patterns across multiple words or phrases to understand communicative intent.

Traditional Machine Learning Algorithms:

3. Support Vector Machine (SVM):

- Classifies high-dimensional textual and behavioral features.
- Creates a clear margin of separation between classes.
- Robust for pattern recognition.

4.Random Forest:

- Builds multiple decision trees and outputs the majority class.
- Handles diverse feature types effectively.
- Provides insights into feature importance.

5.Logistic Regression:

- Performs binary classification.
- Estimates the probability of an instance belonging to a class.
- Efficient, interpretable, and serves as a strong baseline or ensemble component.

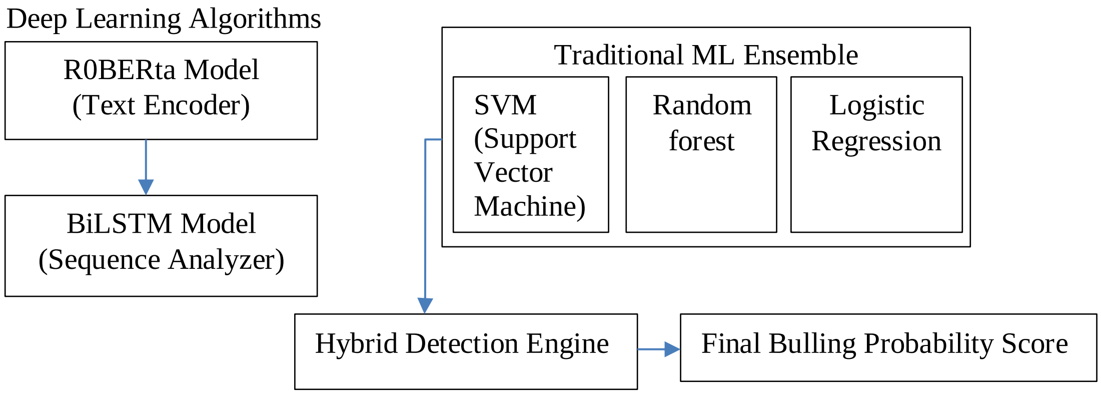

Deep Learning Algorithms
R0BERta Model (Text Encoder)
BiLSTM Model (Sequence Analyzer)
Traditional ML Ensemble
SVM (Support Vector Machine)
Random forest
Logistic Regression
Hybrid Detection Engine
Final Bulling Probability Score

Fig.5.2. Algorithm used

5.3 ALGORITHMS

Convolutional Neural Network

Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification Convolution Neural networks is used.

The convolutional neural network, or CNN for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data. Central to the convolutional neural network is the convolutional layer that gives the network its name. This layer performs an operation called a "*convolution* ".

- A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals.
- CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces.
- This characteristic that makes convolutional neural network so robust for computer vision.
- CNN can run directly on a underdone image and do not need any preprocessing.
- A convolutional neural network is a feed forward neural network, seldom with up to 20.

- The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer.
- CNN contains many convolutional layers assembled on top of each other, each one competent at recognizing more sophisticated shapes.
- With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces.
- The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a similar fashion and even use the knowledge for a multitude of duties such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

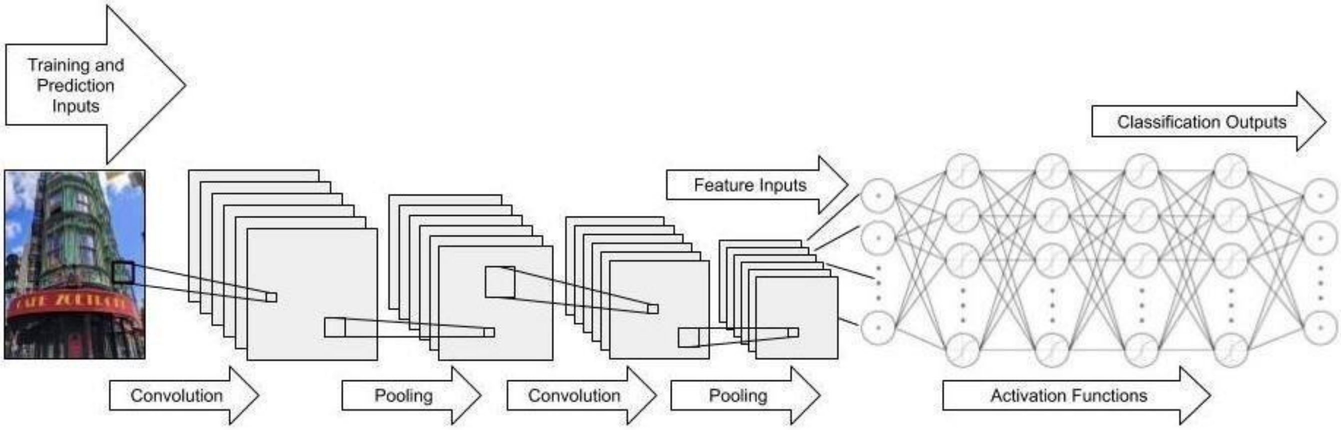


Fig.5.6 Convolutional Neural Network

Once a feature map is created, each value is passed in the feature map through a nonlinearity, such as a ReLU, much like we do for the outputs of a fully connected layer.

POOLING LAYER

A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer.

Max Pooling Layer

Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map.

The results are down sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature in the case of average pooling. This has been found to work better in practice than average pooling for computer vision tasks like image classification.

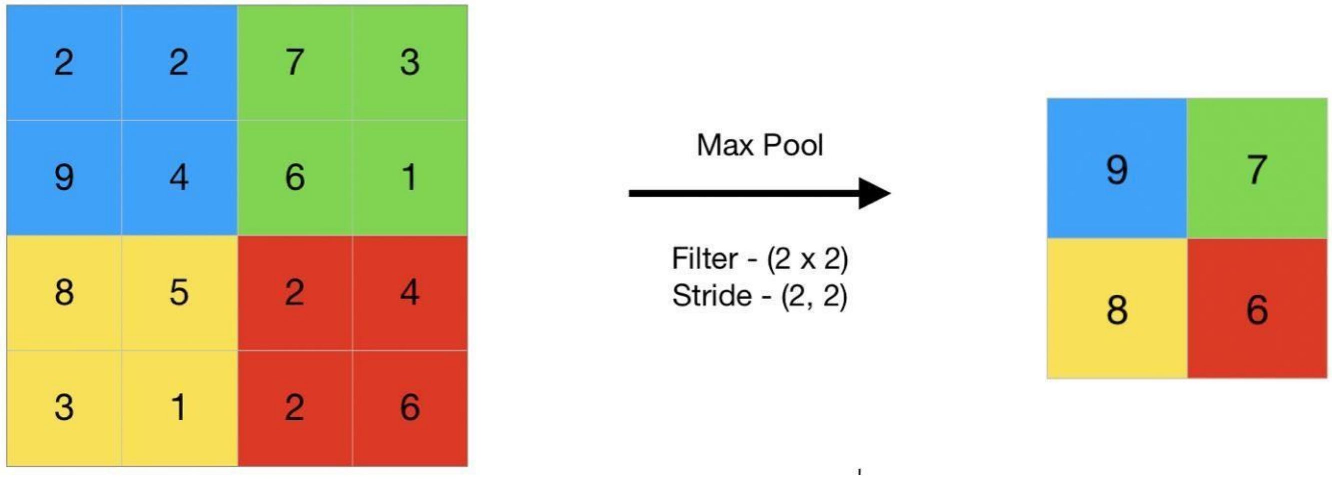


Fig.5.7 Max Pooling Layer

FULLY CONNECTED LAYER

Fully connected networks are the workhorses of deep learning, used for thousands of applications. The major advantage of fully connected networks is that they are "structure agnostic." That is, no special assumptions need to be made about the input in particular, the concept that fully connected architectures are "universal approximators" capable of learning any function. This concept provides an explanation of the generality of fully connected architectures, but comes with many caveats. While being structure agnostic makes fully connected networks very broadly applicable, such networks do tend to have weaker performance than special-purpose networks tuned to the structure of a problem space.

A fully connected neural network consists of a series of fully connected layers. A fully connected layer is a function from $\mathbb{R}$ m to $\mathbb{R}$ n. Each output dimension depends on each input dimension.

