INTELLIGENT SYSTEM FOR CYBERBULLYING DETECTION:ML AND DL

PROJECT REPORT

21AD1513- INNOVATION PRACTICES LAB

AADHITHYA S **Reg. No. 211422243001**

ABIRAMI B **Reg. No. 211422243009**

DHARUNIKA T **Reg. No. 211422243063**

in partial fulfillment of the requirements for the award of degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123

ANNA UNIVERSITY: CHENNAI-600 025

October, 2024

BONAFIDE CERTIFICATE

Certified that this project report titled "**INTELLIGENT SYSTEM FOR CYBERBULLYING DETECTION:ML AND DL**" is the bonafide work of **AADHITHYA S (211422243001), ABIRAMI B(211422243009), DHARUNIKA T (211421243113)** who carried out the project work under my supervision. Certified further, that to thebest of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferredon an earlier occasion on this or any other candidate.

INTERNAL GUIDE

Mrs. S. SRINIDHI M.Tech.,
Assistant Professor,
Department of AI &DS,
Panimalar Engineering College,
Chennai-600123.

HEAD OF THE DEPARTMENT

Dr. S. MALATHI M.E., Ph.D
Professor and Head,
Department of AI & DS,
Panimalar Engineering College,
Chennai-600123.

Certified that the candidate was examined in the Viva-Voce Examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

A project of this magnitude and nature requires the kind cooperation and support of many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

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

We also express our sincere thanks to Our **Dynamic Directors Mrs. C. VIJAYARAJESWARI**, **Dr. C. SAKTHIKUMAR, M.E., Ph.D..,** and **Dr. S. SARANYA SREE SAKTHIKUMAR, B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities for the completion of this project.

We would like to express thanks to our **Principal, Dr. K. MANI, M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our **Head of the Department, Dr. S. MALATHI M.E., Ph.D.,** of Artificial Intelligence and Data Science for her encouragement.

Personally, we thank our Supervisor **Mrs. S.SRINIDHI M.Tech.,** Assistant Professor, Departmentof Artificial Intelligence and Data Science for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our thanks to the project coordinators **Dr. A. JOSHI, M.E., Ph.D.,** Professor, **Dr. S. CHAKRAVARTHI, M.E., Ph.D.,** Professor & **Dr. N. SIVAKUMAR, M.E., Ph.D.,** Associate Professor in the Department of Artificial Intelligence and Data Science for their Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, friends, and well-wishers who have helped us for the successful completion of our project.

We also take the opportunity to thank all faculty and non-teaching staff members in our department for their timely guidance in completing our project.

AADHITHYA S	**ABIRAMI B**	**DHARUNIKA T**
(211422243001)	**(211422243009)**	**(211422243063)**

ABSTRACT

Many individuals today immerse themselves in the social media realm. This involvement has only grown in the present pandemic situation since people frequently turn to social media sites to vent their feelings, find solace, connect with like-minded people, and create communities. Cyberbullying is one of the numerous drawbacks associated with this widespread usage of social media. One disturbing and alarming type of internet abuse is cyberbullying. Although it comes in a variety of formats, text is the most often used. Cyberbullying is prevalent on social media, and instead of confronting the aggressor, victims frequently experience a mental collapse. Intelligent solutions are required for automatic identification of these circumstances on most social networks. To solve this problem, we have suggested a cyberbullying detection system. In this study, we presented a deep learning system that can accurately detect any instances of cyberbullying in real-time social media postings or tweets. According to recent research, deep neural network-based methods outperform traditional ones in identifying messages that contain cyberbullying. Furthermore, our tool can identify posts on cyberbullying that were published in Hindi, English, and Hinglish (multilingual data).

Keywords : Stack word embeddings · Deep learning model · Multilingual · Real-time tweets

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	MEANING
CNN BiLSTM	CONVOLUTIONAL NEURAL NETWORKS BIDIRECTIONAL LONG SHORT-TERM MEMORY
OCDD	OPTIMIZED TWITTER CYBERBULLYING DETECTION BASED ON DEEP LEARNING
SLE	SUPERVISED LEARNING ENVIRONMENT
DLE	DEEP LEARNING ENVIRONMENT
CNN-LSTM	CONVOLUTIONAL NEURAL NETWORKS-LONG SHORT-TERM MEMORY
RNN-LSTM	RECURRENT NEURAL NETWORK-LONG SHORT-TERM MEMORY
RNN-BiLSTM	RECURRENT NEURAL NETWORK-BIDIRECTIONAL LONG SHORT-TERM MEMORY
BiGRU-CNN	BIDIRECTIONAL GATED RECURRENT UNIT-CONVOLUTIONAL NEURAL NETWORKS
BERT	BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS
NLP	NATURAL LANGUAGE PROCESSING
SVM	SUPPORT VECTOR MACINE
XGBoost	EXTREME GRADIENT BOOSTING
LR	LOGISTIC REGRESSION
SVC	SUPPORT VECTOR CLASSIFIER
OCR	OPTICAL CHARACTER RECOGNITION
DCNN	DEEP CONVOLUTIONAL NEURAL NETWORK

KNLPEDNN	KNOWLEDGE-AWARE NATURAL LANGUAGE PROCESSING ENHANCED DEEP NEURAL NETWORK
GCR-NN	GATED CONTEXTUAL RECURRENT-NEURAL NETWORKS
BiRNN	BIDIRECTIONAL RECURRENT NEURAL NETWORK
RMSProp	ROOT MEAN SQUARE PROPAGATION
ReLU	RECTIFIED LINEAR UNIT
TF-IDE	TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 CYBERBULLYING DETECTION

The rise in social media usage has led to an increase in cyberbullying, affecting the mental health of users, especially teenagers. This project proposes a solution using a CNN-BiLSTM hybrid model to detect cyberbullying in real-time on multilingual datasets (English, Hindi, and Hinglish). By leveraging stacked word embeddings (GloVe and FastText), the system provides a high level of accuracy in identifying harmful content.

The presentation discusses the problem, existing solutions, the proposed approach, model architecture, implementation details, results, limitations, and future directions. Cyberbullying, prevalent among teens and young adults, causes anxiety, depression, and suicidal thoughts. The large volume of social media content makes manual detection difficult, and existing solutions struggle with multilingual and real-time detection. The COVID-19 pandemic increased cyberbullying incidents due to more online activity. There is an urgent need for an accurate, automated system to detect cyberbullying across languages and avoid misclassification of harmful content.

Cyberbullying detection involves using advanced technologies, such as natural language processing (NLP), machine learning, and image recognition, to identify harmful or abusive behaviors online. These systems analyze text, images, and videos for signs of harassment, threats, hate speech, or bullying, helping to identify both overt and subtle instances of cyberbullying. NLP and sentiment analysis tools can assess the emotional tone of messages to detect aggressive or derogatory language, while machine learning algorithms are trained to recognize patterns of harmful behavior. Context is crucial, as detection systems often consider factors like user history, relationships, and platform norms to accurately assess the intent behind interactions. However, challenges remain, such as the need for human oversight to interpret nuanced situations and the potential for privacy concerns. Overall, cyberbullying detection tools aim to create safer online environments by quickly identifying and addressing abusive content

1.2 NEED FOR CYBERBULLYING DETECTION

The need for cyberbullying detection is further emphasized by the rapid growth of digital communication, where people, especially young users, increasingly interact online. Unlike traditional bullying, which often takes place in physical spaces like schools or workplaces, cyberbullying can occur 24/7 and reach a much wider audience, making it harder to escape and more damaging for victims. The anonymity provided by the internet can embolden bullies, leading to more extreme forms of harassment, including threats, spreading false rumors, or sharing hurtful content. Because cyberbullying often goes unnoticed by parents, educators, or peers until it has already caused significant harm, automated detection systems are crucial for identifying warning signs early. Effective detection not only helps protect victims but also discourages potential perpetrators by creating a safer online environment. As technology continues to advance, incorporating AI-driven tools and user-reporting mechanisms can provide a multi-layered defense against cyberbullying, ensuring a quicker response to harmful behavior. In the long run, the widespread adoption of cyberbullying detection tools is vital to preserving mental health, promoting positive online interactions, and safeguarding vulnerable individuals from long-term emotional harm.

