CYBERBULLYING DETECTION

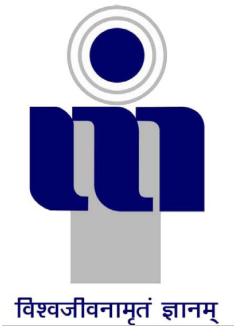
Bachelor of Technology

in

Computer Science and Engineering

by

Maram Vignesh(2021BCS-039)



ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015

JULY 2024

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the report, entitled **CYBERBULLYING DETECTION** , in partial fulfillment of the requirement for Summer Colloquiom 2024 for **Bachelor of Technology in Computer Science and Engineering** and submitted to the institution is an authentic record of my own work carried out during the period *May 2024* to *July 2024* under the supervision of **Prof.Shashikala Tapaswi**. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date:

Name: Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Signature of the Supervisor

ABSTRACT

The arrival of the internet and the proliferation of social media platforms have converted the way people communicate and interact. While these advancements have brought multitudinous benefits, they've also introduced new challenges, particularly in the form of cyberbullying. Cyberbullying refers to the use of digital platforms to threaten, harass,or demean individualities,frequently leading to severe emotional and cerebral consequences. This malicious behavior can be done through colorful online channels, including social media, messaging apps, forums, and gaming communities.

Cyberbullying Detection delves into the development and implementation of a comprehensive system for detecting cyberbullying using Natural Language Processing (NLP) and machine learning techniques. The project addresses the challenge of classifying various forms of cyberbullying by working with a multi-class dataset that includes six distinct labels: 'other cyberbullying', 'not cyberbullying', 'ethnicity', 'religion', 'gender', and 'age'. This balanced dataset ensures a comprehensive analysis across various forms of cyberbullying, providing a more nuanced understanding of online harassment.

A meticulous approach was taken to collect and preprocess the data, utilizing advanced NLP techniques to clean and normalize text. Pre-trained GloVe embeddings were employed to extract meaningful features, which were then used to train various deep learning models, including GRU, Bi-GRU, LSTM, and Bi-LSTM. Additionally, the project explored multimodality by incorporating audio data, converting it to text for further analysis.

A web application was developed to facilitate real-time detection, providing users with an interface to input text and audio for cyberbullying assessment. The results demonstrate the efficacy of the proposed system, particularly the Bi-LSTM model, in accurately identifying different forms of cyberbullying. This project not only advances the technical capabilities in cyberbullying detection but also underscores the importance of integrating multiple data modalities for more robust and reliable outcomes.

ACKNOWLEDGEMENTS

I am grateful to Prof. Shashikala Tapaswi for allowing me to function independently and explore with ideas. I would like to take this opportunity to express my heartfelt gratitude to her not only for her academic guidance but also for her personal interest in the project and constant support as well as confidence-boosting and motivating sessions that proved extremely beneficial and were instrumental in instilling self-assurance and trust. The current work has been nurtured and blossomed mostly as a result of her valuable direction, astute judgment, recommendations, constructive criticism and an eye for perfection. Only because of her tremendous enthusiasm and helpful attitude has the current effort progressed to this point. Finally, I am grateful to the Institution and classmates whose constant encouragement served to renew my spirit, refocus my attention and energy and helped me in carrying out this work.

Maram Vignesh

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

ML	Machine Learning
DL	Deep Learning
LR	Logistic Regression
DT	Decision Tree
RF	Random Forest
NLP	Natural Language Processing
GRU	Gated Recurrent Unit
Bi-GRU	Bidirectional Gated Recurrent Unit
LSTM	Long Short Term Memory
Bi-LSTM	Bidirectional Long Short Term Memory
API	Application Programming Interface

CHAPTER 1

INTRODUCTION

Cyberbullying is when someone utilizes technology to make fun of, hurt, or upset someone else. Willful and repeated harm inflicted through the use of electronic devices such as computers and cell phones. Bullies target their victims in any online forum, social network, computer or video game, message board, or text message on a mobile device.It may involve name calling, threats, sharing private or embarrassing photos, or excluding others.

In addition to harming education, cyberbullying can also influence a student's attendance and academic performance. This is particularly true when bullying happens both online and in person, or when a student has to confront their cyberbully in person. Adolescents who are experiencing the stress of cyberbullying may turn to unhealthy coping strategies, like substance abuse. Teens may have severe instances of self-harm or suicide thoughts.

