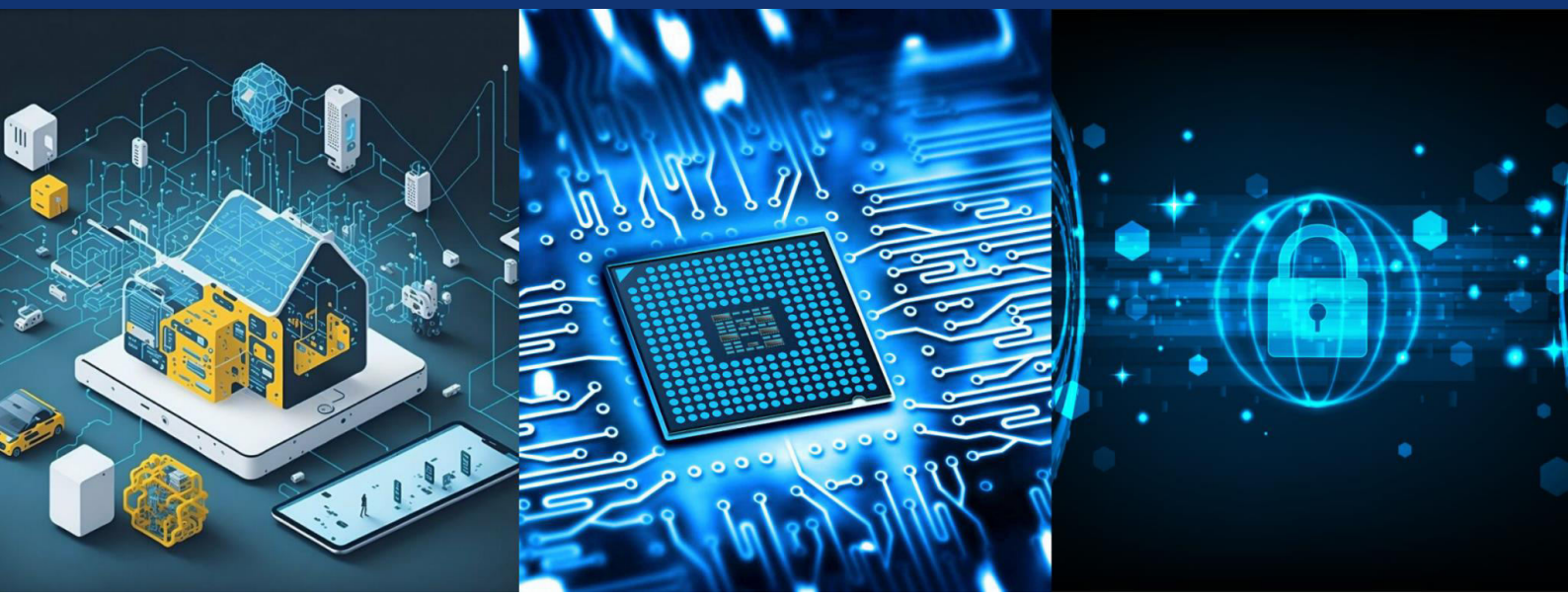
IJIRCCE

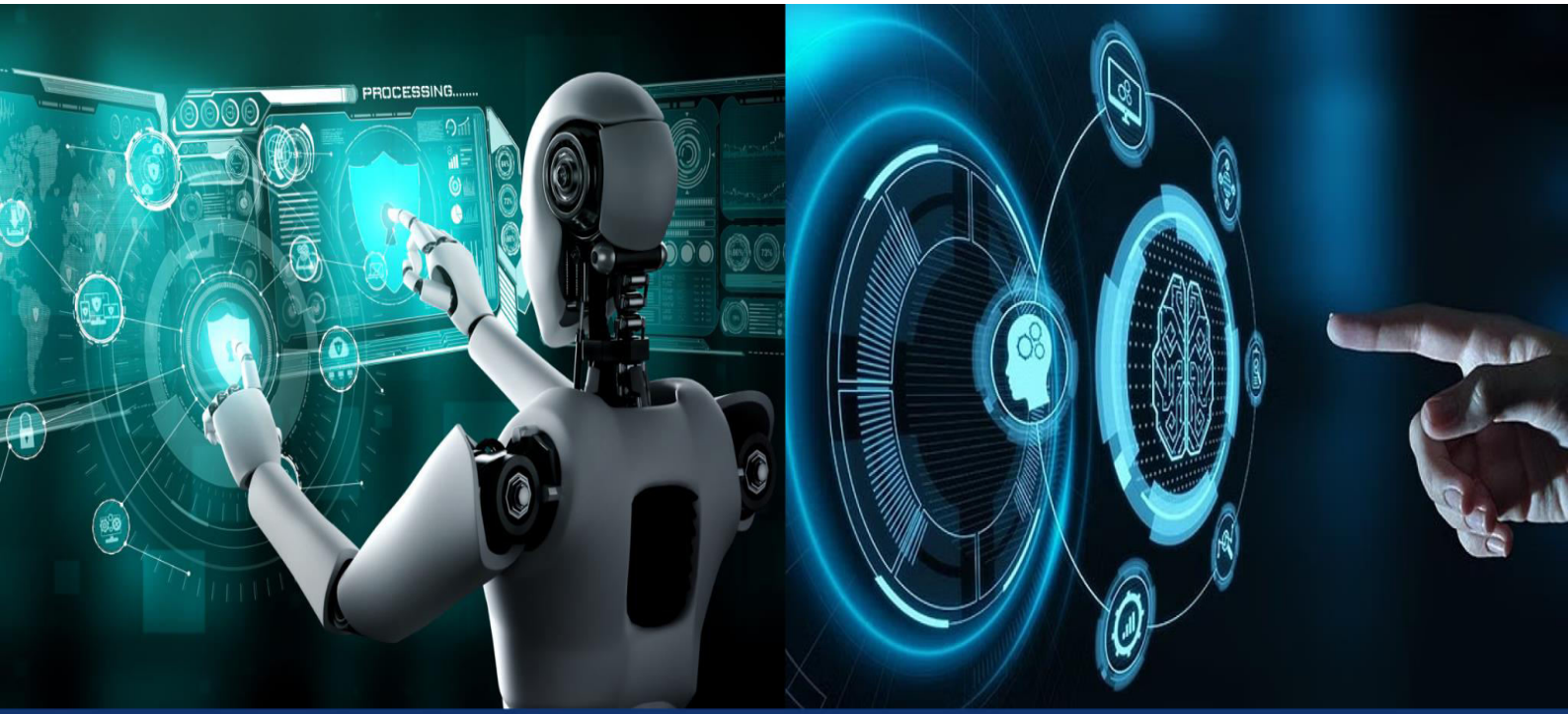
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 4, April 2025

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Leveraging Deep Learning for Cyber Bullying Detection on Social Media Platforms: A Holistic Approach to Mitigate Online Harassment

Thanuja S, Dr. Priya V, Vanthanadevi S, Yuvanidhi S

Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, India

ABSTRACT: Cyberbullying is a growing concern on social media platforms. This research presents an automated system for detecting cyberbullying in social media messages using a combination of supervised machine learning (ML) and natural language processing (NLP) techniques. The system utilizes a pre-trained model to classify messages as bullying or non-bullying based on textual features. It combines traditional ML models, like Logistic Regression, with advanced deep learning methods such as Long Short-Term Memory (LSTM) networks to analyze user-generated content. Preprocessing steps, including tokenization, stop word removal, and TF-IDF vectorization, transform raw text into structured data for model training. The model is trained on a labeled dataset containing both bullying and non-bullying messages to improve classification accuracy. A Streamlit-based web application allows users to input messages and receive real-time feedback on whether the message is classified as cyberbullying. This system aims to assist social media platforms in identifying and preventing cyberbullying, contributing to safer online communities. Experimental results highlight its potential for real-world deployment.