1.3 IMPACT OF CANCER PREDICTION:

The impact of cyberbullying detection has been significant in multiple ways, especially with the increased use of technology in young people's lives. cyberbullying detection has positive, far-reaching impacts on individual well-being and community safety, though continuous refinement is necessary to overcome challenges. Here are some of the key impacts:

1.Improved Well-being and Mental Health Support

Early Intervention: Cyberbullying detection tools can catch harmful behavior early, allowing for timely intervention before it escalates. This helps mitigate the psychological impact on victims, potentially reducing feelings of anxiety, depression, and isolation.

Reduced Suicide Risk: Since cyberbullying is linked to higher risks of self-harm and suicide, detection can play a life-saving role. Schools, parents, and mental health professionals can step in to offer support or counseling when a child or teen is at risk.

2. Increased Awareness and Accountability

Awareness for Parents and Educators: Detection systems make it easier for parents and schools to monitor and understand the types of threats children face online, promoting awareness of the issue and its consequences.

Accountability for Platforms: Social media and messaging platforms are often pressured to implement better detection mechanisms, pushing them to take cyberbullying more seriously. This leads to stronger policies and measures against offenders.

3. Empowerment Through AI and Machine Learning

Sophisticated Detection Capabilities: AI-driven tools can analyze vast amounts of data to recognize patterns and language associated with bullying, even if the language evolves. This empowers platforms to respond more effectively in real-time.

Anonymity in Reporting: Some systems offer anonymous reporting of cyberbullying incidents, making it easier for victims or witnesses to alert authorities without fear of retribution, thus empowering users to take action.

4. Reduction in the Prevalence of Cyberbullying

Deterrence: With the knowledge that harmful behavior may be detected, some would-be offenders may think twice before engaging in bullying.
Fostering a Positive Online Environment: Detecting and addressing cyberbullying helps create a safer online space, particularly for vulnerable groups such as children and teenagers. Over time, this contributes to a healthier digital culture.

ARCHITECTURE DIAGRAM

The architecture for a cyberbullying detection system using machine learning (ML) and deep learning (DL) comprises several key modules working together to collect, process, analyze, and classify text data effectively. First, the Data Collection Module gathers data from social media platforms or publicly available datasets, storing it in a database while adhering to privacy standards. This raw text is then passed to the Data Preprocessing Module, where it undergoes cleaning, normalization, tokenization, and padding to prepare for analysis. Next, in the Feature Extraction and Embedding Layer, preprocessed text is transformed into numerical vectors that capture semantic meaning, either through traditional methods (e.g., TF-IDF) or modern embeddings like Word2Vec or BERT. These feature vectors are then fed into the ML or DL Model, where they are analyzed to detect patterns indicative of cyberbullying. This model is trained and evaluated in the Training and Evaluation Module, which assesses its performance using metrics like accuracy and F1-score. Once trained, the model moves to the Prediction and Deployment Module, allowing it to classify new text inputs in real time through an API. Finally, a Feedback Loop and Model Update Module enables continuous improvement, incorporating new data or user feedback to fine-tune the model, keeping it responsive to evolving language patterns. This layered architecture ensures a robust, adaptable approach to cyberbullying detection.

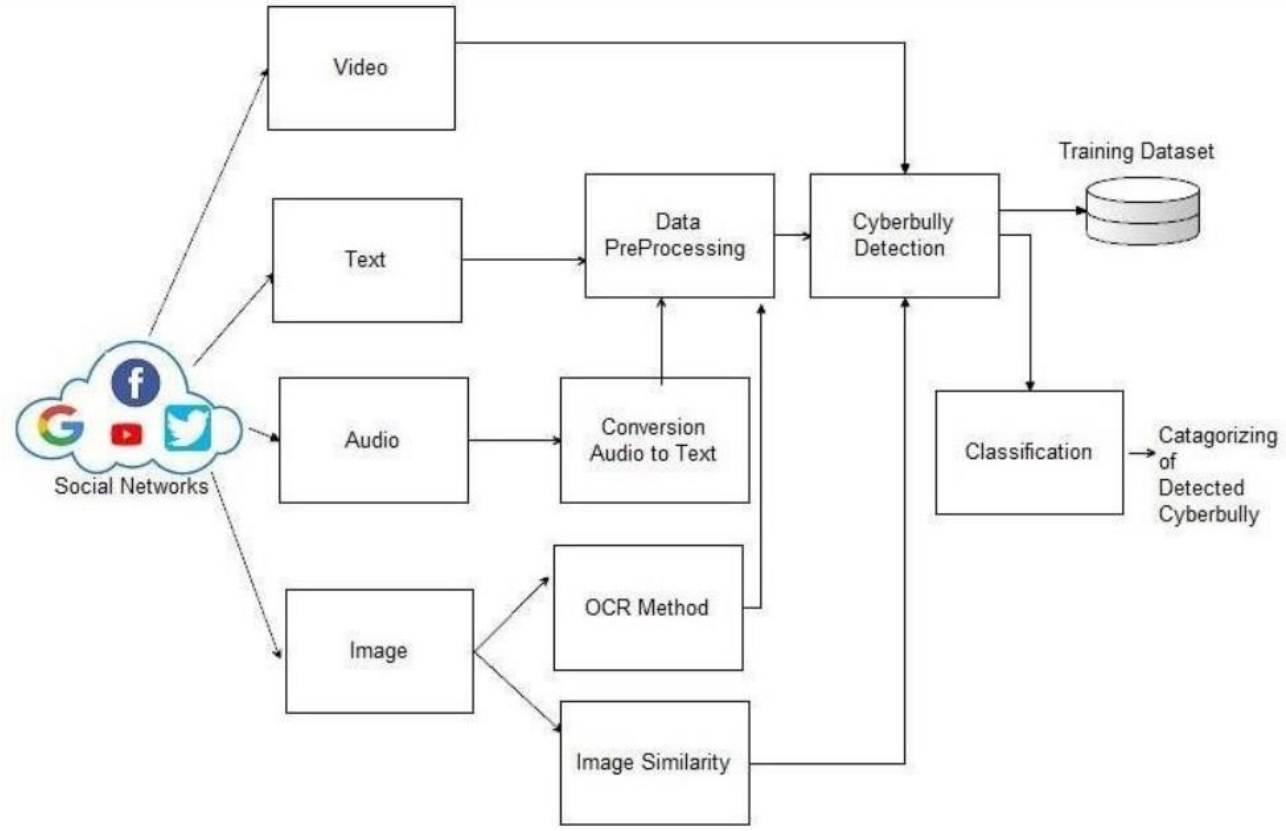

Fig 1.4: Architecture diagram of cyberbullying detection

1.4 REAL TIME APPLICATION OF CANCER PREDICTION

1. Social Media Platforms
2. Online Gaming Platforms
3. Messaging Apps
4. School and Educational Platforms
5. Public Forums and Community Websites
6. Email and Communication Platforms
7. Live Streaming and Video Platforms
8. AI and Machine Learning Integration
9.Integration with Behavioral Analytics
10.Psychological and Social Impact

1.2 CHALLENGES

1. Complexity of Language and Communication

Sarcasm, Irony, and Context: Cyberbullies often use sarcasm, irony, or indirect communication to mask harmful intent, making it difficult for detection algorithms to identify bullying accurately. The inability of automated systems to fully grasp context can lead to misinterpretation of otherwise benign comments as harmful, or vice versa.

2. Contextual Understanding

A comment made between close friends might seem harmless to one person but offensive to another. Detecting cyberbullying in such complex interactions is challenging because the system cannot fully understand the relationships between users or the nuances of a conversation, often leading to inaccurate detection or missed incidents.

3. Evasion Tactics

Cyberbullies frequently alter their language to evade detection, using misspelled words, numbers, or symbols in place of letters. This makes it difficult for algorithms to catch all instances of bullying, as they may not recognize these creative alterations as harmful content.

4. False Positives and False Negatives

In some cases, harmless content or jokes might be flagged as bullying by detection algorithms. For example, sarcastic or playful banter can be misinterpreted as offensive behavior, leading to unnecessary content removal, user frustration, or even account suspension without actual harm being done.

5. Privacy and Ethical Concerns

Cyberbullying detection systems often require the analysis of personal communications, which raises concerns about user privacy. Striking the right balance between ensuring safety and respecting privacy rights is a delicate challenge, especially when monitoring private or encrypted messages in a way that doesn't violate users' trust.