1.1 Context

The spread of social media and online platforms in the current digital era has drastically changed the dynamics of connection and communication. But while connectivity has many advantages, there is also a negative aspect, which is the widespread problem of cyberbullying, which has serious psychological and societal repercussions. The intentional and persistent use of digital media to harass, threaten, or cause harm to specific people or groups is known as cyberbullying. Cyberbullying, in contrast to traditional bullying, can happen anonymously and crosses physical barriers, which increases its impact on victims.

Effective detection and mitigation measures are desperately needed, as the prevalence and severity of cyberbullying are on the rise. With the abundance of internet content, manual monitoring techniques that are based on tradition are no longer sufficient. This calls for the use of sophisticated computational methods, especially those that make use of machine learning and natural language processing, in order to automate the detection process. Through the utilisation of data-driven methodologies, it is possible to improve the effectiveness of cyberbullying detection while also providing stakeholders with prompt solutions.

1.2 Objectives

The primary objective of this project is to develop a robust cyberbullying detection model using advanced machine learning techniques. The model aims to:

- Accurately Identify Abusive Language:Utilize natural language processing algorithms to analyze text data and detect instances of cyberbullying with high accuracy.
- Understand Context: Implement contextual analysis to differentiate between benign and harmful content, reducing false positives.
- Model Training: Explore and implement a range of deep learning models to effectively capture the nuanced patterns and temporal dependencies inherent in cyberbullying content. Evaluate these models rigorously to identify the most effective approach for classification.

CHAPTER 2

LITERATURE REVIEW

2.1 Background of the Project

There has been a lot of study on the use of automated systems for cyberbullying detection,most of it has focused on binary classification to determine whether bullying is occurring or not. The early strategies relied on rule-based approaches and keyword identification, which frequently proved ineffective because of the complex and dynamic language used in cyberbullying. The emergence of machine learning and natural language processing has greatly improved the capacity to identify this kind of behavior by allowing the examination of contextual and semantic data included in text.

2.2 System Overview

The system developed in this project consists of several key components: data collection and preprocessing, feature extraction, model training, and deployment. Data preprocessing involves cleaning and normalizing the text to remove noise and irrelevant information. Feature extraction is performed using pre-trained GloVe embeddings to capture the semantic meaning of words. Various deep learning models, including GRU, Bi-GRU, LSTM, and Bi-LSTM, are then trained on the processed data. The final component involves the deployment of the model within a web application, which includes functionalities for both text and audio input.

2.3 Related Works

Traditional machine learning methods have been studied in the past for the purpose of detecting cyberbullying. Most of the previous works have been on binary classification. In [1], the authors proposed a cyberbullying detection of comments posted in English. TF-IDF was used as a feature representation technique with ML algorithms, namely SVM and Naïve Bayes. Experimental results showed that SVM had a higher accuracy of 71.25% and the Naïve Bayes (NB) had a 52.70% accuracy.

Using a variety of n-gram languages, Hani et al. compared SVM and Neural Network (NN) classifiers on their dataset in [2]. The findings of the investigation indicated that the NN classifier performed better, with an accuracy of 79.8% compared to SVM's 75.3%.It was demonstrated by Islam et al. in [3] that TF-IDF outperforms Bag-of-Words (BoW) in terms of results. SVM outperformed the other algorithms in terms of cyberbullying detection on two datasets of communications from social networks.

Muneer et al. suggested a comparable method in [4], whereby they assessed the efficacy of multiple classifiers using TF-IDF. With an accuracy of 78.57%, the Logistic Regression classifier proved to be the most effective, according to the data. NLP and a number of classifiers were employed by Rahman et al. in [5] to identify instances of bullying. Logistic Regression outperformed SVM, Random Forest, Naive Bayes, and XGBoost in their analysis.

Model accuracy was 72.7% when Raza et al. applied supervised machine learning with the linear regression approach in [6]. They attained an accuracy of 74.4% while utilizing a voting classifier as well.

In a different study, the scientists used machine learning techniques to detect cyberbullying in Arabic comments. There were two groups created out of the dataset: offensive and nonoffensive. Training was done on three ensemble ML models (Bagging, AdaBoost, and RF) and three non-ensemble ML models (Decision Tree (DT), LR, and SVM). Based on experimental data, Bagging outperformed other ensemble machine learning algorithms, scoring 88% of the F1 Score.

