Comprehensive Cyberbullying Detection Using Multiclass NLP, GloVe Embeddings, and Deep Learning Models

Maram Vignesh[1], Prof.Shashikala Tapaswi[2]
Department of Computer Science and Engineering, ABV-IIITM Gwalior, India
[1]vinnumaram@gmail.com,[2]stapaswi@iiitm.ac.in

***Abstract*—The rise of the internet and social media has transformed communication, but also brought challenges like cyberbullying, where individuals are harassed or demeaned online. This paper presents a comprehensive cyberbullying detection system using Natural Language Processing (NLP) and machine learning. The system classifies various forms of cyberbullying with a balanced dataset containing six labels: 'other cyberbullying', 'not cyberbullying', 'ethnicity', 'religion', 'gender', and 'age'. Data preprocessing utilized advanced NLP techniques and pre-trained GloVe embeddings for feature extraction. We trained deep learning models, including GRU, Bi-GRU, LSTM, and Bi-LSTM, and incorporated multimodal data by converting audio to text. A web application was developed for real-time detection, allowing users to input text and audio for assessment. Results demonstrate the system's effectiveness, particularly the Bi-LSTM model, in accurately identifying cyberbullying. This study advances technical capabilities in cyberbullying detection and underscores the importance of multimodal data integration for robust outcomes.**

***Index Terms*—Cyberbullying detection, Natural Language Processing (NLP), Machine Learning, GloVe embeddings, deep learning models, GRU, Bi-GRU, LSTM, Bi-LSTM, multimodal data,web application.**

I. 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.

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.

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.

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.

II. Literature Survey

Traditional machine learning methods have been studied in the past for the purpose of detecting cyberbullying.Using a variety of n-gram languages, Hani et al. compared SVM and Neural Network (NN) classifiers on their dataset in [1]. 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 [2] 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 [3], 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 [4] 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 [5]. They attained an accuracy of 74.4% while utilizing a voting classifier as well.

Most of the previous works have been on binary classification. In [6], 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.

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.

III. 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.

A. 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.

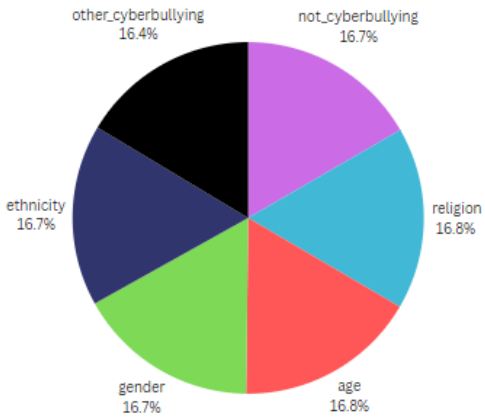

Fig. 1. Distribution of Dataset

B. Data Preprocessing

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.

2) Tokenization:

- 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.

3) Stopword Removal:

- 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.

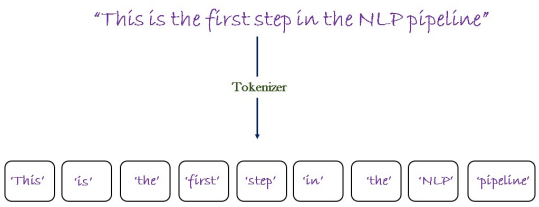

Fig. 2. Tokenization

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.

C. 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.

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.

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.

D. 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.

E. 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.

1) Gated Recurrent Unit: The Gated Recurrent Unit (GRU) is an advanced recurrent neural network (RNN) architecture adept at handling sequential data and capturing temporal dependencies. GRUs mitigate the vanishing gradient problem prevalent in standard RNNs, enhancing their effectiveness for processing long sequences.

Architecturally, GRUs consist of two primary gates: the update gate, which regulates the transmission of past information to future states by determining what to retain and discard, and the reset gate, which decides the extent of past information to forget when computing the current hidden state. At each time step, the update gate and reset gate are computed using the current input and the previous hidden state.