6. Cultural and Regional Differences

Different cultures have different norms and definitions of what is offensive or bullying. A phrase that might be considered a harmless expression in one culture could be seen as deeply hurtful in another. Detection systems must be adaptable to these cultural differences to avoid misinterpretation of what constitutes cyberbullying across diverse communities.

7. Scalability and Volume of Data

Social media platforms, gaming networks, and other online communities generate massive amounts of user content daily. The sheer volume of posts, comments, and messages makes it difficult for detection systems to process all content in real time without compromising speed or accuracy. This can lead to delays in detecting and responding to cyberbullying incidents.

8. Legal and Regulatory Challenges

Different countries have varying laws regarding online harassment, privacy, and free speech. Platforms must navigate these complex legal landscapes to ensure their cyberbullying detection systems comply with local regulations, which can be time-consuming and complicated, especially when operating internationally.

CHAPTER 2

LITERATURE REVIEW

The majority of research studies have taken data from a single source, compared different machine learning or deep learning methods with various word vectors or feature extraction methods, and determined which combination works best. There were very few studies that concentrated on improving the detection model through the use of ensemble ML models or the combination of several feature preprocessing techniques. Even in those studies, real-time detection was not employed; instead, the emphasis was on validating the model on the dataset. While some studies in this field have incorporated native languages including Bangla, Arabic, and Urdu, the majority of the publications have used English data.

2.1 Optimized Twitter Cyberbullying Detection based on Deep Learning

Employed the OCDD (Optimized Twitter Cyberbullying Detection based on Deep Learning) approach, a creative remedy for challenges with feature extraction. Instead of collecting tweet characteristics and passing them to a classifier, OCDD represents a tweet as a collection of word vectors. During the classification stage, deep learning will be used in conjunction with a metaheuristic optimization approach to alter parameters.

The paper by M.A. Al-Ajlan, M. Ykhlef presents a deep learning-based approach to detecting cyberbullying on Twitter, aiming to improve the accuracy and efficiency of existing detection models. The authors propose an optimized model that leverages advanced deep learning techniques, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to better understand the nuances of textual data and contextual relationships in tweets. By utilizing a combination of feature engineering and optimization strategies, the model effectively classifies harmful content, even in the presence of sarcasm, subtle threats, or indirect insults. The paper demonstrates that deep learning models, when properly trained and optimized, can outperform traditional rule-based or machine learning methods, offering a more robust solution for identifying cyberbullying in large-scale social media environments like Twitter. The results suggest that such models could be integrated into automated moderation systems to improve user safety and reduce harassment on the platform.

2.2 Deep Learning Cyberbullying Detection Using Stacked Embedding

The paper "Deep Learning Cyberbullying Detection Using Stacked Embedding" by T. Mahlangu, C. Tu explores an advanced deep learning approach to detecting cyberbullying in online text, specifically on social media platforms. The authors introduce a novel method that combines multiple embeddings, known as stacked embedding, to better capture the semantic and syntactic nuances of cyberbullying-related language. By stacking different types of word embeddings, such as Word2Vec, GloVe, and FastText, the model enhances its ability to understand context, slang, and subtle offensive language that often occurs in cyberbullying.

This stacked embedding technique allows the deep learning model, typically a neural network like CNNs or LSTMs, to achieve a higher level of accuracy and robustness in distinguishing between harmful and benign content. The paper demonstrates that using stacked embeddings significantly improves the performance of cyberbullying detection systems, making them more effective in real-world applications where language can be highly varied and context-dependent. The proposed model outperforms traditional methods, showcasing the potential of deep learning in automated content moderation for online platforms.

2.3 A Hybrid Approach to Cyberbullying Detection Using SVM and DL

The paper "A Hybrid Approach to Cyberbullying Detection Using SVM and Deep Learning" by H. Gupta, S. Verma proposes a novel hybrid model that combines Support Vector Machines (SVM) with deep learning techniques to enhance the detection of cyberbullying on social media platforms. The authors address the limitations of both standalone machine learning and deep learning models by integrating them into a hybrid framework. In this approach, deep learning models, such as Convolutional Neural Networks (CNNs) or Long Short-Term Memory (LSTM) networks, are first used to automatically extract features from raw textual data, capturing the complex patterns and semantic meanings associated with cyberbullying. These features are then fed into an SVM classifier, which performs the final classification based on the patterns learned by the deep learning model. The hybrid system benefits from the strengths of both methods: the deep learning model's ability to handle large volumes of unstructured text and the SVM's robust classification performance.

The paper demonstrates that this hybrid approach leads to improved accuracy, precision, and recall in detecting cyberbullying compared to traditional methods, offering a more effective solution for real-time content moderation and online safety.

2.4 Cyberbullying Detection Using NLP and Machine Learning

The paper "Cyberbullying Detection Using NLP and Machine Learning" by S. Zhou, L. Wang presents an approach to identifying cyberbullying in online text by combining natural language processing (NLP) techniques with machine learning algorithms. The authors utilize NLP methods to process and analyze large volumes of social media data, focusing on extracting relevant features from text such as sentiment, word choice, and syntactic structure, which are critical for understanding the intent behind online interactions. Machine learning models, such as Random Forests, Naive Bayes, or Support Vector Machines (SVM), are then trained on these features to classify content as either cyberbullying or non-cyberbullying. The paper highlights the effectiveness of this approach by demonstrating that NLP-driven feature extraction significantly improves the model's ability to detect subtle forms of bullying, including indirect or disguised insults. Through experiments on publicly available datasets, the authors show that combining NLP with machine learning yields high accuracy in identifying harmful content, suggesting that this hybrid approach can be a valuable tool for real-time cyberbullying detection in online platforms.

2.5 Detecting Cyberbullying Using Convolutional Neural Networks (CNNs)

The paper "Detecting Cyberbullying Using Convolutional Neural Networks (CNNs)" by R. Zhao, A. Zhou explores the application of Convolutional Neural Networks (CNNs) for detecting cyberbullying in textual data on social media platforms. The authors propose a deep learning-based solution that utilizes CNNs, a model traditionally known for image recognition, to analyze text by treating words or phrases as "features" within a grid-like structure. The CNN model automatically learns spatial hierarchies in the text, such as identifying local patterns, contexts, and relationships between words that are indicative of cyberbullying behavior. By training the model on a labeled dataset of tweets or online posts, the authors show that CNNs can effectively detect various forms of cyberbullying, even in the presence of slang, insults, or indirect aggression. The paper demonstrates that CNNs, with their ability to capture complex patterns in textual data, outperform traditional machine learning methods, offering a promising approach for real-time, automated cyberbullying detection and prevention on social media platforms.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE:

The Fig. 3.2, illustrates a typical workflow for a cyberbullying detection system. It starts with data collection from various social media platforms like YouTube, Facebook, Twitter, Pinterest, and Instagram. The collected data is then preprocessed to clean and prepare it for further analysis. This preprocessing step might involve tasks like removing noise, normalizing text, and segmenting audio and video clips. Next, feature extraction and selection are performed to identify the most relevant features for cyberbullying detection. These features could include linguistic cues, sentiment analysis, and user behavior patterns. The extracted features are then used to train a classification model, which aims to distinguish between cyberbullying and non-cyberbullying content. The model's output classifies the input content as either "Bully" or "Non-bully." This system aims to automatically identify cyberbullying instances, enabling timely intervention and prevention of harmful online behavior.