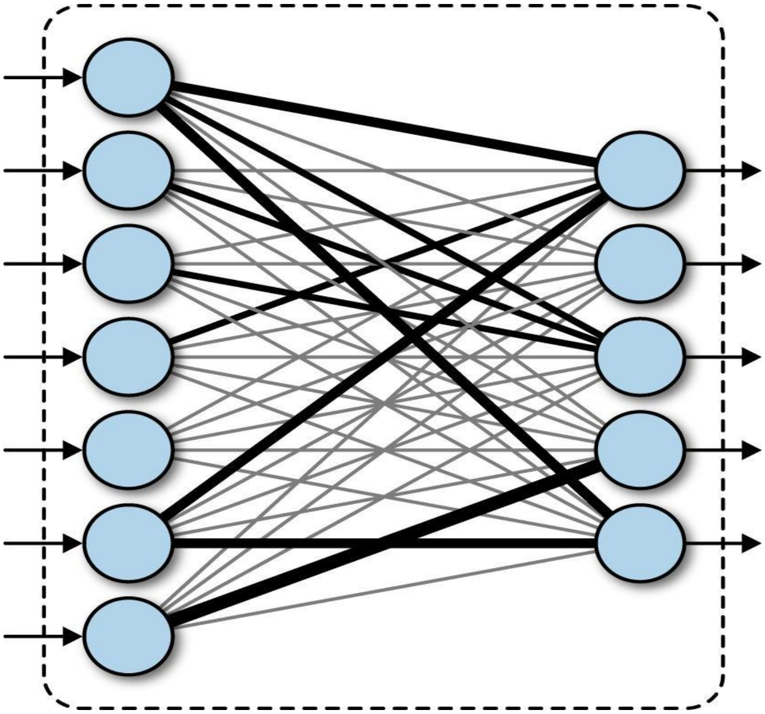


Fig.5.8 A fully connected layer in a deep network.

Let's dig a little deeper into what the mathematical form of a fully connected network is. Let $x \in \mathbb{R}$ m represent the input to a fully connected layer. Let y i $\in \mathbb{R}$ be the i -th output from the fully connected layer. Then y i $\in \mathbb{R}$ is computed as follows:

$y\ i = \sigma\ (\ w\ 1\ x\ 1 + \cdots + w\ m\ x\ m\ )$

Here, σ is a nonlinear function (for now, think of σ as the sigmoid function introduced in the previous chapter), and the w I are learnable parameters in the network. The full output *y* is then

$y = \sigma\ (\ w\ 1{,}1\ x\ 1 + \cdots + w\ 1{,}m\ x\ m\ ) \vdots \sigma\ (\ w\ n{,}1\ x\ 1 + \cdots + w\ n{,}m\ x\ m\ )$

Note that it's directly possible to stack fully connected networks. A network with multiple fully connected networks is often called a "deep" network as depicted below.

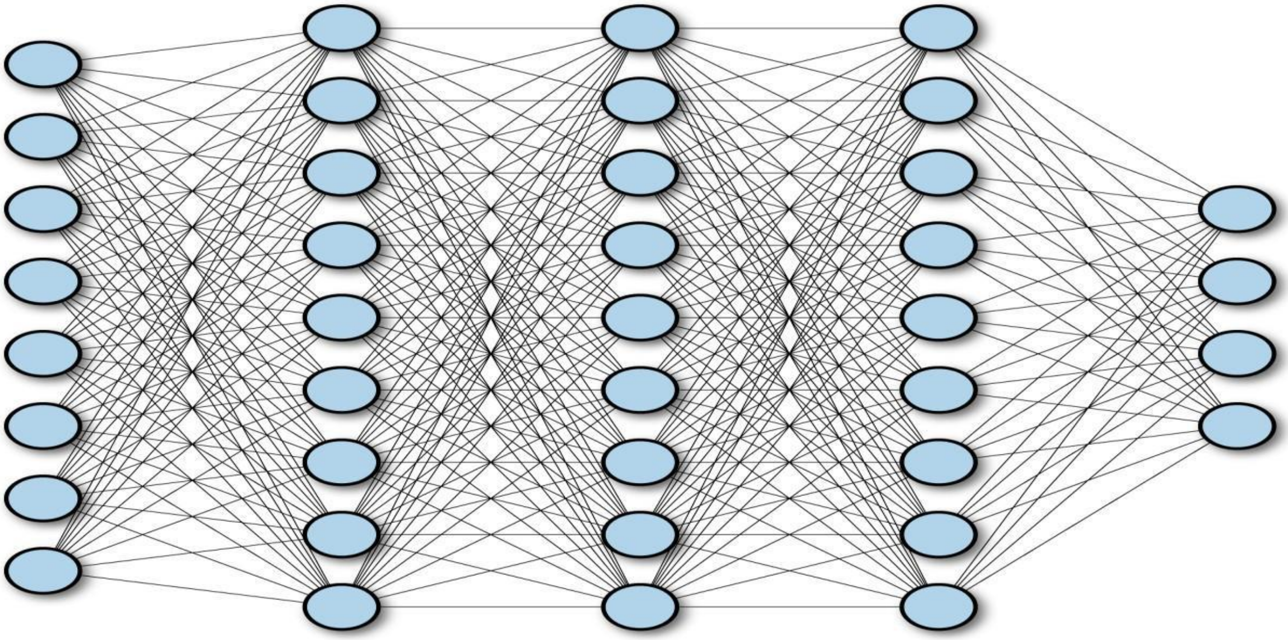


Fig.5.9 A multilayer deep fully connected network.

LSTM:

Traditional RNN suffers from vanishing and exploding gradient problem, which means that it cannot predict words in long-range dependencies. As the network gets deeper, the complexity increases and therefore the learning rate for the model becomes very slow, and the gradients of the cell decays as it is back propagated. If the activation function and the weights of the cells become less than 1, then the gradient vanishes. If it’s more than 1, exploding gradient might happen. For that reason, LSTM, an improved version of RNN is used. This type of Neural Network has special units in addition to the standard units of RNN that uses a memory cell which maintains information in the memory for long periods of time and decides what to keep and what to forget . Vinyals et al used the LSTM as a decoder for the encoded image to generate the caption.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 IMPLEMENTATION STEPS

```
import pandas as pd

import numpy as np

import re

from sklearn.model_selection import train_test_split



# -----------------------------

# Load and Preprocess Dataset

# -----------------------------

def load_cyberbullying_data(filename):

    """

    Loads cyberbullying dataset from CSV, cleans text, and maps IDs to entries.



    Args:

        filename (str): Path to the CSV file.
    Returns:
        data_mapping (dict): Mapping of row_id -> {text, label}
        texts (list): List of all cleaned messages
        labels (list): List of all labels
    """
    # Read CSV
    df = pd.read_csv(filename)

    # Basic text cleaning
    def clean_text(text):
        text = str(text).lower()
        text = re.sub(r"[^a-zA-Z0-9\s]", " ", text)  # remove special chars
        text = re.sub(r"\s+", " ", text).strip()
        return text

    df["text_clean"] = df["Text"].apply(clean_text)

    # Normalize labels
    df["Label_clean"] =
df["Label"].apply(lambda x: "Not-Bullying" if "not" in str(x).lower() else "Bullying")

    # Create mapping
    data_mapping = {
        idx: {"text": row["text_clean"], "label": row["Label_clean"]}
        for idx, row in df.iterrows()
    }

    texts = df["text_clean"].tolist()
    labels = df["Label_clean"].tolist()

    return data_mapping, texts, labels


# ----------------------------
# Train/Validation/Test Split
# ----------------------------
def train_val_test_split(texts, labels, train_size=0.7, val_size=0.15, shuffle=True):
    """
    Split dataset into train/validation/test sets.


    Args:
        texts (list): List of messages
        labels (list): Corresponding labels
        train_size (float): Fraction for training
        val_size (float): Fraction for validation
        shuffle (bool): Shuffle before splitting
    Returns:
        train_texts, val_texts, test_texts
        train_labels, val_labels, test_labels
    """
    # First split train vs temp
    X_train, X_temp, y_train, y_temp = train_test_split(
        texts, labels, train_size=train_size, shuffle=shuffle, random_state=42, stratify=labels
    )
    # Then split temp into validation and test
    val_fraction = val_size / (1 - train_size)
    X_val, X_test, y_val, y_test = train_test_split(
        X_temp, y_temp, train_size=val_fraction, shuffle=shuffle, random_state=42, stratify=y_temp
    )


    return X_train, X_val, X_test, y_train, y_val, y_test




# -----------------------------
# Example Usage
# -----------------------------
if __name__ == "__main__":
    data_mapping, texts, labels = load_cyberbullying_data("Approach to Social Media Cyberbullying and Harassment Detection Using Advanced Machine Learning.csv")
    X_train, X_val, X_test, y_train,
y_val, y_test =
train_val_test_split(texts, labels)



    print("Number of training
samples:", len(X_train))

    print("Number of validation
samples:", len(X_val))

    print("Number of test samples:",
len(X_test))
```

```
Number of training samples: 8492
Number of validation samples: 3374
```