KEYWORDS: Automated Cyberbullying Detection, Digital Harassment Identification, AI Driven Social Media Monitoring, NLP Based Cyberbullying Filter, Real-Time Bullying Classifier

I. INTRODUCTION

In the digital age, platforms like Twitter, Facebook, Instagram, and TikTok have redefined communication, enabling global interaction and real-time sharing. However, they also bring challenges, particularly cyberbullying, which is distinct from traditional bullying due to its anonymity and rapid content spread [1]. Cyberbullying has serious psychological effects, including anxiety, depression, and suicidal tendencies, especially for vulnerable groups like teenagers, young adults, and marginalized communities [2]. Reports show alarming levels of online harassment, emphasizing the need for scalable, automated solutions to replace impractical manual moderation [3]. This study focuses on cyberbullying detection using machine learning (ML) and natural language processing (NLP) techniques. By employing supervised learning models like Logistic Regression and Long Short-Term Memory (LSTM) networks, the system accurately analyzes social media messages to identify harmful content. Extensive text preprocessing, including tokenization, stop word removal, and TF-IDF vectorization, converts raw messages into structured data for analysis. These methods help the system account for the complex and nuanced nature of online communication, achieving high accuracy in detecting abusive behavior. Ethical considerations and cultural sensitivity are integral to this research, ensuring the system is inclusive and respects linguistic diversity. This study aims to create safer, more inclusive digital environments and contribute to global efforts to combat online harassment [4]. It also acknowledges the intersectionality of online abuse with misinformation, hate speech, and privacy violations. The proposed system could be expanded to detect and mitigate these issues, offering a comprehensive framework for promoting respectful online communication as digital platforms evolve.

II. LITERATURE REVIEW

The rise of offensive language on social media platforms has sparked extensive research into detecting and mitigating such behavior using advanced machine learning and deep learning methods. In a study by Kaur et al. (2024), a multi-class classification model for offensive language detection was introduced. The model, based on a dataset of 14,200 labeled tweets, categorized text into three levels: offensive language detection, categorization, and target identification. Preprocessing techniques such as tokenization, TF-IDF for feature selection, and word cloud visualizations improved

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

the data quality [5]. Models like bidirectional gated recurrent units (Bi-GRU), multi-dense long short-term memory (LSTM), and bidirectional LSTM achieved impressive results, with Bi-LSTM reaching 99.9% accuracy for target identification and the lowest loss of 0.01. Park and Fung (2017) applied a hybrid CNN-based approach to classify racist and sexist language using a multi-class framework [6].

Their study, processing 20,000 tweets, found that SVM achieved 83.9% precision, though improvements were needed in robustness. Similarly, Chakraborty and Seddiqui (2019) used NLP techniques to predict abusive language in Bengali texts with models like SVM, Naive Bayes, and CNN combined with LSTM. SVM provided the highest accuracy at 78%, though the authors noted that additional features were needed to enhance performance. Nayel and Shashirekha (2019) focused on detecting hate speech in Indo-European languages, reporting good results despite a small dataset. Pitenis et al. (2020) studied offensive content detection in Greek texts using the Offensive Greek Tweet Dataset of 4,800 tweets [7].

Their findings highlighted the effectiveness of LSTM and GRU, which outperformed other algorithms in terms of accuracy. Likewise, Ibrohim and Budi (2018) developed a hate speech and abusive language detection model for Indonesian tweets, utilizing classifiers like Random Forest and SVM. The model achieved a moderate accuracy of 77.36%, suggesting that more robust classification techniques were needed. Akhter et al. (2021) created an abusive language detection model for Roman Urdu comments using CNN, achieving an accuracy of 96.2%, showcasing the potential of deep learning in multilingual contexts. Haque et al. (2023) carried out multi-class sentiment classification for Bengali social media comments, analyzing 42,036 Facebook posts [8].