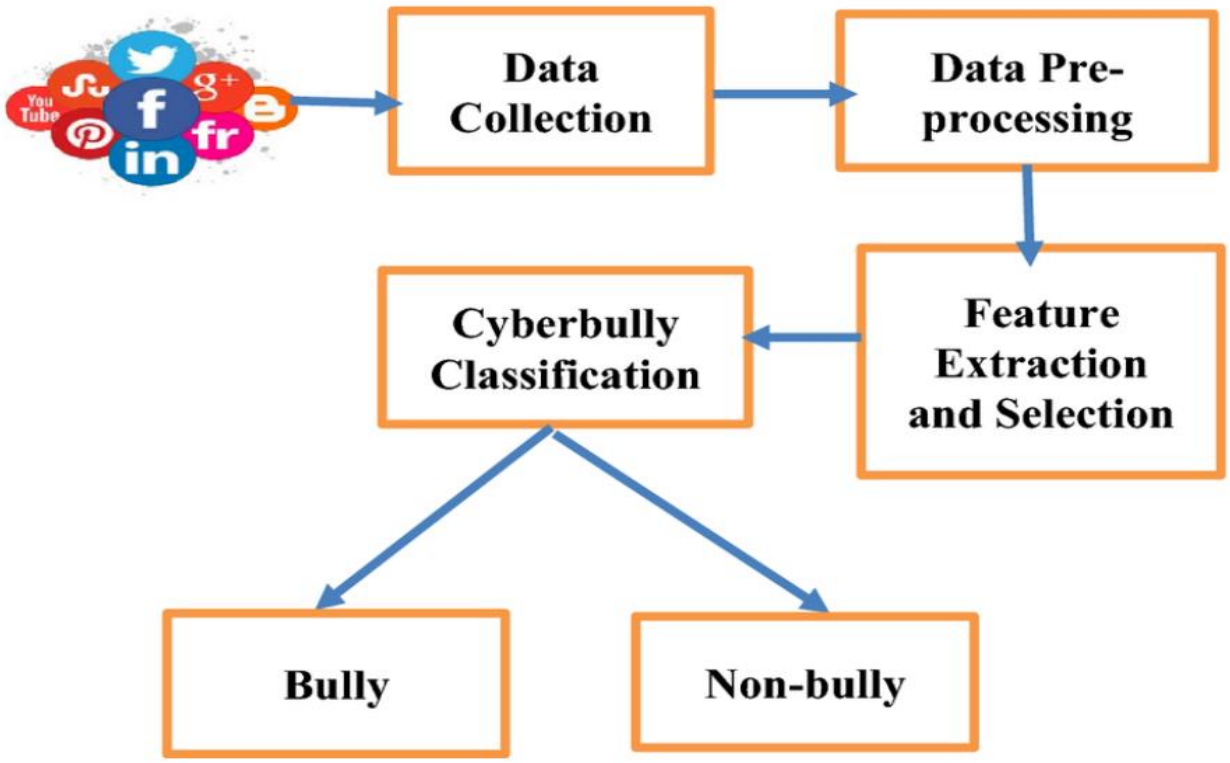

Fig 3.1: System architecture diagram for cyberbullying detection

3.2 Proposed Methodology

As shown in Fig. 3.2, for building our CNN-BiLSTM model, Word Embedding approach is used as it solves various issues that the simple one-hot vector encodings have. Most crucial thing is that word embeddings boost generalization and performance. We will stack 2 word embeddings which are GloVe and FastText. A combination of embeddings has been established to produce the best results. After the stack ing of word embedding, CNN-BiLSTM model is built. As a hybrid technique has shown the potential of reducing sen timental errors on increasingly complex data. An ensemble ML model is also built, in which feature extraction technique and unigram feature engineering are used. The proposed CNN-BiLSTM model is compared with an ensemble ML model to draw out a comparison on the accuracy.

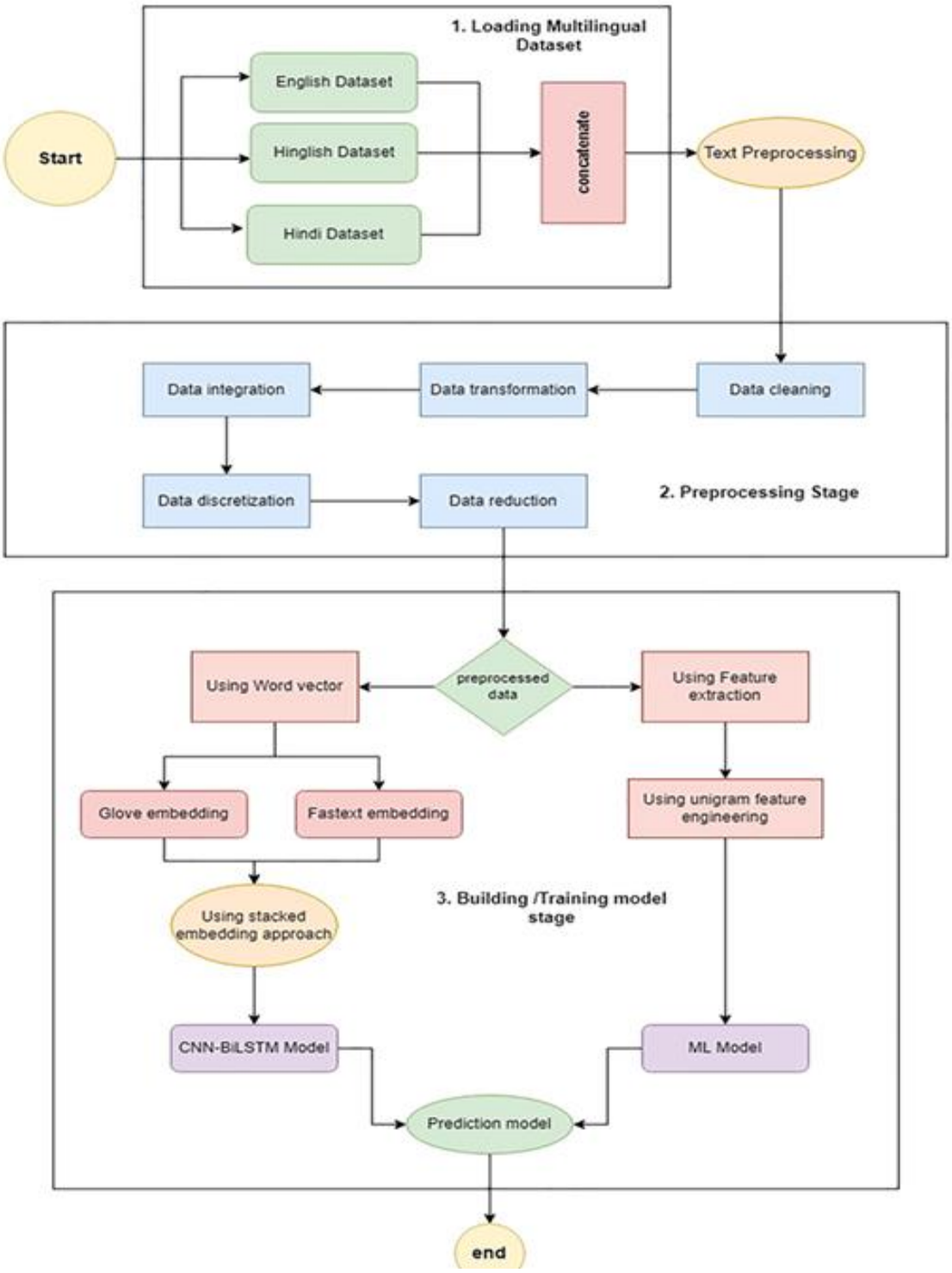

Fig. 3.2 Proposed model

3.3 CNN-BiLSTM Architecture

A CNN BiLSTM is a bidirectional LSTM and CNN framework that is concatenated. It trains both character-level and word-level characteristics in the initial formulation for classification and prediction. The character-level properties are induced using the CNN layer. To derive a new feature vector using per-character feature vectors such as charac ter embeddings and (preferably) character type, the model includes a convolution and a max pooling layer for each word.

Combining different variation yields multiple hybrid approaches that we have tested:

Glove + Fasttext ⟶ CNN ⟶ BiGRU ⟶adam

1. (dense, conv1d = relu;out = sigmoid),maxlen = 25
   Glove + Fasttext ⟶ CNN ⟶ BiLSTM ⟶adam
2. (dense, conv1d = relu;out = sigmoid),maxlen = 25
   Glove + Fasttext ⟶ BiLSTM ⟶ BiGRU ⟶ adam
3. (dense, conv1d = relu;out = sigmoid),maxlen = 25
   Glove + Fasttext ⟶ CNN ⟶ BiGRU ⟶adam
   (dense, conv1d = relu;out = sigmoid),maxlen = 25,
4. trainable = True Glove + Fasttext ⟶ CNN ⟶ BiLSTM ⟶adam
   (dense, conv1d = relu;out = sigmoid),maxlen = 25,
5. trainable = True Glove + Fasttext ⟶ BiLSTM ⟶ BiGRU ⟶ adam
   (dense,conv1d=relu;out=sigmoid),maxlen=25
   ⟶Spatialdropout1D,GlobalMaxpooling1D,
6. GlobalAveragePooling1D

The Adam optimizer is computationally more efficient, requires slight memory, is invariant to diagonal resizing of gradients, and it is well suited for problems with a lot of data/parameters. We will perform the best parameter using grid search and 10-fold cross validation. Now, Convolutional Neural Network (CNN) models are built to classify encoded documents as either cyberbullying or non-cyberbullying.