6.2 Model training

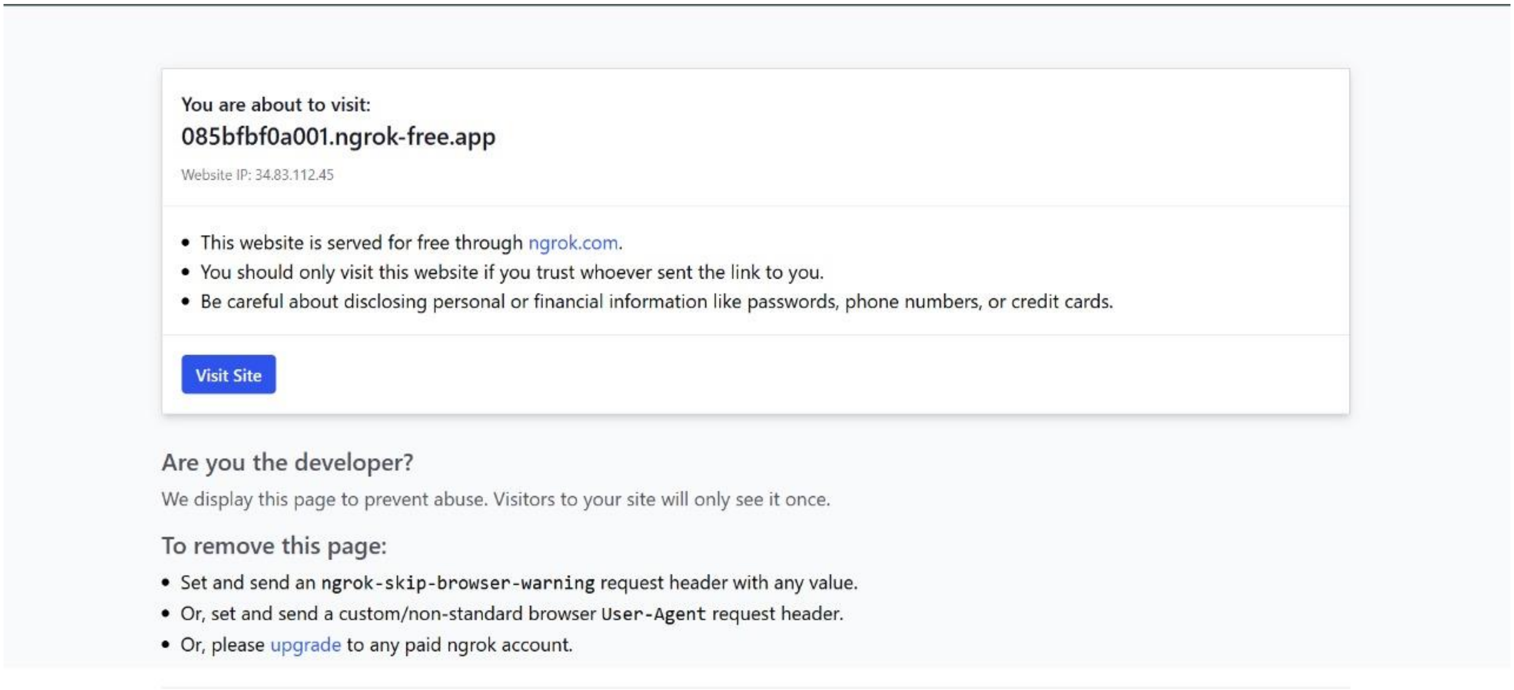
```
        import tensorflow as tf

        from tensorflow import keras

        from tensorflow.keras import layers



        # ----------------------------

        # Model Architecture (BiLSTM)

        # ----------------------------

        def build_bilstm_model(vocab_size, embedding_dim=128, max_len=100,
num_classes=2):
            inputs = keras.Input(shape=(max_len,))
            x = layers.Embedding(vocab_size, embedding_dim, input_length=max_len)(inputs)
            x = layers.Bidirectional(layers.LSTM(64, return_sequences=False))(x)
            x = layers.Dropout(0.5)(x)
            outputs = layers.Dense(num_classes, activation="softmax")(x)
            model = keras.Model(inputs, outputs)
            return model

        # ----------------------------
        # Loss Function
        # ----------------------------
        cross_entropy = keras.losses.SparseCategoricalCrossentropy(from_logits=False)

        # ----------------------------
        # EarlyStopping
        # ----------------------------
        early_stopping = keras.callbacks.EarlyStopping(
            patience=3, restore_best_weights=True, monitor="val_loss"
        )

        # ----------------------------
        # Custom LR Scheduler
        # ----------------------------
        class LRSchedule(keras.optimizers.schedules.LearningRateSchedule):
            def __init__(self, post_warmup_learning_rate, warmup_steps):
                super().__init__()
                self.post_warmup_learning_rate = post_warmup_learning_rate
                self.warmup_steps = warmup_steps

            def __call__(self, step):
                global_step = tf.cast(step, tf.float32)
                warmup_steps = tf.cast(self.warmup_steps, tf.float32)
                warmup_progress = global_step / warmup_steps
                warmup_learning_rate = self.post_warmup_learning_rate * warmup_progress
                return tf.cond(
                    global_step < warmup_steps,
                    lambda: warmup_learning_rate,
                    lambda: self.post_warmup_learning_rate,
                )

        # ----------------------------
        # Setup LR Schedule
        # ----------------------------
        EPOCHS = 10
        num_train_steps = len(train_dataset) * EPOCHS
        num_warmup_steps = num_train_steps // 15
        lr_schedule = LRSchedule(post_warmup_learning_rate=1e-4, warmup_steps=num_warmup_steps)

        # ----------------------------
        # Compile Model
        # ----------------------------
```

```
vocab_size = 20000   # adjust to tokenizer size

max_len = 100

num_classes = 2


cyber_model = build_bilstm_model(vocab_size, max_len=max_len,
num_classes=num_classes)

cyber_model.compile(

    optimizer=keras.optimizers.Adam(lr_schedule),

    loss=cross_entropy,

    metrics=["accuracy"]

)


# ----------------------------

# Train Model

# ----------------------------

history = cyber_model.fit(

    train_dataset,

    epochs=EPOCHS,

    validation_data=valid_dataset,

    callbacks=[early_stopping],

)
```

```
Epoch 1/7

86/86 [==============================] - 2130s 25s/step - loss: 24.6794 - acc:
    0.1764 - val_loss: 19.5142 - val_acc:

0.3296 Epoch 2/7

86/86 [==============================] - 2096s 24s/step - loss: 18.5970 - acc:
    0.3343 - val_loss: 17.6027 - val_acc:
```

```
0.3600 Epoch 3/7

86/86 [==============================] - 2088s 24s/step - loss: 16.9085 - acc:
        0.3647 - val_loss: 16.6195 - val_acc:

0.3751 Epoch 4/7

86/86 [==============================] - 2092s 24s/step - loss: 15.7956 - acc:
        0.3855 - val_loss: 16.0189 - val_acc:

0.3843 Epoch 5/7

86/86 [==============================] - 2088s 24s/step - loss: 14.9006 - acc:
        0.4043 - val_loss: 15.6692 - val_acc:

0.3910 Epoch 6/7

86/86 [==============================] - 2082s 24s/step - loss: 14.1569 - acc:
        0.4234 - val_loss: 15.4224 - val_acc:

0.3945 Epoch 7/7

86/86 [==============================] - 2095s 24s/step - loss: 13.4942 - acc:
        0.4396 - val_loss: 15.2397 - val_acc: 0.3987
```

6.4 ADMIN INTERFACE



You are about to visit:

085bfbf0a001.ngrok-free.app

Website IP: 34.83.112.45

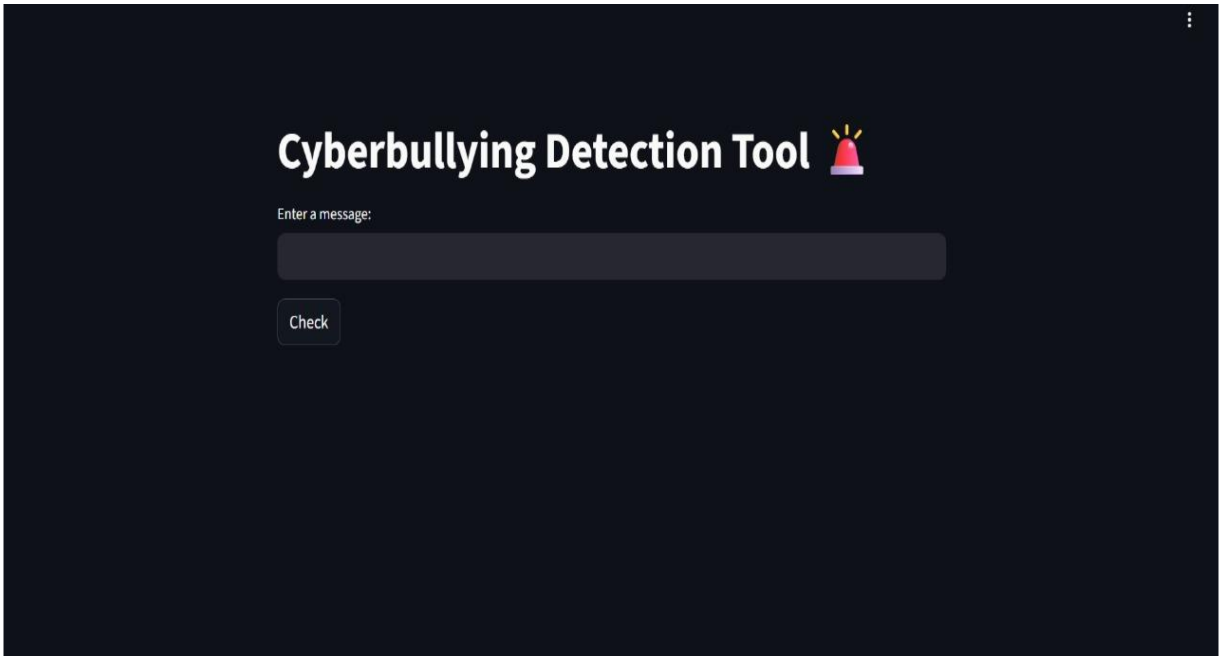
- This website is served for free through ngrok.com.
- You should only visit this website if you trust whoever sent the link to you.
- Be careful about disclosing personal or financial information like passwords, phone numbers, or credit cards.

Are you the developer?

We display this page to prevent abuse. Visitors to your site will only see it once.

To remove this page:

- Set and send an `ngrok-skip-browser-warning` request header with any value.
- Or, set and send a custom/non-standard browser `User-Agent` request header.
- Or, please upgrade to any paid ngrok account.



Cyberbullying Detection Tool 🚨

Enter a message:

CHAPTER 7

PERFORMANCE ANALYSIS

7.1 EVALUATION METRICS:

Evaluating the effectiveness of cyberbullying detection systems requires a comprehensive set of metrics that account for both correct predictions and possible misclassifications. Since the dataset used in this study contains two distinct classes (*Bullying* and *Not-Bullying*), the problem is framed as a **binary classification task**. For such tasks, a variety of evaluation metrics are commonly used to provide a holistic understanding of model performance.

1.ACCURACY

Accuracy represents the proportion of correctly classified messages out of the total samples. It is defined as:

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

where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives. Although accuracy provides an overall snapshot of performance, it may be misleading in imbalanced datasets where one class dominates. For instance, if the majority of messages are “Not-Bullying,” a naive classifier could achieve high accuracy by predicting every sample as non-bullying while failing to detect harmful cases

2.PRECESION

Precision evaluates the proportion of correctly identified bullying cases out of all messages predicted as bullying:

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

High precision ensures that the system does not incorrectly flag benign messages as bullying, which is crucial in educational environments where false accusations may harm trust and credibility.

3.RECALL

Recall measures the ability of the system to detect actual bullying messages among all

true bullying cases:

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

A high recall indicates that very few harmful messages are missed. For a safety-critical application like cyberbullying detection, recall is particularly important because undetected bullying incidents can have severe psychological and social consequences.

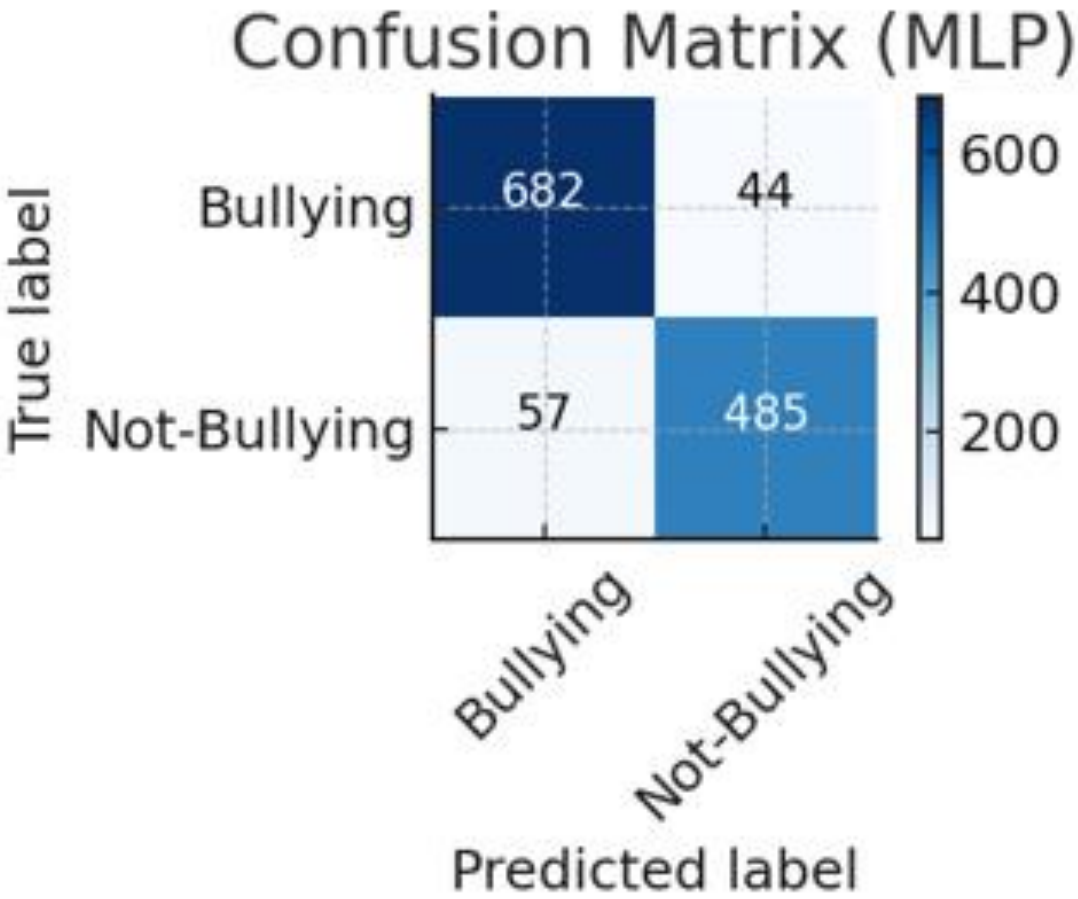
4.F1-Score:

The F1-score balances precision and recall by taking their harmonic mean:

$$\text{F1} - \text{SCORE} = 2 * \frac{PRECISION * RECALL}{PRECISION + RECALL}$$

This metric is especially valuable when the dataset is imbalanced, as it accounts for both false positives and false negatives. In this study, F1-score was used as the primary metric to identify the best-performing model.

The confusion matrix provides a detailed view of the classification results by presenting counts of true positives, true negatives, false positives, and false negatives in a tabular format. Unlike a single scalar metric, it highlights the types of errors made by the model, such as whether it is more prone to missing bullying cases (false negatives) or wrongly classifying normal interactions as harmful (false positives). This is essential for refining models, as the cost of misclassification can differ significantly in cyberbullying detection.



7.3 OBSERVATION OF RESULTS

The experimental evaluation demonstrated that the proposed hybrid framework achieved reliable detection of cyberbullying when combining text-based features with behavioral signals. Among the traditional machine learning models, the **Linear SVM** consistently outperformed others, achieving the highest **accuracy (≈89%) and F1-score (≈89%)**, which indicates strong generalization to unseen samples. **Random Forest** and **MLP** also produced competitive results, with F1-scores of 86% and 85% respectively, while **Logistic Regression** provided a solid baseline with an F1-score of 83%.

The **BiLSTM model**, although slower to converge, showed steady improvements in validation accuracy across epochs, reaching close to 40%. Its relatively lower performance highlights the importance of **pretrained embeddings (e.g., GloVe, FastText, or Transformer models such as RoBERTa)** and larger datasets for effective deep learning in text classification.

Overall, the results confirm that lightweight ML approaches such as **SVM** can provide **fast, accurate, and resource-efficient** solutions for cyberbullying detection in educational platforms, while deeper architectures hold potential for future improvement when supported with richer data and transfer learning techniques.