The candidate hidden state is then derived using the reset gate, and the final hidden state is a linear interpolation between the previous hidden state and the candidate hidden state, controlled by the update gate. GRUs offer a simplified structure with fewer parameters than LSTMs, leading to faster training and reduced computational expense, while often delivering comparable or superior performance on specific tasks. GRUs are widely applied in fields such as speech recognition,

language modeling, and time series prediction, owing to their efficiency and robustness in handling sequential data.

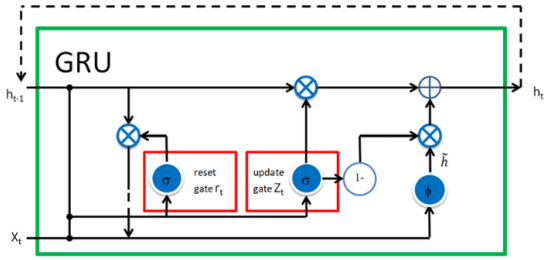

Fig. 3. GRU Architecture

2) Bidirectional Gated Recurrent Unit: Bidirectional GRUs (Bi-GRUs) are an advanced variant of the standard Gated Recurrent Units (GRUs) designed to process input sequences in both forward and backward directions, effectively capturing dependencies from both past and future contexts. In terms of architecture, Bi-GRUs utilize two GRUs: a Forward GRU, which processes the sequence from the beginning to the end, and a Backward GRU, which processes the sequence from the end to the beginning. At each time step, the outputs of these GRUs are concatenated to form the final output, thereby enriching the representation of the input sequence.

The functioning of Bi-GRUs involves independently computing the hidden states for both forward and backward GRUs at each time step. These hidden states are then concatenated to produce the final output, allowing the model to leverage information from both directions. This bidirectional processing provides a more comprehensive contextual understanding, which is particularly beneficial in tasks requiring the consideration of both past and future contexts.

The advantages of using Bi-GRUs include improved contextual understanding and performance. By integrating information from both directions, Bi-GRUs offer a richer representation of the input data, enhancing the model's ability to understand context. This leads to improved performance in various tasks, such as text classification and named entity recognition, where context is crucial. Bi-GRUs are widely applied in natural language processing (NLP) tasks, including sentiment analysis and machine translation, where their ability to capture bidirectional dependencies results in more accurate and robust models.

3) Long Short Term Memory: Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) architecture designed to address the vanishing gradient problem and capture long-term dependencies more effectively. The architecture of LSTMs includes a cell state, which acts as the memory of the network and carries information across long time steps. Three gates regulate the flow of information: the forget gate determines which information from the cell state should be discarded, the input gate decides which new information should be added to the cell state, and the output gate controls the output based on the cell state and the current input.

At each time step, these gates compute their respective values using the current input and the previous hidden state, allowing the network to update the cell state and compute the final hidden state efficiently. LSTMs handle long sequences effectively, making them suitable for tasks such as machine translation, speech recognition, and video analysis, where capturing long-term dependencies is crucial. Their versatility allows them to be applied to a wide range of sequential data tasks, providing robust and accurate performance.

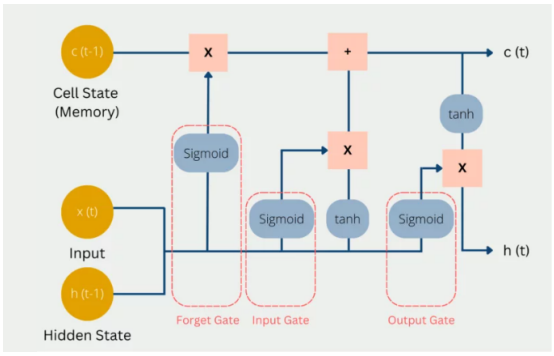

Fig. 4. LSTM Architecture

4) Bidirectional Long Short Term Memory: Bidirectional Long Short-Term Memory (Bi-LSTM) networks are an advanced extension of LSTMs that process input sequences in both forward and backward directions, thereby capturing dependencies from both past and future contexts. In terms of architecture, Bi-LSTMs consist of a forward LSTM that processes the sequence from the beginning to the end and a backward LSTM that 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, allowing the network to utilize information from both directions simultaneously. Functionally, at each time step, the forward and backward LSTMs compute their hidden states independently, and the final output is obtained by concatenating these hidden states.