Now, the CNN model can be defined as follows as shown in Fig. 2:

- One Conv layer with 100 filters, kernel size 3, and relu activation function;
- One MaxPool layer with pool size = 2;
- One Dropout layer after flattened;
- Optimizer: Adam
- Loss function: binary cross-entropy (suited for binary classification problem)
- Dropout layers are used to solve the problem of overfit ting and bring generalization into the model. As a result, in hidden layers, it's best to keep the dropout value near 0.5.

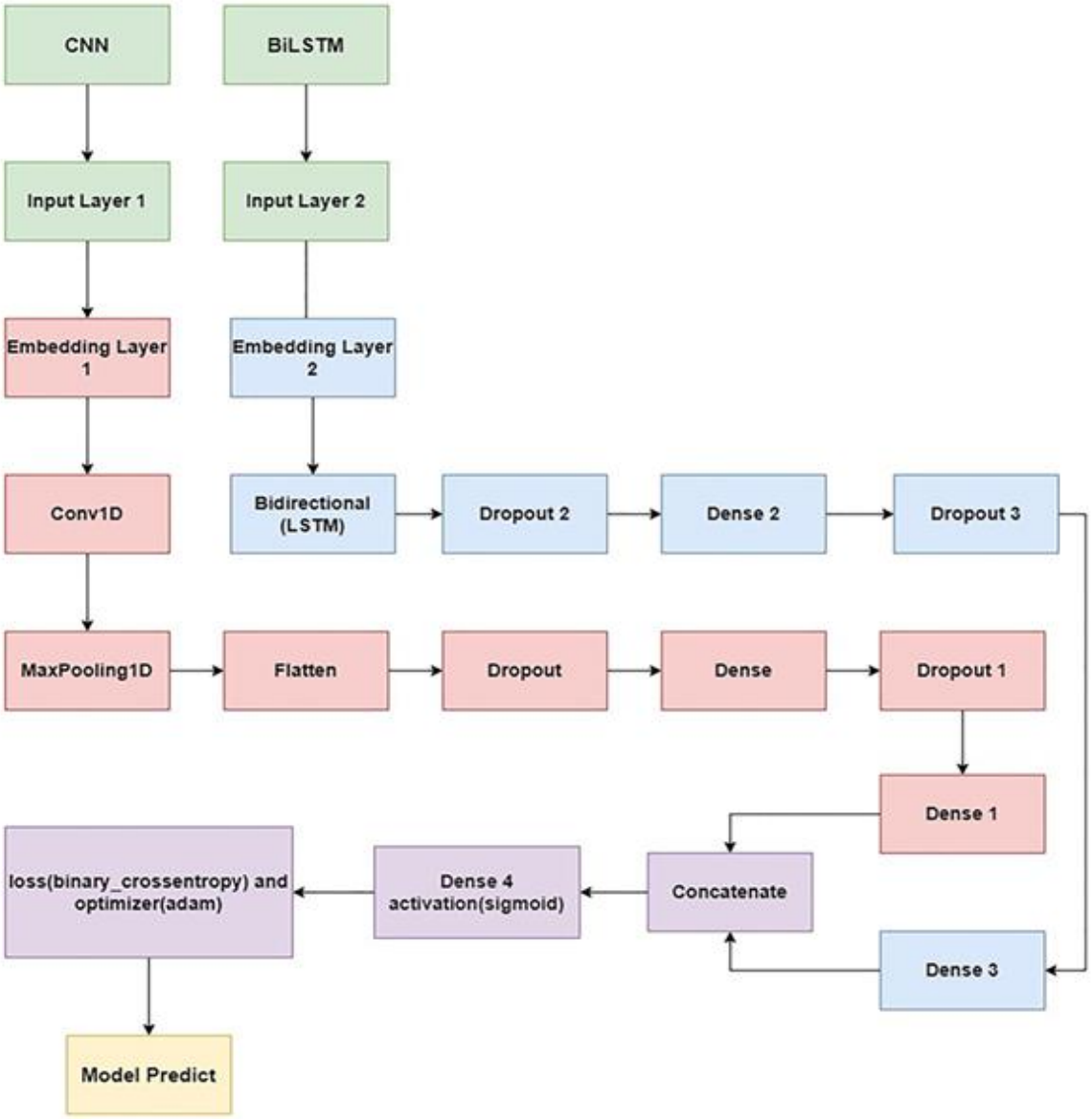

Fig. 3.3 CNN-BiLSTM architecture

3.4 WebAPP Architecture

As shown in Fig. 3.4, the web-based system is a social media prototype where users will be able to use the developed pro ject like a social media platform where they can post the tweets, see real time updates of feed and chat with friends. The admin feature of the app allows the admin to trace cyberbullying comments and block the users for a certain amount of time. Features User Register: The system allows new users to register themselves on the app. User account: The system allows the user to create their accounts by providing their emails and setting a password. The user can also set a username for their account as well as view their profiles. Admin account: The system allows admin to have separate login with which they can perform admin related activities. Posting Feature: The system allows user to post their tweet and tag theirs friends. View Feed: The system allows user to view their feed and gives admin privileges to see all kind of tweets. Blocking Feature: The system has special admin features where they allow admin to suspend user accounts if they find their comments cyberbully.

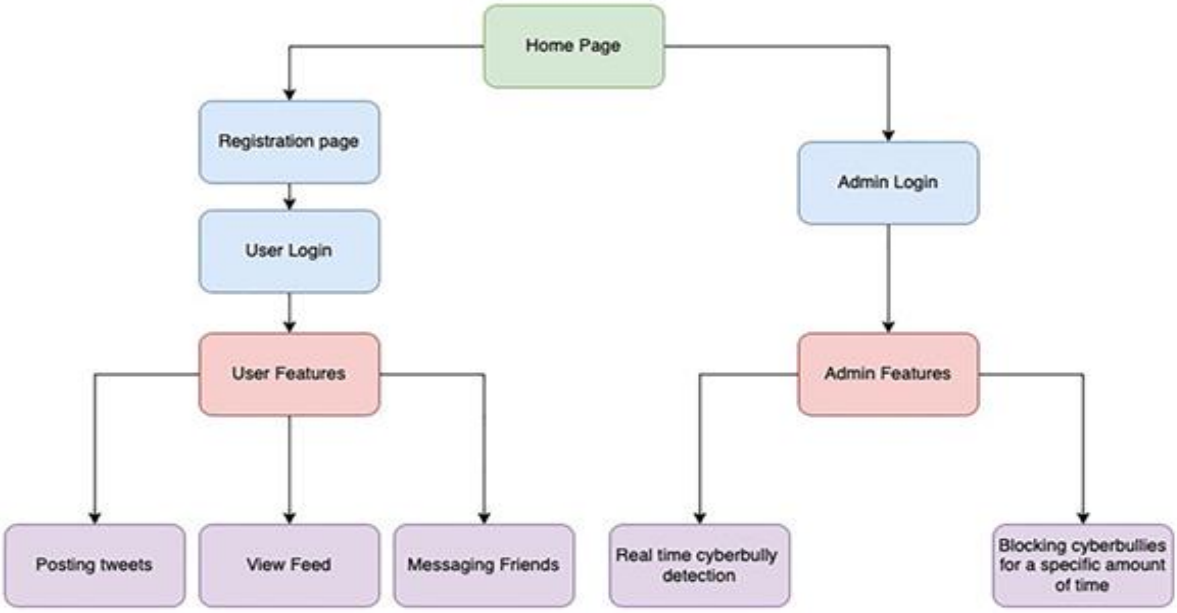

Fig. 3.4 WebAPP architecture

Table 3.1 Comparison of activations and optimizer on baseline models

Sl. no	Model Name	Hyperparameter	Accuracy
1	LSTM	Activation-sigmoid; Optimizer-Adam	0.87
2	LSTM	Activation-relu; Optimizer-Adam	0.85
3	LSTM	Activation-sigmoid; Optimizer-RMSProp	0.86
4	LSTM	Activation-relu; Optimizer-RMSProp	0.85