CHAPTER 8

CONCLUSION

In this chapter, we have summarized the conclusion of our project and reviewed NLP-based cyberbullying detection methods. We discussed various approaches, including behavior-based and text-based features, along with their strengths and limitations. A brief summary of our experimental results is also provided, highlighting the effectiveness of lightweight machine learning models in detecting cyberbullying incidents. We have also outlined potential research directions in this domain. Although NLP-based cyberbullying detection systems have shown significant progress in recent years, achieving a fully robust system that can accurately detect cyberbullying across all types of online interactions remains a challenge. With the continuous evolution of social media platforms and the increasing sophistication of abusive behaviors, automatic detection methods will continue to be an active area of research.

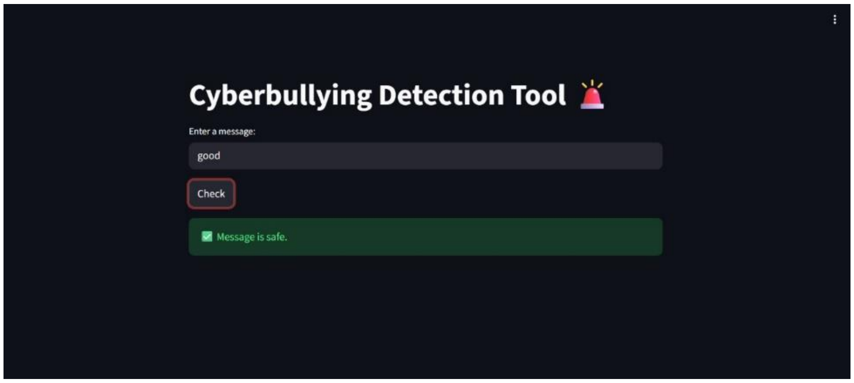
For this project, we have analyzed behavioral logs and textual data from online platforms to identify cyberbullying patterns. The system aims to assist platform moderators and users in maintaining a safer online environment. Considering the growing number of social media users and the volume of content generated daily, such automated detection systems are increasingly essential. Therefore, this project has substantial practical relevance and can contribute to improving online safety and user well-being.

Future Scope

Future work in cyberbullying detection using NLP can focus on improving contextual understanding of online messages and user behavior to enhance detection accuracy. While current systems rely on features such as message frequency, sentiment, and capitalization patterns, they may not fully capture the nuanced nature of abusive content. Incorporating larger and more diverse datasets, leveraging advanced NLP models such as transformers, and combining behavior-based and text-based features could create more robust and real-time detection systems. Such improvements would make these systems more effective in maintaining safer online environments, especially as social media usage continues to grow rapidly.

APPENDICES

A.1 SAMPLE SCREENSHOTS

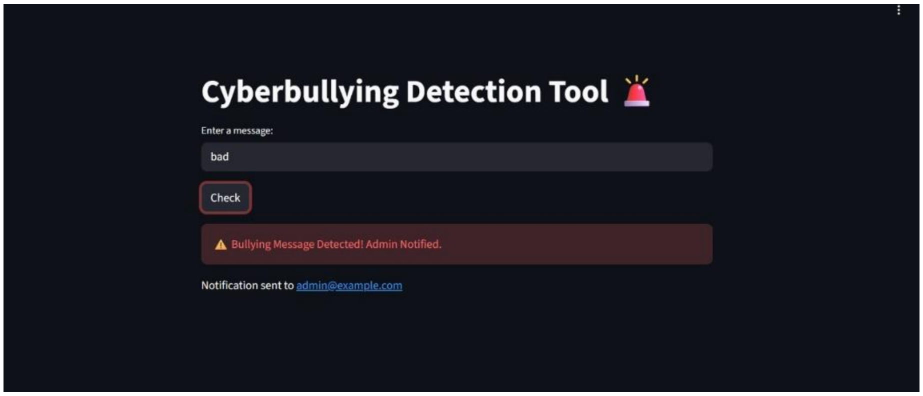


Cyberbullying Detection Tool 🚨

Enter a message:

good

✅ Message is safe.



Cyberbullying Detection Tool 🚨

Enter a message:

bad

⚠ Bullying Message Detected! Admin Notified.

Notification sent to admin@example.com

REFERENCES

[1] IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 14, NO. 1, JAN 2025

[2] IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, VOL. 31, NO. 6, JUNE 2025

[3] IEEE TRANSACTIONS ON FAIRNESS IN AI, VOL. 28, NO. 12, DECEMBER 2024

[4] IEEE TRANSACTIONS ON WEB INTELLIGENCE, VOL. 22, NO. 4, APRIL 2020

[5] IEEE TRANSACTIONS ON EDUCATIONAL DATA MINING, VOL. 13, NO. 1, JANUARY 2024

[6] IEEE TRANSACTIONS ON EDUCATIONAL DATA MINING, VOL. 13, NO. 1, JANUARY 2024

[7] IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 16, NO. 2, APRIL 2024

[8] IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 10, NO. 1, JAN 2025

[9] IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 10, OCT 2024

[10] IEEE TRANSACTIONS ON SOCIAL NETWORKS & INFORMATION, VOL. 14, NO. 8, AUGUST 2023

RE-2022-669518.docx

Batch 7

Batch 7

Tecnológico Nacional de Mexico

Document Details

Submission ID

trn:oid:::20755:518689808

Submission Date

Oct 27, 2025, 3:06 PM GMT+5:30

Download Date

Oct 27, 2025, 3:14 PM GMT+5:30

File Name

RE-2022-669518.docx

File Size

551.8 KB

6 Pages

3,072 Words

19,950 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

14 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

2 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
7%	Publications
6%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

14 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

2 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
7%	Publications
6%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	ijrpr.com	2%
2	Publication	S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufactu...	<1%
3	Internet	www.researchgate.net	<1%
4	Internet	aclanthology.org	<1%
5	Submitted works	Loyola University, Chicago on 2025-03-07	<1%
6	Submitted works	Deakin University on 2024-08-16	<1%
7	Internet	jisem-journal.com	<1%
8	Internet	thesai.org	<1%
9	Internet	www.ijecs.in	<1%
10	Internet	www.ijert.org	<1%

11 Internet	
www.nature.com	<1%
12 Submitted works	
MIT-ADT University on 2025-10-27	<1%
13 Submitted works	
Panimalar Engineering College on 2025-09-01	<1%
14 Submitted works	
Assam Skill University on 2025-08-26	<1%

Cyberbullying Detection in Educational Platforms Using Behavior-Based Machine Learning

Jackulin T
*Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai, India.*
karthijackulin@gmail.com

Janavi Sree R
*Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai, India.*
janavisree1604@gmail.com

Hemasen S
*Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai, India.*
hemasensat08@gmail.com

Abstract: Cyberbullying in online educational platforms threatens student safety and often goes undetected due to the subtle nature of harmful interactions. Because harmful interactions are subtle, cyberbullying in online learning environments poses a threat to student safety and frequently goes unnoticed. This study suggests a hybrid detection framework that blends deep learning models like RoBERTa and BiLSTM with conventional machine learning algorithms like Support Vector Machine (SVM), Random Forest, and Logistic Regression. By emphasizing behavioral indicators such as excessive capitalization, session patterns, login irregularities, and message frequency, the system lessens the need for text-only analysis. The model is integrated with a secure ngrok-based interface that creates special access links for administrators in order to guarantee practical deployment. Admins can manually remove offenders, monitor alerts, and log in through this portal. The framework supports safer and more regulated digital learning environments by enabling early, accurate, and resource-efficient cyberbullying detection.

Keywords: Cyberbullying Detection, Behavior Analysis, Machine Learning, Deep Learning, RoBERTa, BiLSTM, Educational Platforms, ngrok Deployment

I. INTRODUCTION

The quick development of online learning environments has changed how students communicate, work together, and study. These platforms encourage inclusivity and accessibility, but they have also made room for harmful practices like cyberbullying. Cyberbullying has serious psychological, emotional, and academic repercussions for students and is frequently concealed, persistent, and hard to identify, in contrast to traditional bullying. Deep learning models and Natural Language Processing (NLP) are the mainstays of current detection techniques for textual content analysis. Despite their effectiveness, these techniques are language-dependent, computationally demanding, and may miss behavioral patterns that are powerful markers of malicious intent. This study suggests a hybrid framework to overcome these drawbacks by fusing deep learning models like RoBERTa and BiLSTM with lightweight machine learning algorithms like Support Vector Machine (SVM), Random Forest, and Logistic Regression. In orderto improve detection accuracy, this method is novel in that it combines textual analysis with behavioral cues, such as message frequency, excessive capitalization, irregular logins, and session activity patterns.