Supervised machine learning techniques were used for hate speech recognition. Three categories were used to create and annotate a dataset of 5000 Roman Urdu tweets: simple-complex, offensive-hate speech, and neutral-hostile. LR did a good job of accurately differentiating between speech that was hurtful or hateful.

CHAPTER 3

METHODOLOGY

The methodology section outlines the systematic approach taken to develop the cyberbullying detection system. It encompasses data collection, preprocessing, model development, training, and evaluation.

3.1 Data Collection

Multiple datasets related to cyberbullying were collected, each containing a variety of text samples labeled with different types of bullying or non-bullying content. The chosen dataset has two Columns named Text and Cyberbullying type and is balanced to ensure an equal representation of each of the six labels: 'other cyberbullying', 'not cyberbullying', 'ethnicity', 'religion', 'gender', and 'age'. This balanced dataset was crucial in training models that could accurately distinguish between the different classes.This dataset contains more than 47000 tweets labelled according to the class of cyberbullying.

3.2 Data Preprocessing

3.2.1 Text Cleaning

- Removing Special Characters and Punctuation: Text data often contains special characters, punctuation marks, and numbers that do not contribute to the analysis. Removing these elements ensures that the text is clean and free from noise. This step involves using regular expressions to filter out unwanted characters.

- Lowercasing: Converting all characters to lowercase ensures uniformity and prevents the model from treating the same words with different cases as separate entities. For example, "Cyberbullying" and "cyberbullying" would be considered the same word.

3.2.2 Tokenization

- Definition: Tokenization is the process of breaking down text into individual words or tokens. This is a fundamental step in NLP as it prepares the text for further processing.
- Implementation: Libraries like NLTK (Natural Language Toolkit) and SpaCy provide efficient tokenizers that can handle complex tokenization tasks, including dealing with punctuation, special characters, and contractions. For example, "I'm" would be tokenized into ["I", "am"].

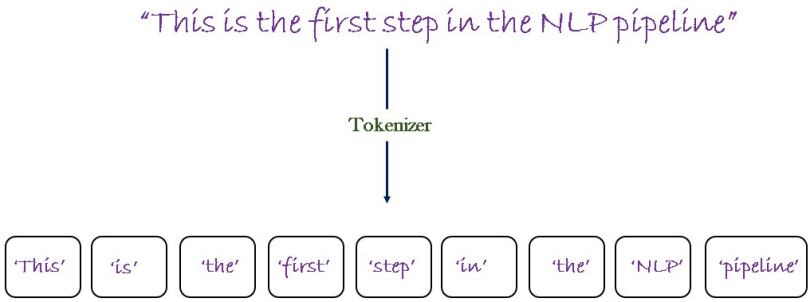


Figure 3.1: Tokenization

3.2.3 Stopword Removal

- Definition: Stopwords are common words that do not contribute significant meaning to the text, such as "and", "the", "is", etc. Removing stopwords reduces the dimensionality of the data and focuses on the meaningful words.
- Implementation: Using predefined lists from libraries like NLTK or SpaCy, stopwords are removed from the tokenized text. This step helps in reducing noise and improving the performance of machine learning models.

3.2.4 Lemmatization and Stemming

- Lemmatization: This process reduces words to their base or root form while ensuring that the transformed word is a valid word in the language. For example, "running" is lemmatized to "run", and "better" is lemmatized to "good". Lemmatization considers the context and parts of speech to produce meaningful root forms.

- Stemming: This technique also reduces words to their root form but may not always result in a valid word. For example, "running" might be stemmed to "run", but "runner" might be stemmed to "runn". Stemming is faster but less accurate compared to lemmatization.
- Implementation: Libraries like NLTK provide tools like WordNetLemmatizer and PorterStemmer to perform these operations. Choosing between lemmatization and stemming depends on the specific requirements of the analysis.

3.3 Feature Extraction

In NLP, feature extraction is the process of converting unprocessed textual data into numerical representations that are comprehensible and processable by machine learning models. Because text data must be transformed into a structured numerical format while maintaining the semantic meaning of the words and phrases, this step is crucial because models work on numerical data.

3.3.1 Embedding Techniques

Embedding techniques are advanced feature extraction methods that convert words or phrases into dense vectors of real numbers. These vectors capture the semantic meanings and relationships between words. Unlike traditional methods such as Bag-of-Words or TF-IDF, which produce sparse and high-dimensional representations, embeddings produce dense and low-dimensional vectors, making them more efficient and effective for machine learning tasks.