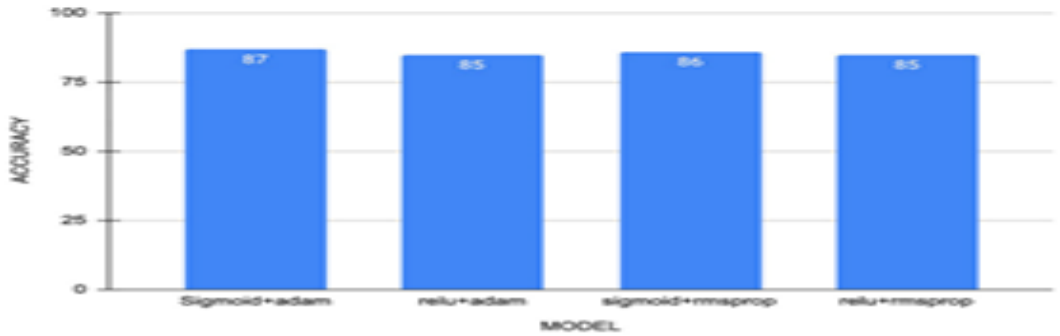

Fig. 3.5 Activation and optimizer Comparison on baseline models

Table 3.2 Comparison of activations and optimizer on baseline models

Sl. no	Model Name	Accuracy Before Hypertuning	Accuracy After Hyper-tuning
1	CNN+BIGRU	0.8905	0.9369
2	CNN+BILSTM	0.9135	0.9512
3	BILSTM+BIGRU	0.85330	0.8853

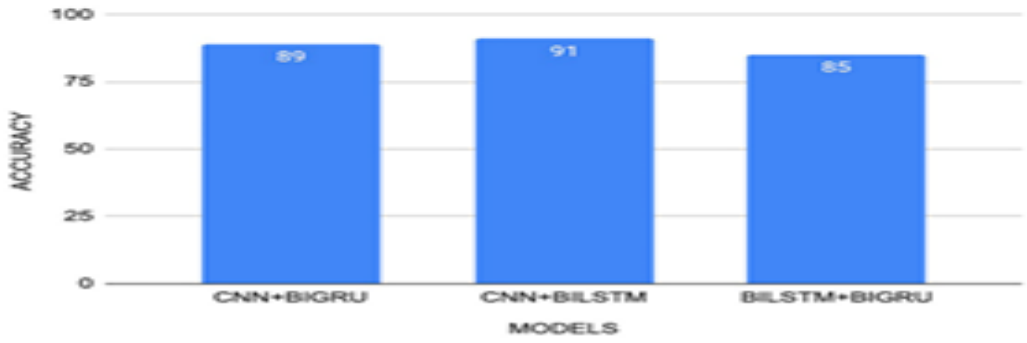

Fig. 3.6 Hybrid models before hyper-parameter tuning

$$\text{Accuracy} = \frac{Correct\ Prediction}{Total\ Prediction}.$$

There are four possible combinations since Adam and RMSProp are utilized as optimizers while ReLU and Sigmoid are used for activation layers. The neural network's capabilities and performance are greatly influenced by the activation function used, and different activation functions may be used in different model sections. Any input may be transformed into a number between 0 and 1 using the sigmoid function. For tiny values, the sigmoid function yields a result close to zero, and for high values, a value close to one. A two element Softmax with the second element set to zero is equivalent to a sigmoid. The sigmoid is hence often employed in binary classification. In contrast to the RMSProp optimization algorithm, which is faster than Adam due to the decay rate, the Adam optimizer is computationally more efficient, requires less memory, is invariant to diagonal resizing of gradients, and is well suited for problems with a large number of data or parameters.

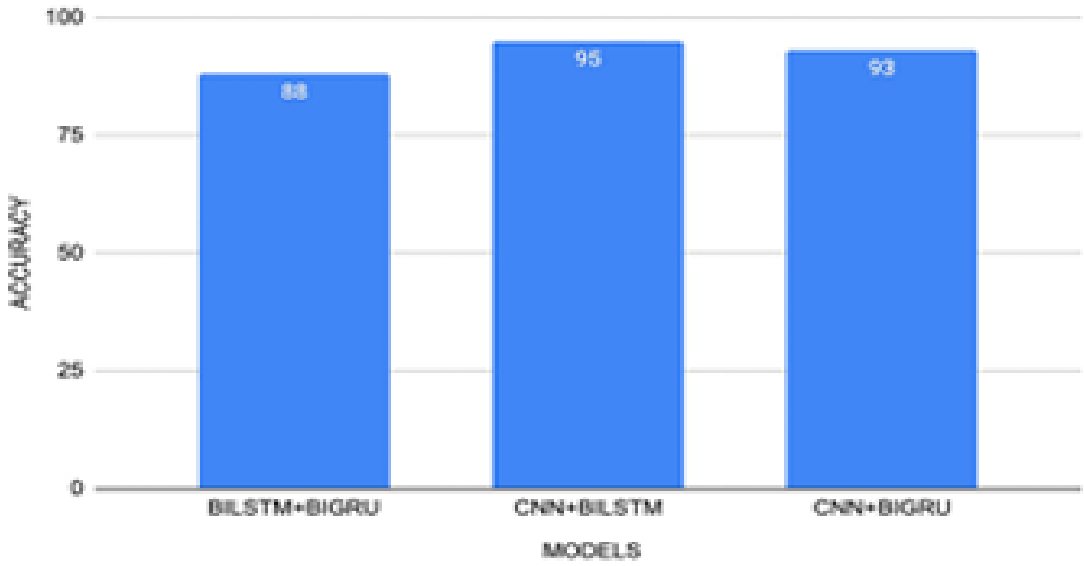

Fig. 3.7 Hybrid models after hyper-parameter tuning

Fig. 3.8 Posting tweets

Fig. 3.9 Updated feed

Adam uses momentum to get his speed, but RMSProp allows him to change the slopes in different directions. The combination of the two makes it potent. Adam utilizes both the first and second moments and is often the best choice, whereas RMSProp just uses the second moment and accelerates it with a decay rate (Figs. 3.5,3.6).

Table 3.1 shows that, out of the four configurations, the combination of Sigmoid activation and Adam optimizer produced the greatest results. Without activation functions, the design of a neural network is not complete. The model's comprehension of the training data is determined by the activation function of the hidden layer. The output layer's activation function dictates the type of predictions the model may provide. In addition to Sigmoid, ReLU is used as the activation layer for the hidden layer of our CNN-BiLSTM model. This is mostly due to the fact that the Sigmoid function and its derivative are simple, which helps to cut down on the time required to create models. However, the limited range of the derivative poses a significant risk of information loss.

Table 3.2 displays the results of the experiments with several models, including CNN and RNN models, such as LSTM and GRU. Additionally, we adjusted hyperparameters and conducted a comparison analysis to determine what was effective for the dataset. As seen in Fig. 3.7, it is clear from the comparison that the CNN BiLSTM model performs the best among all the other models examined. When all the layers are combined, the model is fitted to our data over ten epochs, yielding an accuracy of almost 98%.