Their approach categorized text into four classes (religious, sexual, political, and acceptable) and used classifiers like logistic regression and SVM. The highest accuracy of 93.3% was achieved through stochastic gradient descent and random forest. Sabry et al. (2022) examined abusive content in tweets using Bi-LSTM on datasets like HASOC 2021 and OLID 2019, achieving F1 scores of 83.05% and 82.90%, respectively, though they noted challenges in handling long-range dependencies in text. Several studies emphasize the importance of contextual embeddings and multi-layer architectures for improving performance. For example, Bidirectional LSTM (Bi-LSTM) and multi-dense LSTM models are recognized for their ability to process sequential data effectively. When combined with pre-trained embeddings such as Word2Vec and BERT, these models enable robust classification and identification of abusive language across different languages and platforms. Common preprocessing steps like tokenization, stemming, and punctuation removal are universally acknowledged as essential for enhancing the quality of training data. Despite significant progress in abusive language detection, challenges such as imbalanced datasets, real-time detection, and generalization to unseen data persist [9]. Future research could explore integrating multimodal datasets, including text, images, and audio, to further improve the accuracy and versatility of detection systems in various online environments.

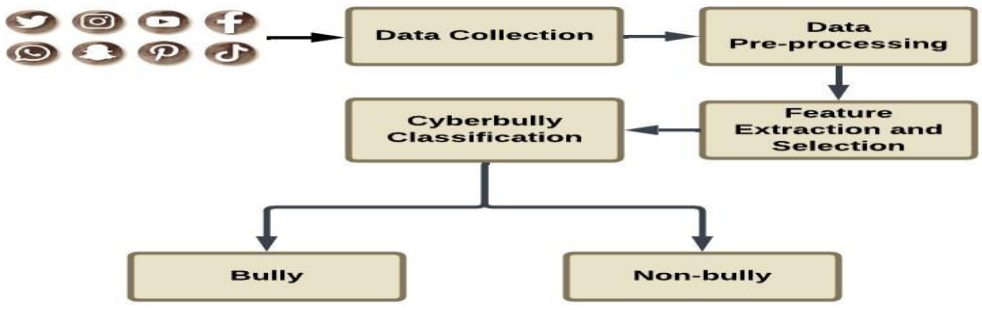


Fig.1. Flow diagram of the cyber bullying detection system architecture.

III. METHODOLOGY

Despite advancements in cyber bullying and offensive language detection, existing models still face several challenges that limit their effectiveness. One key issue is the difficulty in understanding sequential context, as many systems struggle to capture the flow and meaning of conversations, leading to misclassifications. To address this, the proposed

model employs Long Short-Term Memory (LSTM) networks, which excel at retaining and analyzing context across sequences, thereby improving classification accuracy [10]. Another challenge is the presence of irrelevant data, such as stop words and symbols, which can dilute model performance. To mitigate this, the system applies robust preprocessing techniques, including tokenization, stop word removal, and TF-IDF vectorization, ensuring the use of only meaningful data for analysis [11]. Additionally, existing models often struggle with detecting subtle offensive language, particularly in nuanced or context-dependent scenarios. To overcome this, the model combines Logistic Regression with LSTM networks, leveraging the efficiency of machine learning for feature extraction and the contextual understanding provided by deep learning. The architecture is structured to address these challenges throughout the process. It begins with data collection, followed by preprocessing, feature extraction, and classification. By integrating machine learning and deep learning techniques, the model effectively classifies text into "bully" and "non-bully" categories, offering a more accurate and reliable solution for detecting offensive language across online platforms [12].
The system architecture, as illustrated in Fig. 1. is designed to address the challenges in detecting cyber bullying and offensive language systematically. It comprises four main components: Data Collection, Preprocessing, Feature Extraction and Selection, and Classification, each contributing to the accurate and reliable identification of "bully" and "non-bully" content.