3.3.1.1 GloVe Embeddings

Pre-trained word vectors called GloVe (Global Vectors for Word Representation) embeddings use a huge corpus's co-occurrence statistics to determine the semantic links between words. In a continuous vector space, every word is represented as a high-dimensional vector, and words with comparable meanings have similar vectors.

- Loading Pre-Trained GloVe Embeddings: We load pre-trained GloVe embeddings and create an embedding matrix where each word in our vocabulary is represented by a GloVe vector. This matrix is then used in the embedding layer of your neural network.
- Using the Embedding Matrix: During training, each word in our input texts is converted to its corresponding GloVe vector using this embedding matrix. These vectors are fed into the neural network, which learns to classify texts based on these dense representations.

3.4 Multimodal Approach

The predictive model's accuracy and resilience are improved by utilizing both textual and audio data in the multimodal approach to cyberbullying detection. This method can capture the whole range of cyberbullying incidents by merging several data modalities, which may not be evident from a single form of data on its own.

- Audio data related to cyberbullying instances was collected. This data includes voice recordings of cyberbullying instances, which are then converted into text using speech recognition tools such as the Google Speech-to-Text API.
- The Google Speech-to-Text API is utilized to convert audio files into textual format. This involves transcribing spoken words into text, which can then be processed using the same techniques applied to the primary text data.

3.5 Model Exploration

In Cyberbullying Detection, deep learning techniques are favored over simple linear models due to their superior ability to capture complex patterns and relationships within data. Cyberbullying often involves nuanced language, contextual subtleties, and varied linguistic expressions that linear models struggle to interpret effectively. Deep learning models, such as GRU, LSTM, and their bidirectional variants, excel in processing and understanding sequential data, enabling them to recognize intricate patterns in both text and audio inputs. These models leverage layers of neurons to learn hierarchical representations, allowing for the detection of sophisticated features and context-dependent cues critical for accurate cyberbullying identification.

3.5.1 GRU(Gated Recurrent Unit)

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that is designed to handle sequential data and capture dependencies over time. GRUs address the vanishing gradient problem commonly encountered in standard RNNs, making them more effective for long sequences.

1. Architecture

- Update Gate: Controls how much of the past information needs to be passed along to the future. It helps decide which part of the past information to discard and which to keep.

- Reset Gate: Determines how much of the past information to forget. This gate helps the model decide the amount of past information to consider when computing the current hidden state.

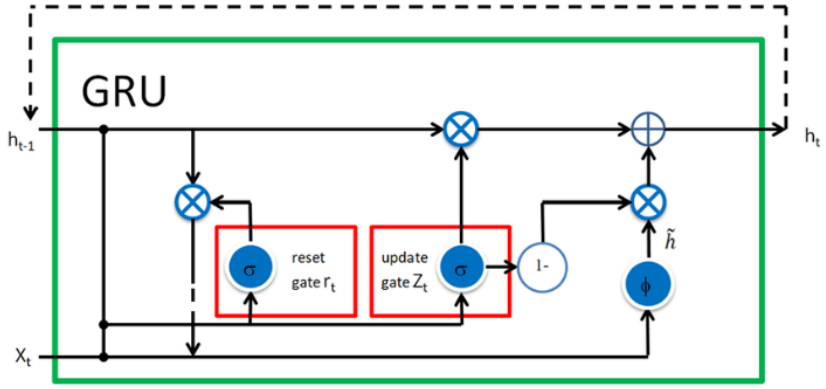


Figure 3.2: GRU Architecture

2. Functioning

- At each time step, the update gate z_t and the reset gate r_t are computed using the current input x_t and the previous hidden state h_{t-1}.
- The candidate hidden state h'_t is then computed using the reset gate.
- The final hidden state h_t is a linear interpolation between the previous hidden state and the candidate hidden state, controlled by the update gate.

3. Advantages

- Simplified Structure: GRUs have fewer parameters than LSTMs, making them faster to train and computationally less expensive.
- Performance: GRUs often perform similarly or even better than LSTMs on certain tasks, despite their simpler structure.

4. Applications

- GRUs are used in tasks such as speech recognition, language modeling, and time series prediction.

3.5.2 Bi-GRU(Bidirectional Gated Recurrent Unit)

Bidirectional GRUs (Bi-GRUs) are an extension of the standard GRU that processes the input sequence in both forward and backward directions, capturing dependencies from both past and future contexts.