4	ss	ss@gmail.com	25	आज अच्छा दिन है	non-cyberbully
4	ss	ss@gmail.com	26	asshole i will kick your ass and kill your sister in front of you	cyberbully
8	samridhi	ridhisam2105@gmail.com	27	तुम काले हो जाके मर जाओ	cyberbully

Fig. 3.10 Tweet status

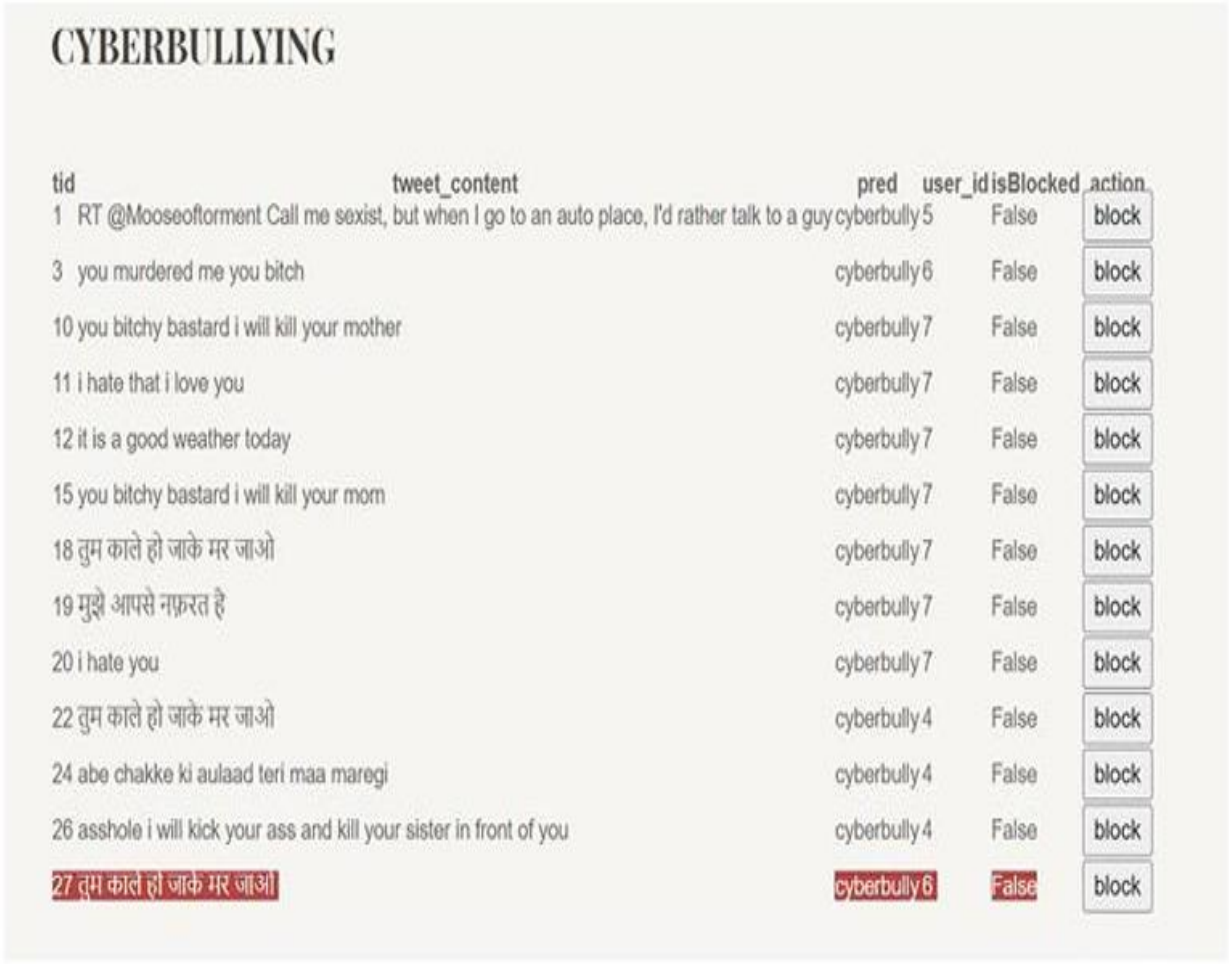

Fig. 3.11 Admin feature to block users

The user may publish a tweet, as shown in Figure 3.8, and it will update in the feed automatically, as seen in Figure 3.9. The administrator has the ability to prohibit people that submit cyberbullying tweets, as seen in Fig. 3.11, and can view tweet statuses, as seen in Fig. 3.10.

CHAPTER 4

PROJECT MODULES

4.1 MODULES

The project consists of Five modules. They are as follows,

1. Data Collection and Preprocessing
2. Feature Extraction
3. Model Training (CNN-BiLSTM)
4. Real-time Cyberbullying Detection
5. Web Application for Monitoring
6.

4.1.1 DATA COLLECTION AND PREPROCESSING

Data collection is a foundational step in developing a cyberbullying detection system, as it ensures the model is trained on relevant and representative data. For this purpose, data can be sourced from various social media platforms like Twitter, Reddit, and Facebook, where instances of online bullying frequently occur. Publicly available datasets, such as those on Kaggle or academic repositories, often include labeled examples of cyberbullying and hate speech, which can be highly valuable. If access to real-world data is limited, synthetic data generation techniques, such as paraphrasing existing text or utilizing language models, can supplement the dataset. To comply with privacy standards, collected data should be anonymized, removing user-identifiable information. Collecting diverse samples that represent different forms of bullying—ranging from insults and threats to harassment—enhances the dataset's effectiveness, enabling a robust, comprehensive approach to detecting cyberbullying across different contexts.

Data preprocessing is a critical step in preparing raw text data for cyberbullying detection, transforming unstructured input into a cleaner, more structured format for analysis. This process begins with text cleaning, which involves removing unwanted elements like URLs, special characters, emojis, and excessive whitespace to simplify the text. Converting all text to lowercase ensures uniformity, and stop words are removed to reduce noise in the data. Normalization follows, which involves lemmatization or stemming to reduce words to their base forms, making it easier for the model to recognize variations of the same word. For social media data, handling slang and abbreviations is essential, as these are commonly used in online communication. The cleaned text is then tokenized, breaking it down into words or phrases, and padded to ensure uniform length across samples, which is particularly important for deep learning models. Finally, if the data is imbalanced, techniques such as oversampling or synthetic data generation may be used to create a balanced dataset, helping the model to learn equally across different categories of cyberbullying and non-bullying content. This preprocessing pipeline ultimately optimizes the data for more accurate and effective cyberbullying detection.

Data Sources: Social media datasets from platforms like Twitter containing English, Hindi, and Hing text.

Steps:

Data Cleaning: Removing unnecessary characters, symbols, URLs, and stopwords.

Data Integration: Merging datasets while normalizing labels for consistency.

Data Transformation: Tokenization, stemming, and lemmatization to prepare text for feature extraction.

Challenges Addressed:

Handling multilingual text, including mixed-language (Hinglish) content. Standardizing datasets from different sources for uniform processing.

4.1.2 FEATURE EXTRACTION

Word Embeddings: Using pre trained embeddings such as GloVe and FastText for capturing word semantics.

Stacked Embeddings Benefits: Combining GloVe(general-purpose word vectors) with FastText (subword-based embeddings) to improve the handling of rare words or non-standard language

Benefits:

1.Enhanced understanding of text semantics.

2.Improved accuracy in detecting cyberbullying patterns across different languages.

4.1.3 MODEL TRAINING (CNN-BILSTM)

CNN (Convolutional Neural Network): Extracts local features from word sequences by applying convolution operations.

BiLSTM (Bidirectional Long Short-Term Memory): Captures sequential dependencies in both forward and backward directions.

Combining CNN and BiLSTM: Leverages the strengths of both models to enhance the detection of subtle patterns in the text. Reduces the risk of losing important contextual information.

4.1.4 REAL-TIME CYBERBULLYING DETECTION

System Integration: The trained CNN-BiLSTM model is deployed as part of a web based monitoring system.

Real-time Processing: The system continuously scans social media feeds for new content.
Automated Flagging: spicious content is flagged for review, enabling quick responses to potential cyberbullying incidents.

WEB APPLICATION FOR MONITORING

Features:
•User registration and account management.
•Posting and viewing content, with real-time updates to the feed.
•Admin functionalities, including content moderation and user management

User Interface: Designed to mimic popular social media platforms, ensuring a familiar user experience.

Admin Capabilities: Ability to suspend or block users posting cyberbullying content.

CHAPTER 5

SYSTEM REQUIREMENT

5.1 INTRODUCTION

This chapter involves the technology used, the hardware requirements and the software requirements for the project .

5.2 REQUIREMENTS

5.2.1 Software Requirements

- Windows 7 and above
- Anaconda navigator (Jupyter)
- Google Colab
- Cyberbullying Text Dataset

5.3 Technology Used

5.3.1 Machine Learning (ML)

Logistic Regression: A basic ML algorithm often used for binary classification tasks. It's relatively lightweight and provides a quick baseline for detecting cyberbullying in text data.