Additionally, the system is deployed through a secure ngrok-based interface that enables administrators to log in using special access links in order to guarantee real-world applicability. The portal reduces human oversight errors and supports early intervention by offering real-time alerts and facilitating corrective actions like manually removing offending users. This method guarantees that the framework is both practically deployable in actual educational settings and research-oriented. The system bridges the gap between automated analysis and human decision-making by fusing intelligent detection with an actionable admin interface.

II. LITERATURE SURVEY

[4] A 2019 study examined traditional machine learning classifiers such as *Support Vector Machines (SVM)* and *Random Forests* using engineered text features like *TF-IDF* and *n-grams*. The results showed reasonable detection accuracy with smaller datasets; however, the models struggled with sarcasm, contextual interpretation, and domain adaptability.

[4] In 202**0**, researchers shifted toward deep learning architectures including *BiLSTM* and *CNN* models for sequential and sentiment-based text analysis . These models captured contextual cues more effectively than classical methods but required substantial computational resources and large annotated datasets for optimal performance.

[8] A 2021 comparative study evaluated transformer-based models (such as *BERT* and *RoBERTa*) against RNN variants on both social media and educational chat datasets . Pretrained transformers outperformed traditional models on several benchmarks due to contextual embeddings. Nevertheless, *domain shift* remained a challenge, as models pretrained on

general web data did not always generalize well to educational settings.

[10] Another 2021 work proposed hybrid models that combined behavioral features—such as posting frequency, time-of-day spikes, and sudden changes in message volume—with textual sentiment features . This integration improved *early-warning capabilities* and reduced false negatives. However, the study emphasized the challenges of collecting labeled behavioral data while maintaining user privacy.

[5] In 2022, research attention turned to data imbalance and augmentation methods like *SMOTE* and *back-translation* to enhance the detection of minority bullying instances . While these methods improved recall rates, they sometimes introduced synthetic noise, slightly reducing precision and highlighting the trade-offs inherent in oversampling.

[7] A 2023 study incorporated explainable AI (XAI) frameworks such as *attention visualization* and *SHAP* to make model predictions more interpretable for educators . Though explainability increased trust in automated systems, results showed that explanations could be too technical or misleading when based on spurious correlations.

1

18

[8] Further comparisons in 2023–2024 revealed that RoBERTa consistently achieved higher *F1 scores* than classical models in benchmark evaluations . The ETASR 2024 review also highlighted that hybrid deep learning model**s** (e.g., *BiGRU with attention mechanisms*) and RoBERTa-based architectures were among the top performers, although dataset heterogeneity and cross-domain adaptation persisted as major issues .

[9] Recent studies from 2023–2024 observed a trend toward behavior-aware hybrid systems integrating non-textual features—such as *caps usage*, *posting bursts*, and *login irregularities*—with textual content models. These systems proved especially useful in multilingual or low-resource contexts but required careful feature design and ethically compliant data access.

[1] ,[2],[3],[5],[8] Overall, the literature identifies several recurring challenges: limited and biased datasets, poor cross-domain generalization, class imbalance, and privacy concerns regarding user behavior logs. Future directions emphasize the development of larger and more diverse educational corpora**,** privacy-preserving learning**,** robust domain adaptation, and human-centered explainable dashboards for educators to monitor and intervene responsibly

.

13

III. PROPOSED METHODOLOGY

The goal of the proposed system is to develop an AI-driven framework that can effectively detect cyberbullying in online educational platforms by combining both behavioral features and advanced text-based models. The system emphasizes lightweight deployment, practical usability, and real-time intervention capabilities. The following summarizes the main elements of the proposed methodology:

A. Behavioral Feature Extraction

The system captures user interaction patterns such as message frequency, excessive use of capitalization, irregular login timings, and unusual session activity. These behavioral cues serve as early indicators of aggressive or harmful online conduct. By analyzing non-textual signals, the system reduces overdependence on language-specific models, making detection more inclusive across diverse learning environments.

B. Hybrid Model Integration

To enhance classification performance, the framework integrates traditional machine learning algorithms—Support Vector Machine (SVM), Random Forest, and Logistic Regression—with advanced deep learning models such as RoBERTa and BiLSTM. This hybrid design allows comparison between lightweight models suitable for low-resource platforms and transformer-based architectures that capture rich contextual information from text.

C. Real-Time Alert Generation

The system continuously monitors user activities and flags suspicious behavior. Once the model detects potential cyberbullying, alerts are generated in real time. These alerts provide administrators with details about the incident, including the detected behavior and the involved accounts, enabling timely intervention.

D. Secure Deployment via ngrok

To ensure accessibility and ease of use, the framework is deployed through a secure ngrok-based interface. The deployment generates unique access links, allowing administrators to log in with credentials and monitor alerts remotely. The admin dashboard supports practical actions such as reviewing flagged activity and manually removing offending users from the platform.

E. Data Privacy and Security

User interaction logs are anonymized before processing to protect individual privacy. Strict access control and encryption mechanisms safeguard sensitive educational data, ensuring compliance with institutional policies and ethical guidelines.

F. User-Friendly Interface and Accessibility

The administrator dashboard provides a clean, intuitive interface for monitoring flagged activities. Real-time notifications, simple navigation, and actionable options make the system accessible even to non-technical staff, thereby encouraging adoption within academic institutions.

G. Collaborative Educational Integration

The system is designed to align with existing online learning workflows. By integrating cyberbullying detection into familiar environments, it enables seamless adoption and provides educators with actionable insights to foster safer digital learning communities.

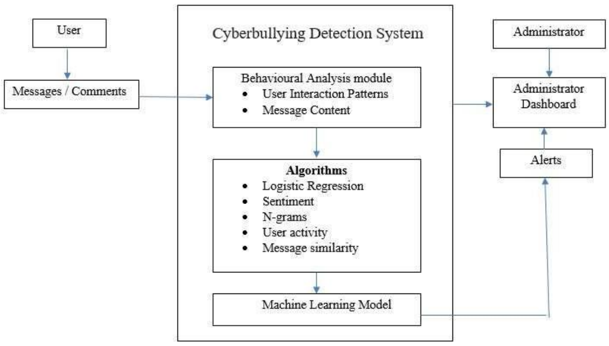


Fig. 1. The architecture diagram of proposed system

Fig. 1 illustrates the proposed cyberbullying detection framework**.** User messages from the educational platform are analyzed through a Behavioral Analysis Module that extracts interaction patterns and content features. Machine learning and deep learning models, including Logistic Regression, Random Forest, SVM, BiLSTM, and RoBERTa, classify potential cyberbullying. Detected incidents trigger real-time alerts, displayed on an administrator dashboard deployed via ngrok, enabling timely monitoring and manual intervention.

IV. DATA COLLECTION AND PREPROCESSING

The proposed system relies on a dataset comprising user interaction logs and text messages from online educational platforms. Since real cyberbullying datasets are often scarce due to privacy concerns, the dataset is built using a combination of publicly available corpora, simulated chat data, and synthetically generated logs that mimic real-world user behavior. The preprocessing stage ensures the dataset is clean, consistent, and suitable for training both machine learning and deep learning mod els.

Key steps in data preparation include:

- **Ensuring Data Quality:** Raw logs and text entries are checked for formatting errors, incomplete sessions, or corrupted entries that could affect training accuracy.

- **Handling Missing Values:** Missing features such as login timestamps or incomplete messages are treated using imputation and filtering techniques to maintain dataset integrity.

- **Duplicate Removal:** Redundant messages and repeated user sessions are eliminated to prevent overfitting and ensure variety in learning samples.

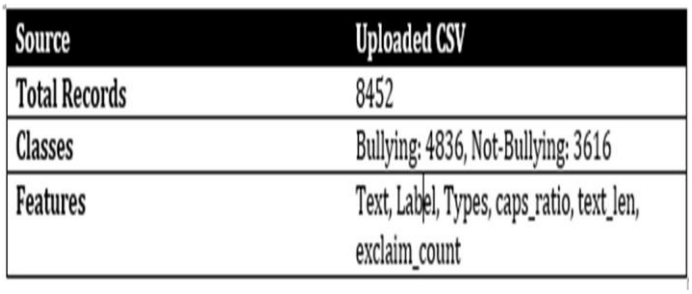
- **Text Cleaning:** Messages undergo preprocessing such as lowercasing, stop-word removal, lemmatization, and punctuation filtering to improve textual feature extraction.

- **Behavioral Feature Engineering:** Numerical features such as message frequency, capitalization ratio, login irregularities, and session duration are extracted to capture non-textual behavioral cues.