1. Architecture

- Forward GRU: Processes the sequence from the beginning to the end.
- Backward GRU: Processes the sequence from the end to the beginning.
- The outputs of both GRUs are concatenated at each time step to form the final output.

2. Functioning

- At each time step, the forward and backward GRUs compute their hidden states independently.
- The final output is obtained by concatenating the hidden states from both directions.

3. Advantages

- Contextual Understanding: Bi-GRUs provide a richer representation of the input by considering both past and future contexts, which is particularly useful in tasks where context is crucial.
- Improved Performance: Bi-GRUs often outperform unidirectional GRUs on tasks like text classification and named entity recognition.

4. Applications

- Bi-GRUs are commonly used in natural language processing tasks, such as sentiment analysis and machine translation.

3.5.3 LSTM(Long Short Term Memory)

Long Short-Term Memory (LSTM) networks are a type of RNN architecture that addresses the vanishing gradient problem and captures long-term dependencies more effectively.

1. Architecture

- Cell State: The cell state is the memory of the network, capable of carrying information across long time steps.
- Forget Gate: Determines which information from the cell state should be discarded.
- Input Gate: Decides which new information should be added to the cell state.
- Output Gate: Controls the output based on the cell state and the current input.

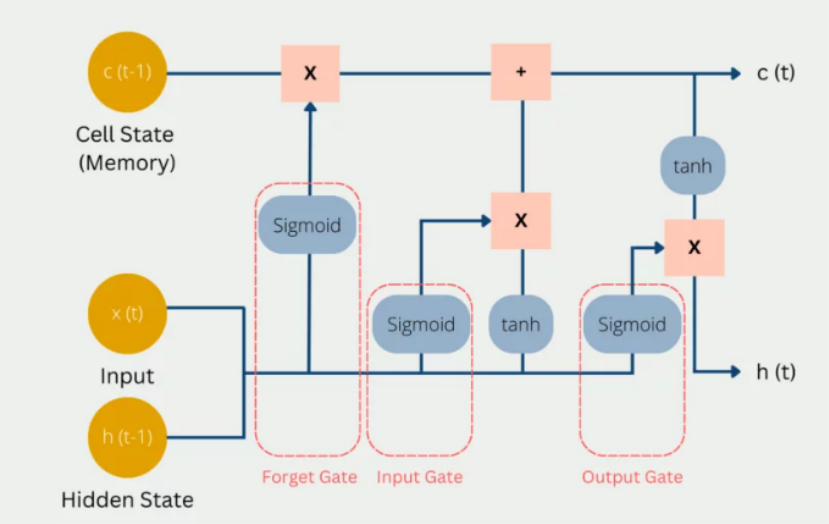


Figure 3.3: LSTM Architecture

2. Functioning

- At each time step, the forget gate f_t, input gate i_t, and output gate o_t are computed using the current input x_t and the previous hidden state h_{t-1}.
- The candidate cell state C'_t is computed, and the cell state C_t is updated by combining the old cell state and the candidate cell state, controlled by the forget and input gates.
- The final hidden state h_t is computed using the output gate and the updated cell state.

3. Advantages

- Handling Long Sequences: LSTMs can capture long-term dependencies effectively, making them suitable for tasks involving long sequences.
- Versatility: LSTMs are highly versatile and can be applied to a wide range of sequential data tasks.

4. Applications

- LSTMs are widely used in applications such as machine translation, speech recognition, and video analysis.

3.5.4 Bi-LSTM(Bidirectional Long Short Term Memory)

Bidirectional LSTMs (Bi-LSTMs) are an extension of LSTMs that process the input sequence in both forward and backward directions, capturing dependencies from both past and future contexts.

1. Architecture

- Forward LSTM: Processes the sequence from the beginning to the end.
- Backward LSTM: Processes the sequence from the end to the beginning.
- The outputs of both LSTMs are concatenated at each time step to form the final output.

2. Functioning

- At each time step, the forward and backward LSTMs compute their hidden states independently.
- The final output is obtained by concatenating the hidden states from both directions.

3. Advantages

- Contextual Understanding: Bi-LSTMs provide a richer representation of the input by considering both past and future contexts.
- Enhanced Performance: Bi-LSTMs often outperform unidirectional LSTMs on tasks that require understanding context from both directions.

4. Applications

- Bi-LSTMs are commonly used in tasks such as text classification, named entity recognition, and question answering.