TF-IDF (Term Frequency-Inverse Document Frequency): A feature extraction method that transforms text data into numerical features. It captures the importance of words in a document relative to the entire corpus, making it a popular choice for natural language processing (NLP) tasks.

Technologies:

Scikit-Learn: A popular Python library for machine learning that provides tools for data preprocessing, model building (like logistic regression), and feature extraction (like TF-IDF).

5.3.2 Natural Language Processing (NLP)

- NLP techniques process and transform text data into formats understandable by ML and DL models.

  Preprocessing Steps: Includes removing URLs, punctuation, and extra whitespace, as well as converting text to lowercase. These steps are necessary to clean and standardize the text data.

  Tokenization and Padding: For DL models, tokenizing (splitting text into tokens/words) and padding (ensuring consistent input size) allow the model to handle input sequences of varying lengths.

  Technologies:

  Regular Expressions (re): A Python module used for text preprocessing and cleaning tasks.

  TensorFlow / Keras: Used for tokenizing text and padding sequences to prepare input for deep learning models.

5.3.3 Deep Learning (DL)

LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) designed to handle sequential data, especially useful for understanding context in text data. LSTM networks are widely used in NLP because they can capture dependencies across words in a sentence.

Bidirectional LSTM: Enhances LSTM performance by processing data from both directions, improving the model's ability to understand context.

Embedding Layer: Maps words to dense vectors of fixed size, capturing semantic relationships between words. Word embeddings help the model learn the meaning of words in context.

Technologies:

TensorFlow / Keras: A deep learning framework that includes predefined layers (like Embedding, LSTM, Bidirectional LSTM) and optimizers. TensorFlow makes it easier to build, train, and evaluate neural network models for NLP tasks.

5.3.4 Optimization and Evaluation

- RMSProp Optimizer: An optimization algorithm in TensorFlow/Keras that adjusts the learning rate based on recent gradients. It's particularly suited for RNN models as it helps stabilize training.
- Evaluation Metrics: Accuracy, precision, recall, and F1-score are computed to assess model performance.

Technologies:

- TensorFlow / Keras: Used to build and compile models, select optimizers like RMSProp, and train and evaluate the models.
- Scikit-Learn: Provides tools for calculating evaluation metrics such as accuracy, precision, recall, and F1-score.

5.3.5 Environment Setup

- Jupyter Notebook, Google Colab, or Local IDEs (like PyCharm or VS Code) to write and execute code.
- GPU Support (optional but recommended for DL models): Speeds up training for larger datasets or deep learning models.

CHAPTER 6

CONCLUSION

This study addresses and proposes a methodology for automatically identifying cyberbullying text on multilingual data. Resolving this problem is essential for managing multilingual social media content and shielding users from the damaging effects of abusive language and verbal abuse. We look at how well our different neural network models function. The most accurate network is CNN-BiLSTM. With its LSTM layer, the CNN BiLSTM can learn global features and long-term dependencies, whereas the CNN alone can only train local characteristics from word n-grams. Future studies will examine both image and video components to determine whether cyberbullying can be automatically identified.

REFERENCES

1. Al-Ajlan, M.A., Ykhlef, M.: Optimized twitter cyberbullying detection based on deep learning. In: 2018 21st Saudi Computer Society National Computer Conference (NCC), pp. 1–5 (2018). IEEE
2. Mahlangu, T., Tu, C.: Deep learning cyberbullying detection using stacked embbedings approach. In: 2019 6th International Confer ence on Soft Computing & Machine Intelligence (ISCMI), pp. 45–49 (2019). IEEE
3. Alam, K.S., Bhowmik, S., Prosun, P.R.K.: Cyberbullying detec tion: an ensemble based machine learning approach. In: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp. 710–715 (2021). IEEE
4. Dewani A, Memon MA, Bhatti S. Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for roman urdu data. J Big Data. 2021;8(1):1–20.
5. Luo, Y., Zhang, X., Hua, J., Shen, W.: Multi-featured cyberbully ing detection based on deep learning. In: 2021 16th International Conference on Computer Science & Education (ICCSE), pp. 746–751 (2021). IEEE
6. Yadav, J., Kumar, D., Chauhan, D.: Cyberbullying detection using pre-trained bert model. In: 2020 International Conference SN Computer Science on Electronics and Sustainable Communication Systems (ICESC), pp. 1096–1100 (2020). IEEE
7. Ahmed, M.T., Rahman, M., Nur, S., Islam, A., Das, D.: Deploy ment of machine learning and deep learning algorithms in detect ing cyberbullying in bangla and romanized bangla text: A com parative study. In: 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Tech nologies (ICAECT), pp. 1–10 (2021). IEEE
8. Mahat, M.: Detecting cyberbullying across multiple social media platforms using deep learning. In: 2021 International Conference on Advance Computing and Innovative Technologies in Engineer ing (ICACITE), pp. 299–301 (2021). IEEE
9. Jain, N., Hegde, A., Jain, A., Joshi, A., Madake, J.: Pseudo-con ventional approach for cyberbullying and hate-speech detection. In: 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), pp. 1–8 (2021). IEEE
10. Iwendi, C., Srivastava, G., Khan, S., Maddikunta, P.K.R.: Cyber bullying detection solutions based on deep learning architectures. Multimedia Systems, 1–14 (2020)
11. Aind, A.T., Ramnaney, A., Sethia, D.: Q-bully: a reinforcement learning based cyberbullying detection framework. In: 2020 Inter national Conference for Emerging Technology (INCET), pp. 1–6 (2020). IEEE
12. Ketsbaia, L., Issac, B., Chen, X.: Detection of hate tweets using machine learning and deep learning. In: 2020 IEEE 19th Interna tional Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 751–758 (2020). IEEE
13. Pradhan, A., Yatam, V.M., Bera, P.: Self-attention for cyberbul lying detection. In: 2020 International Conference on Cyber Situ ational Awareness, Data Analytics and Assessment (CyberSA), pp. 1–6 (2020). IEEE
14. Sahana, B., Sandhya, G., Tanuja, R., Ellur, S., Ajina, A.: Towards a safer conversation space: Detection of toxic content in social media (student consortium). In: 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), pp. 297–301 (2020). IEEE

15. Berrimi, M., Moussaoui, A., Oussalah, M., Saidi, M.: Attention based networks for analyzing inappropriate speech in arabic text. In: 2020 4th International Symposium on Informatics and Its Applications (ISIA), pp. 1–6 (2020). IEEE
16. Dang, C.N., Moreno-García, M.N., De la Prieta, F.: Hybrid deep learning models for sentiment analysis. Complexity 2021 (2021)
17. Yuvaraj N, Chang V, Gobinathan B, Pinagapani A, Kannan S, Dhiman G, Rajan AR. Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. Comput Electr Eng. 2021;92: 107186.
18. Roy PK, Tripathy AK, Das TK, Gao X-Z. A framework for hate speech detection using deep convolutional neural network. IEEE Access. 2020;8:204951–62.
19. Al-Makhadmeh Z, Tolba A. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing. 2020;102(2):501–22.
20. Yadav, Y., Bajaj, P., Gupta, R.K., Sinha, R.: A comparative study of deep learning methods for hate speech and offensive language detection in textual data. In: 2021 IEEE 18th India Council Inter national Conference (INDICON), pp. 1–6 (2021). IEEE.
21. Lee E, Rustam F, Washington PB, El Barakaz F, Aljedaani W, Ashraf I. Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble gcr nn model. IEEE Access. 2022;10:9717–28.
22. d'Sa, A.G., Illina, I., Fohr, D.: Bert and fasttext embeddings for automatic detection of toxic speech. In: 2020 International Multi-Conference on:"Organization of Knowledge and Advanced Technologies"(OCTA), pp. 1–5 (2020). IEEE
23. Jiang, L., Suzuki, Y.: Detecting hate speech from tweets for senti ment analysis. In: 2019 6th International Conference on Systems and Informatics (ICSAI), pp. 671–676 (2019). IEEE
24. Mohaouchane, H., Mourhir, A., Nikolov, N.S.: Detecting offensive language on arabic social media using deep learning. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 466–471 (2019). IEEE 25.
25. Dubey, K., Nair, R., Khan, M.U., Shaikh, S.: Toxic comment detection using lstm. In: 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), pp. 1–8 (2020). IEEE