A. Data collection and Exploration

The proposed system architecture begins with a structured sequence of steps, starting with data preprocessing. The foundation of the proposed system is built upon a carefully curated dataset consisting of text messages labeled for toxicity. The dataset, comprising a total of 33,754 entries, is structured into two distinct columns: *TEXT* and *TOXICITY*. The *TEXT* column contains the actual text messages, while the *TOXICITY* column categorizes each message as either *toxic* or *non-toxic*, forming the basis for the classification task. Specifically, messages labeled as *toxic* contain offensive, harmful, or inappropriate content, while those labeled as *non-toxic* are considered benign or neutral. This binary classification task aims to develop a system capable of distinguishing between these two categories with high accuracy. Upon a deeper analysis of the dataset, a notable issue emerged: a significant class imbalance. Out of the 33,754 samples, 22,447 messages were labeled as *non-toxic*, while 11,289 messages were marked as *toxic*. This imbalance between the two classes posed a challenge for the machine learning models, as models trained on imbalanced datasets are more likely to favor the majority class. Consequently, the model may develop a bias toward predicting non-toxic messages, leading to suboptimal performance when it comes to identifying toxic content. This type of class imbalance is common in many real-world datasets, and handling it effectively is critical for building a robust classifier [13]. Several strategies, such as oversampling the minority class or undersampling the majority class, or using specialized evaluation metrics, would need to be considered to mitigate this issue. Additionally, during the exploratory data analysis phase, it was observed that 18 entries contained missing text data, which could potentially undermine the integrity of the dataset. Missing data, particularly in the *TEXT* column, is problematic, as it limits the available information for training the model. To ensure the analysis remains accurate and reliable, these entries were carefully removed from the dataset. By excluding incomplete data, the dataset was refined, providing a cleaner and more consistent input for the subsequent analysis [14]. This initial phase of data exploration and cleaning underscored the critical importance of preprocessing. The steps taken addressing class imbalance and removing missing entries are essential in preparing the data for feature extraction and model training. Proper data preprocessing is fundamental to ensuring the effectiveness of machine learning models, as it directly impacts their ability to generalize and make accurate predictions [15]. With a clean and well-balanced dataset, the system can be trained to accurately identify toxic messages, setting the stage for the development of a robust and reliable classification model.

B. Data Preprocessing

To prepare the dataset for machine learning and deep learning models, a series of comprehensive preprocessing steps were carried out to ensure the removal of noise, irrelevant information, and inconsistencies. These steps were essential to improve the quality of the data, thereby increasing the performance and reliability of the predictive models. The first crucial step in the preprocessing pipeline involved cleaning the text data. Since textual data is inherently messy and may contain unnecessary characters, this cleaning process was essential to streamline the input for the models. All characters in the text were converted to lowercase, ensuring uniformity and eliminating any inconsistencies caused by varying cases (e.g., “Toxic” vs. “toxic”). By standardizing the text in this manner, the models could process the words in a consistent format. Next, punctuation marks, such as commas, periods, exclamation points, and question marks, were removed. Punctuation does not contribute meaningful information to the context of the text when performing sentiment analysis or toxicity classification. By removing these characters, the dataset was further simplified, allowing

the models to focus on the core words and phrases that carry the most valuable semantic content. The subsequent step was the removal of stop words, which are common, high-frequency words that typically do not contribute to the meaning of a sentence, such as "and," "the," "is," "in," etc. Using the Natural Language Toolkit (NLTK), a popular Python library for natural language processing, a predefined list of stop words was employed to filter out these non-informative terms [16]. The removal of stop words reduced redundancy and improved the efficiency of the models by focusing the analysis on more meaningful, content-rich words. In addition to text-based transformations, the categorical labels in the TOXICITY column were converted into numerical values using label encoding. Machine learning algorithms typically require numerical data as input, and categorical labels like "toxic" and "non-toxic" are not directly compatible with most models. To address this, non-toxic messages were encoded as 0, and toxic messages were encoded as 1. This binary encoding was a critical transformation, enabling the models to interpret and process the toxicity labels effectively [17]. By carrying out these preprocessing steps—text cleaning, punctuation removal, stop word elimination, and label encoding the quality and consistency of the dataset were significantly enhanced. These transformations not only reduced unnecessary noise but also ensured the dataset was in a format suitable for advanced tasks like feature extraction and model training. The improved dataset formed the basis for the subsequent stages of the machine learning pipeline, including feature extraction and classification. These preprocessing techniques played a vital role in enabling the models to focus on essential aspects of the text data, ultimately facilitating accurate predictions and ensuring the development of a robust and reliable system for toxicity classification.