CHAPTER 4

RESULTS

4.1 Experimental Analysis

In our cyberbullying detection project, we explored various deep learning models, namely GRU, Bi-GRU, LSTM, and Bi-LSTM, to evaluate their performance in terms of accuracy, precision, and recall. The results indicate significant differences in the efficacy of these models, highlighting the superiority of advanced sequential models over simpler ones.

The GRU model achieved an accuracy of 57.91%, a precision of 82.74%, and a recall of 36.89%. While its precision was relatively high, indicating its ability to correctly identify positive instances of cyberbullying, its lower recall suggests it missed a considerable number of true positive cases.

The Bi-GRU model, an extension of GRU that processes data in both forward and backward directions, showed an improvement over the basic GRU. With an accuracy of 60.66%, a precision of 84.15%, and a recall of 40.57%, Bi-GRU demonstrated a better balance between precision and recall, albeit with modest overall performance. This improvement can be attributed to the model's ability to capture more contextual information from the input sequences, which is crucial for understanding the intricacies of cyberbullying content.

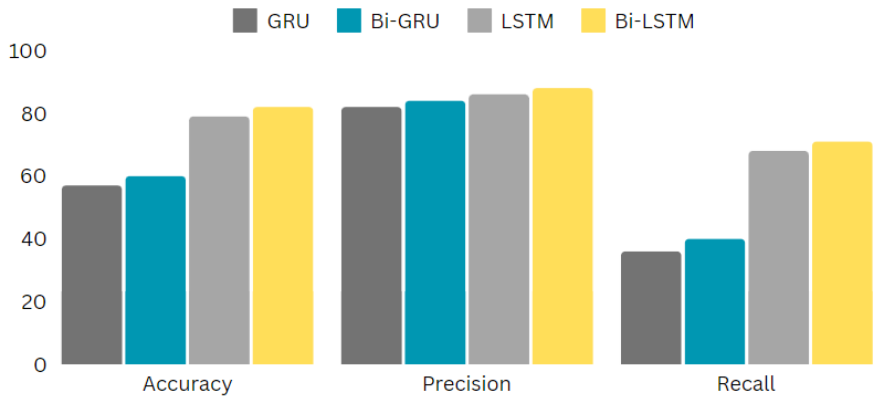


Figure 4.1: Summary of Findings

The LSTM and Bi-LSTM models, designed to address the limitations of GRUs in handling long-term dependencies, outperformed the GRU-based models significantly. The LSTM model achieved an accuracy of 79.46%, a precision of 86.66%, and a recall of 68.86%, showcasing its strength in capturing long-term dependencies and nuanced patterns in the data.

The Bi-LSTM model further enhanced performance, achieving the highest metrics with an accuracy of 81.25%, a precision of 88.02%, and a recall of 71.83%. The bidirectional nature of Bi-LSTM allows it to consider both past and future contexts, making it highly effective in understanding the complex and context-dependent nature of cyberbullying. These results underline the importance of using advanced deep learning architectures to improve the detection of cyberbullying, ensuring a more robust and reliable identification process.

Model	Accuracy	Precision	Recall	F1 Score
GRU	57.91%	82.74%	36.89%	51.02%
Bi-GRU	60.66%	84.15%	40.57%	54.74%
LSTM	79.46%	86.66%	68.86%	76.74%
Bi-LSTM	81.25%	88.02%	71.83%	79.10%

Figure 4.2: Comparision Table

4.2 Website Implementation

4.2.1 Overview of the Website

The website was developed to provide a user-friendly interface for the cyberbullying detection system. It allows users to input text and audio data, which are then processed by the trained models to detect instances of cyberbullying. The website serves as an accessible platform for deploying the detection models in a real-world scenario.

4.2.2 Technologies Used

The development of the website utilized various technologies and frameworks to ensure a robust and scalable solution. The backend was built using Flask, a lightweight Python web framework, which handles model inference and data processing. The frontend was developed using React, a popular JavaScript library for building interactive user interfaces.

4.2.3 Website Features

The website includes several key features:
Home Page: The "Home" page serves as the landing page of the website and provides an overview of the cyberbullying detection project. It is designed to be welcoming and informative, guiding users through the website's main functionalities.
About Page: The "About" Page provides detailed information about the project,functionalities and all about how the project is implemented.
Predict: This takes to the page where a user can choose the type of input for prediction.
Text Input: Users can input text directly into the website, which is then preprocessed and analyzed by the detection model.
Audio Input: Users can upload audio files, which are converted to text using speech-to-text technology and then processed by the model.
Prediction Display: The website displays the prediction results, indicating whether the input data contains cyberbullying content and the specific type if detected.