This bidirectional processing provides a richer representation of the input by considering both past and future contexts, leading to enhanced performance in tasks that require understanding context from both directions. Bi-LSTMs are commonly used in natural language processing tasks such as text classification, named entity recognition, and question answering, where capturing comprehensive context is essential for accurate predictions.

IV. Experiments and Results

We used the F-score measure to evaluate the performance of the presented model. They are defined as follows.

1. True Positive (TP): Instances where the model correctly identifies a text or audio input as belonging to a specific type of cyberbullying.

2. True Negative (TN): Instances where the model correctly identifies a text or audio input as not belonging to any type of cyberbullying.

3. False Positive (FP): Instances where the model incorrectly identifies a text or audio input as belonging to a specific type of cyberbullying when it actually does not.
4. False Negatives (FN): Instances where the model incorrectly identifies a text or audio input as not belonging to any type of cyberbullying when it actually does.
5. Precision: It measures how many of the identified positive instances (i.e., cases of cyberbullying) are actually correct. It is calculated as given in Eq-1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

6. Recall: It measures how many of the actual positive instances (i.e., cases of cyberbullying) the model correctly identifies. It is calculated as given in Eq-2.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

7. Accuracy: It measures the overall correctness of the model's predictions.It is calculated as defined in Eq-3.

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{3}$$

8. F1 Score: It is the harmonic mean of precision and recall, providing a single metric that balances both concerns.It is calculated as defined in Eq-4.

$$F1Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{4}$$

A. Results and 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.

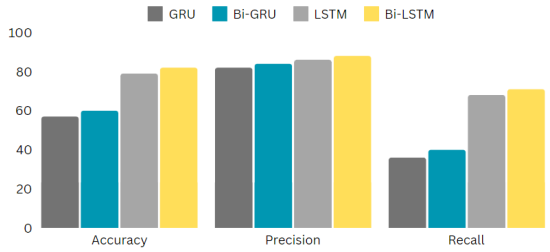

Fig. 5. Summary of Findings

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%

Fig. 6. Comparision Table

B. Website Implementation

The website developed for the cyberbullying detection project offers a user-friendly interface for inputting text and audio data, which are processed by trained models to detect cyberbullying. Built with Flask for the backend and React for the frontend, the website ensures robust and scalable performance. Key features include a "Home" page that provides an overview of the project, an "About" page with detailed implementation information, and a "Predict" page where users can choose between text or audio input. Text input is preprocessed and analyzed directly, while audio files are converted to text using speech-to-text technology before analysis. The results display whether the input contains cyberbullying content and specify the type if detected.

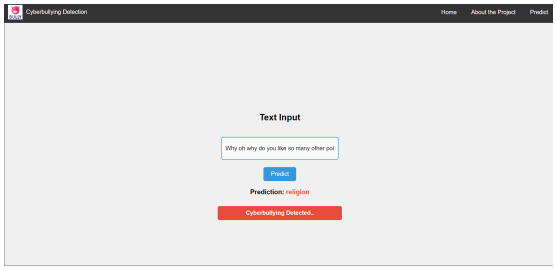

Fig. 7. Example of Prediction using Text

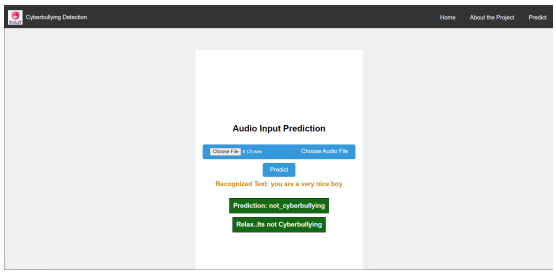

Fig. 8. Example of Prediction using Audio

V. Conclusion

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.

References

[1] J. Hani, N. Mohamed, M. Ahmed, Z. Emad, E. Amer, and M. Ammar, "Social media cyberbullying detection using machine learning," *Int. J. of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019.

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

[3] A. Muneer and S. M. Fati, "A comparative analysis of machine learning techniques for cyberbullying detection on twitter," *Future Internet*, vol. 12, no. 11, p. 187, 2020.

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

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

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

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