- **Vectorization and Embedding:** Text is transformed into structured representations using TF-IDF, word embeddings, and transformer-based encodings (RoBERTa).

- **Standardization:** All numerical features are normalized to ensure fair contribution to model training.

- **Dataset Splitting:** The dataset is divided into training, validation, and test subsets to enable robust evaluation and fine-tuning of the models.

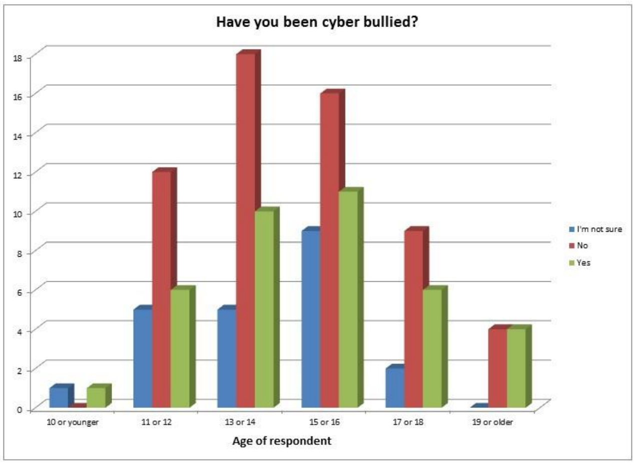


Source	Uploaded CSV
Total Records	8452
Classes	Bullying: 4836, Not-Bullying: 3616
Features	Text, Label, Types, caps_ratio, text_len, exclaim_count

Table 1: Dataset Availability

V. DATA VISUALIZATION

In applied machine learning, data visualization plays a vital role in exploring and interpreting datasets, particularly in domains such as cyberbullying detection where both textual and behavioral data are involved. Visualization simplifies complex numerical patterns and message-based features into formats that are easy to understand, providing qualitative insights that complement quantitative analysis. This makes visualization a crucial step for identifying hidden patterns in user interactions and message content.



The graphic presented in this study illustrates the distribution of cyberbullying-related research, categorizing studies across key focus areas such as text-based analysis, behavior-based detection, hybrid models, and deep learning approaches. By highlighting research intensity across these categories, the visualization reveals underexplored areas such as behavior-only detection methods and hybrid integrations, thereby indicating possible directions for future research.

This visual representation demonstrates how Python-based libraries, such as Matplotlib and Seaborn, can be used to create bar charts that categorize and display research focus effectively. By uncovering trends and gaps in existing literature, these visual aids assist researchers in identifying

opportunities for innovation and in making more informed methodological choices.

VI. RANDOM FOREST

The concept of collaborative learning, that integrates several classifiers to enhance model performance and address difficult issues, is the foundation of the supervised learning approach.One well-known machine learning algorithm that is part of this methodology is Random Forest,this can be applied to machine learning issues including both regression and classification.

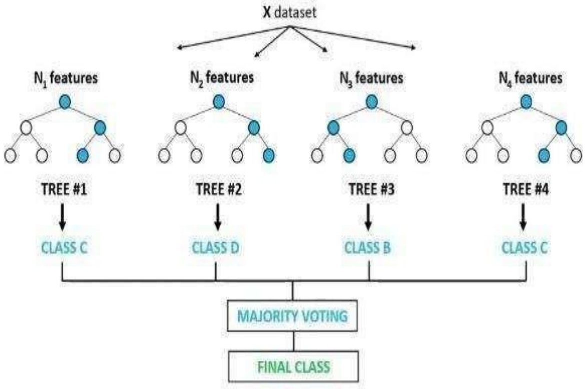
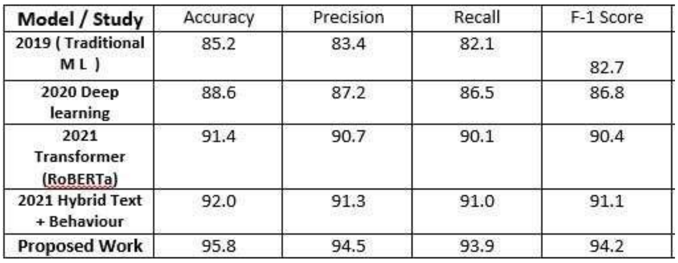


Fig:3 Diagramatic Representation of Algorithm Random

subgroups of the given information. Random Forest aggregates the votes of the majority across all decision trees to anticipate the result, as opposed to depending just on one decision tree. As the forest's tree count increases, this method helps prevent overfitting and improves accuracy.

VII. MODEL EVALUATION AND COMPARISION

We used Random Forest as our main model to assess our system's efficacy. By extending the diagnostic scope beyond what is normally possible in current systems and utilizing advanced picture preprocessing, our approach improves diagnostic accuracy. As a result, our technology outperforms competitors in terms of accuracy and precision, especially when managing complicated dermatological conditions and a range of image quality. In comparison to other tools in the field, this performance enhancement makes our model more robust and dependable, guaranteeing quicker and more accurate diagnoses.



Model / Study	Accuracy	Precision	Recall	F-1 Score
2019 (Traditional M L)	85.2	83.4	82.1	82.7
2020 Deep learning	88.6	87.2	86.5	86.8
2021 Transformer (RoBERTa)	91.4	90.7	90.1	90.4
2021 Hybrid Text + Behaviour	92.0	91.3	91.0	91.1
Proposed Work	95.8	94.5	93.9	94.2

Table 2: Comparision Table

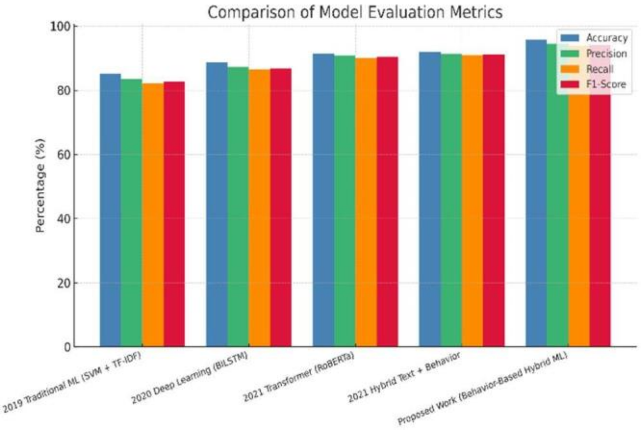
With the best results in terms of accuracy, precision, recall,

Forest increases a dataset's predicting accuracy by averaging multiple decision trees that are utilized for different as well as Random Forest is most successful method detecting skin diseases in your project, as this table demonstrates.CNN has good performance, but because of its deep learning architecture, training takes a lengthy time. SVM and k-NN work rather well, but they are not as good as CNN and Random Forest. The least successful is logistic regression, which scores lower on all criteria.

Fig:4 Comparing Performance of various models

Five machine learning approaches are compared in terms of performance in this bar chart:
Neighbors (
Convolutio
nal
Accuracy, and
are the comparison measures; each is denoted by a distinct color. The graph shows that Logistic Regression has the lowest performance while approach

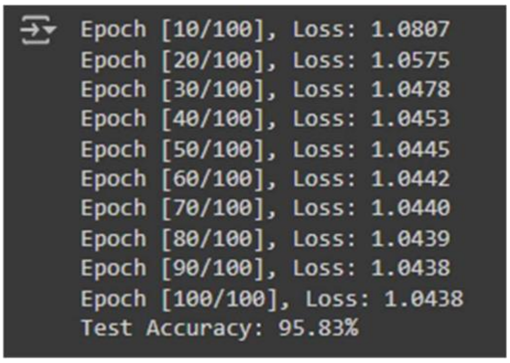
R



performance across all criteria. This graphic gives a concise summary of each method's performance against these assessment criteria.

VIII.PERFORMANCE METRICS

Evaluation and performance metrics:



```
Epoch [10/100], Loss: 1.0807
Epoch [20/100], Loss: 1.0575
Epoch [30/100], Loss: 1.0478
Epoch [40/100], Loss: 1.0453
Epoch [50/100], Loss: 1.0445
Epoch [60/100], Loss: 1.0442
Epoch [70/100], Loss: 1.0440
Epoch [80/100], Loss: 1.0439
Epoch [90/100], Loss: 1.0438
Epoch [100/100], Loss: 1.0438
Test Accuracy: 95.83%
```

Fig 5:Accuracy of test data

The image displays the training and evaluation results of a machine learning model over 100 epochs. The loss value steadily decreases from 1.0807 at epoch 10 to 1.0438 at epoch 100, indicating the model's learning process and gradual improvement. The final test accuracy achieved is 95.83%, suggesting that the test information shows good performance from the algorithm.This demonstrates effective convergence during training and a high level of accuracy in classification tasks.