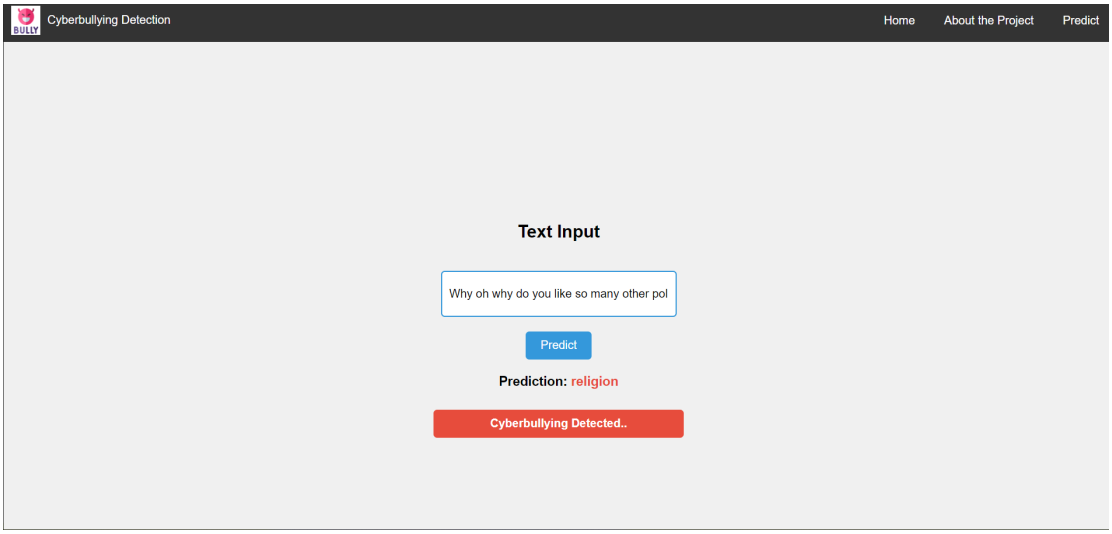


Figure 4.3: Example of Prediction using Text

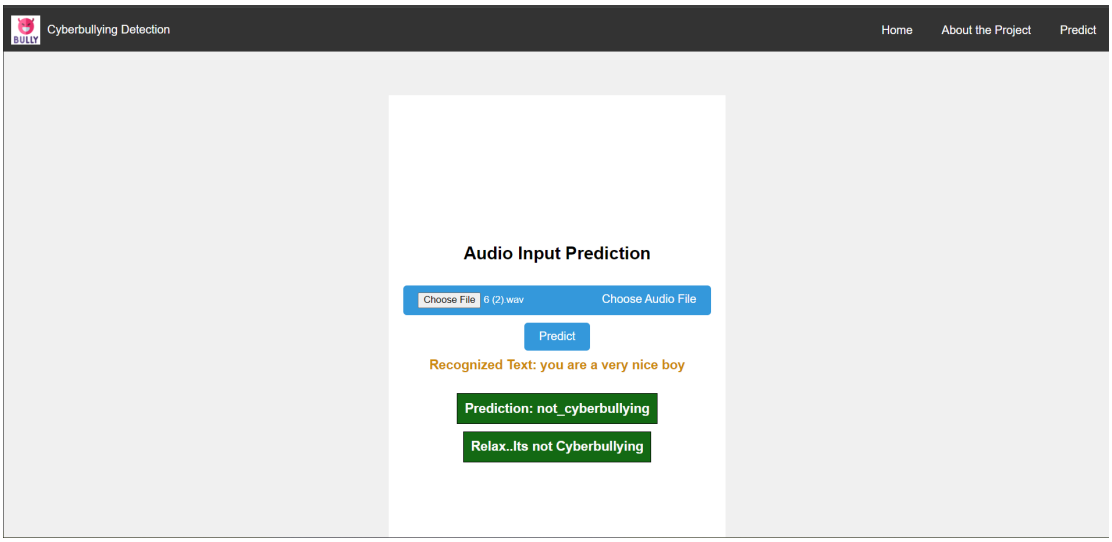


Figure 4.4: Example of Prediction using Audio

CHAPTER 5

CONCLUSION

5.1 Summary

In conclusion, this project represents a holistic approach to tackling cyberbullying through advanced machine learning techniques and innovative multimodal data processing.Leveraging natural language processing (NLP), we applied techniques such as tokenization, stemming, and stop word removal to transform textual data into a format conducive to model training. Concurrently, our integration of Google Speech-to-Text API enabled seamless conversion of audio inputs into text, expanding our detection capabilities to include spoken content.

At the core of our methodology lay the exploration and implementation of cutting-edge deep learning architectures—GRU, Bi-GRU, LSTM, and Bi-LSTM. Through extensive experimentation and meticulous model comparison, we demonstrated the superior performance of Bi-LSTM in terms of accuracy, precision, and recall for cyberbullying detection tasks. These models not only outperformed traditional linear classifiers but also exhibited remarkable adaptability to the complex and context-sensitive nature of abusive language online.

5.2 Future Scope

- Integration of Additional Data Modalities: Incorporate image and video analysis to detect cyberbullying behaviors in multimedia content.
- Real-Time Monitoring and Alerts: Implement systems for real-time monitoring of online interactions and immediate alert systems for potential cyberbullying incidents.

- Deployment in Social Media Platforms: Extend the application to major social media platforms through APIs or browser extensions to provide widespread protection and send real time reports to the users and police.

- Collaboration with Educational Institutions: Partner with schools and universities to integrate the system into educational curricula and promote awareness and prevention of cyberbullying.

5.3 Limitations

- Data Availability and Quality: The effectiveness of the models heavily depends on the availability and quality of labeled data. Limited or biased datasets may affect the model's ability to generalize to diverse forms of cyberbullying across different demographics and platforms.

- Algorithm Complexity and Scalability: Deep learning models like LSTM and Bi-LSTM are computationally intensive and require substantial resources for training and inference. This may limit the scalability of the solution, especially on resource-constrained devices or in real-time applications.

- Real-time Processing: Achieving real-time detection capabilities may be constrained by the latency in data processing and model inference. Balancing accuracy with speed is critical for timely intervention in cyberbullying incidents.

5.4 Novelty

The novelty of our approach lies in several key innovations throughout our cyberbullying detection project. Firstly, we utilized a meticulously curated dataset comprising six balanced labels—'othercyberbullying', 'not cyberbullying', 'ethnicity', 'religion', 'gender', and 'age'—which diverges from traditional binary classifications, allowing for more nuanced predictions across diverse categories of cyberbullying behaviors. Our preprocessing pipeline integrated advanced techniques to clean and tokenize text inputs, leveraging GloVe embeddings to enrich semantic understanding and enhance model performance. We explored and implemented sophisticated deep learning architectures including GRU and LSTM models, supplemented by effective callbacks and early stopping mechanisms to optimize training and prevent overfitting.Moreover, extending beyond model development, we innovatively incorporated these capabilities into a user-friendly web application where user can predict the type of cyberbullying using text or audio as input.

REFERENCES

[1] Dalvi, R. R., Chavan, S. B. and Halbe, A.: 2020, Detecting a twitter cyberbullying using machine learning, *Proceedings of 4th International Conference on Intelligent Computing and Control Systems*.

[2] Hani, J., Mohamed, N., Ahmed, M., Emad, Z., Amer, E. and Ammar, M.: 2019, Social media cyberbullying detection using machine learning, *Int. J. of Advanced Computer Science and Applications* **10**(5).

[3] Islam, M. M., Uddin, M. A., Islam, L., Akter, A., Sharmin, S. and Acharjee, U. K.: 2020, Cyberbullying detection on social networks using machine learning approaches, *IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, IEEE, pp. 1–6.

[4] Muneer, A. and Fati, S. M.: 2020, A comparative analysis of machine learning techniques for cyberbullying detection on twitter, *Future Internet* **12**(11), 187.

[5] Rahman, M. H. U., Divya, M., Reddy, B. R., Kumar, K. S. and Vani, P. R.: 2022, Cyberbullying detection using natural language processing, *Ijiraset* .

[6] Raza, M. O., Memon, M., Bhatti, S. and Bux, R.: 2020, Detecting cyberbullying in social commentary using supervised machine learning, *Proc. of the Future of Information and Communication Conference (FICC)*, Springer, pp. 621–630.

[7] Teng, T. H. and Varathan, K. D.: 2023, Cyberbullying detection in social networks: A comparison between machine learning and transfer learning approaches, *Research Article* .