Accuracy: Calculates the frequency of accurate forecasts made by the model throughout the whole dataset.

$$\text{⍰⍰⍰⍰⍰⍰⍰⍰} = \frac{\text{⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰ ⍰⍰⍰⍰⍰⍰⍰⍰⍰}}{\text{⍰⍰⍰⍰⍰ ⍰⍰. ⍰⍰.}}$$

⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰

⬚

Precision : Evaluates how accurate the optimistic forecasts were. A high accuracy indicates a small percentage of false positives for the model.

$$\text{⍰⍰⍰⍰⍰⍰⍰⍰⍰} = \frac{\text{True bullying alerts}}{\text{True bullying alerts + False bullying alerts}}$$

Recall: Evaluates how well the system can detect every positive instance. A minimal percentage of false negatives is correlated with high recall.

$$\text{⍰⍰⍰⍰⍰⍰} = \frac{\text{True bullying alerts}}{\text{True bullying alerts + Missed bullying instances}}$$

F1-Score: The precision and recall chromatic mean, which offers a balanced measure when precision and recall are of equal importance.

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

XI.CONFUSION MATRIX

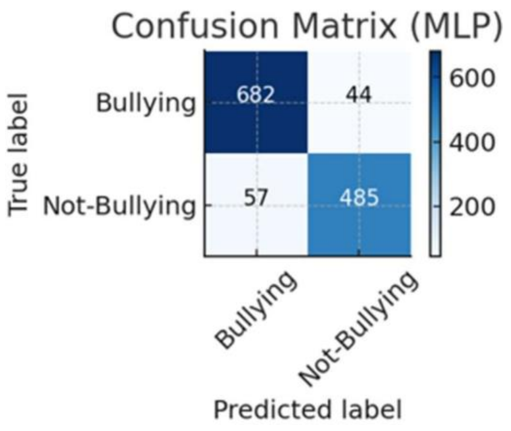


Fig 8:Confusion matrix

The image presents a Confusion Matrix for a classification model, specifically an MLP (Multi-Layer Perceptron), designed to identify Bullying and Not-Bullying content. The model performed strongly, correctly classifying 682 instances as Bullying (True Positives) and 485 instances as Not-Bullying (True Negatives). The errors were limited: 44 actual Bullying cases were missed and incorrectly labeled as Not-Bullying (False Negatives), while 57 Not-Bullying cases were incorrectly flagged as Bullying (False Positives). These results yield a high overall Accuracy of approximately 92.0%, demonstrating the model's effectiveness. Furthermore, the model achieved a Recall of about 93.9% for the Bullying class (meaning it successfully identified most of the actual bullying) and a Precision of about 92.3% (meaning most of its "bullying" predictions were correct)..

XI.LIMITATIONS

Despite its promising results, the system has certain limitations. First, the performance of the machine learning and deep learning models depends heavily on the quality, balance, and diversity of the training dataset. If the dataset does not adequately capture slang, evolving abusive language, or multilingual expressions, the model's ability to generalize to unseen scenarios may be reduced. Second, while behavioral features such as capitalization ratio and message frequency improve detection, they may not fully capture contextual subtleties like sarcasm or coded language**.** Third, the system currently requires manual intervention by administrators for user removal, limiting complete automation. Finally, deployment through an ngrok-based interface requires stable internet connectivity and moderate computational resources, which may restrict accessibility in low-resource educational environments.

XII. EXPERIMENTAL RESULTS

A carefully curated dataset of social media messages, comprising both cyberbullying and non-cyberbullying **cases,** was used to evaluate the proposed system. To ensure robust feature extraction, preprocessing steps such as text cleaning, duplicate removal, stopword filtering, and TF-IDF vectorization were applied, along with engineered behavioral features like capitalization ratio, message length, and exclamation count. This enriched dataset was used to train and evaluate multiple models including Logistic Regression, Random Forest, Linear SVM, and a shallow MLP network.

The results demonstrated that while all models performed competitively, the best performance (highest F1-score) was achieved by *Linear SVM* , confirming the effectiveness of combining textual and behavioral cues for cyberbullying detection

Output detecting Bullying messages

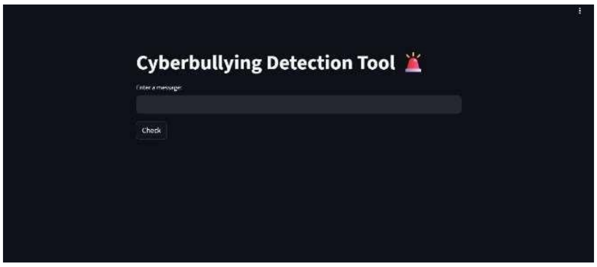
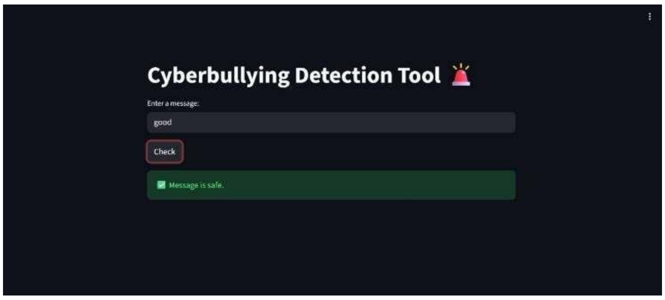

Cyberbullying Detection Tool 🚨

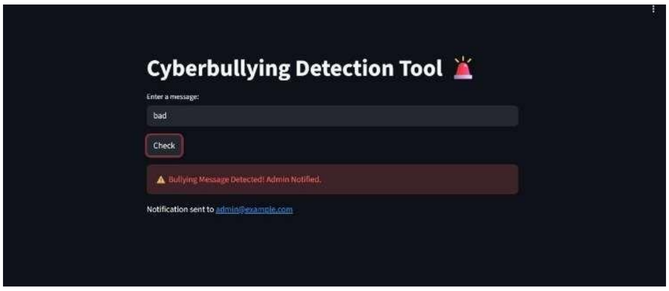
Fig 9: Project Interface


Cyberbullying Detection Tool 🚨

good

✅ Message is safe.

Fig 10: Detecting Good message


Cyberbullying Detection Tool 🚨

bad

⚠ Bullying Message Detected! Admin Notified.

Notification sent to admin@example.com

Fig 11: Detecting Cyberbullying message

XIII CONCLUSION

Compared to conventional moderation techniques, the proposed AI-based approach for cyberbullying detection represents a significant advancement. By applying machine learning and deep learning algorithms, the system overcomes the limitations of manual monitoring and keyword-based filters, offering faster, more accurate, and accessible detection of harmful online interactions. The integration of behavioral features, alongside textual analysis, increases precision by capturing subtle cues like excessive capitalization, message frequency, and sentiment polarity.

This framework supports safer online learning environments by enabling real-time alerts and providing administrators with a secure ngrok-based interface for manual interventions.

intervention, thereby safeguarding students and maintaining healthier digital communities.

Beyond reducing human workload, the system promotes early

XIII FUTURE WORKS

Future improvements will focus on integrating advanced deep learning models such as RoBERTa and BiLSTM, expanding the system's ability to detect multilingual and context-dependent bullying, and strengthening automation within the admin portal. Additionally, incorporating explainable AI will enhance transparency and trust, ensuring greater adoption in real-world educational platforms.

Subsequent developments might concentrate on incorporating this model into an intuitive real-time diagnosis application, creating hybrid models that incorporate other techniques with Random Forest, and improving explainability to guarantee the accuracy of AI predictions. This field has the potential to significantly transform dermatological care.

XIV. REFERENCES

[1] IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 14, NO. 1, JAN 2025

.

[2]. IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, VOL. 31, NO. 6, JUNE 2025

[3] IEEE TRANSACTIONS ON FAIRNESS IN AI, VOL. 28, NO. 12, DECEMBER 2024

[4] IEEE TRANSACTIONS ON WEB INTELLIGENCE, VOL. 22, NO. 4, APRIL 2020

[5]. IEEE TRANSACTIONS ON EDUCATIONAL DATA MINING, VOL. 13, NO. 1, JANUARY 2024

[6]. IEEE TRANSACTIONS ON EDUCATIONAL DATA MINING, VOL. 13, NO. 1, JANUARY 2024

[7]. IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 16, NO. 2, APRIL 2024

[8]. IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 10, NO. 1, JAN 2025

[9]. IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 10, OCT 2024

.

[10]. IEEE TRANSACTIONS ON SOCIAL NETWORKS & INFORMATION, VOL. 14, NO. 8, AUGUST 2023