C. Feature Extraction

Once the dataset was cleaned and preprocessed, the next crucial step in the pipeline was feature extraction, which aimed to convert the textual data into a numerical representation suitable for machine learning and deep learning models. This was achieved using the **Term Frequency-Inverse Document Frequency (TF-IDF)** vectorization technique, a widely-used method in natural language processing (NLP) for transforming text data into numerical features while preserving the important characteristics of the text.

The **TF-IDF** method works by assigning a weight to each word in the corpus, reflecting its relative importance within the context of the entire dataset. This approach involves two components: **Term Frequency (TF)** and **Inverse Document Frequency (IDF)**. **TF** measures how often a term appears in a document, essentially quantifying the frequency of a word in the given text. **IDF** assigns higher weight to words that are less common across all documents in the dataset, effectively reducing the weight of words that appear frequently across multiple documents, as these are likely to be less informative in distinguishing between texts [18]. By combining these two measures, the TF-IDF vectorizer captures both the local significance of a word within a document and its global relevance across the entire dataset. The result is a sparse matrix where each row represents a document, and each column corresponds to a term, with the matrix values representing the TF-IDF weight for each term in each document. For the specific case of this study, the **TF-IDF vectorizer** was configured to extract the top 5,000 features from the text data. This selection was made to strike a balance between computational efficiency and the retention of critical information.

While the TF-IDF technique can generate thousands of features, limiting the number of features to the most significant 5,000 allowed for faster model training and reduced memory consumption, without sacrificing the model's ability to capture the essential characteristics of the text. The selection of these top features was based on their importance in terms of both frequency within individual documents and their ability to distinguish between toxic and non-toxic messages across the dataset. This transformation allowed the system to focus on the most relevant terms that were likely to contribute meaningfully to the classification task. By prioritizing these important features, the classification models were able to operate more efficiently and effectively, placing greater emphasis on the words that carry the most discriminatory power. Additionally, the TF-IDF approach helped to eliminate redundant or less significant elements, further enhancing the model's performance [19]. The resulting TF-IDF features were then used as input for the subsequent classification models, providing them with a refined and informative representation of the text data. This feature extraction process ensured that the models had access to a set of features that was both computationally manageable and rich in the information necessary for accurate prediction. By focusing on the most informative terms, the system was well-prepared to classify text messages as either toxic or non-toxic in a way that maximized predictive accuracy.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

D. Model Selection and classification

After preprocessing the dataset and extracting relevant features using the **TF-IDF** method, we proceeded to the critical step of model selection to classify messages as either toxic or non-toxic. For this task, two distinct approaches were chosen: a **Logistic Regression** model, representing traditional machine learning techniques, and a **Long Short-Term Memory (LSTM)** network, which is a deep learning approach designed to process sequential data [20]. The first model selected was **Logistic Regression**, which was chosen due to its simplicity, computational efficiency, and ability to handle high-dimensional data. Logistic Regression is a probabilistic model that estimates the likelihood of a sample belonging to a particular class (toxic or non-toxic) based on the features derived from the dataset. This model was particularly well-suited for this classification task because of its ability to quickly process large datasets while remaining computationally efficient. Given the high-dimensional nature of text data, which is represented by the TF-IDF features, Logistic Regression was able to identify key distinguishing terms that helped classify messages effectively. To train the Logistic Regression model, the **TF-IDF features** were used, with the aim of representing the relative importance of each word within the dataset. The model was trained over 1,000 iterations, optimizing the weights to minimize the loss function and ensuring the best fit for the classification task. This approach allowed the model to efficiently classify messages as either toxic or non-toxic based on the learned features. While Logistic Regression provided a solid foundation for the classification task, it was limited in its ability to handle the **context-dependent** and **sequential nature** of language, especially in the case of subtle or nuanced toxic expressions [21]. To address this limitation, we introduced a more advanced model based on **Long Short-Term Memory (LSTM)** networks. LSTMs, a type of **Recurrent Neural Network (RNN)**, are particularly effective at learning dependencies within sequential data, making them well-suited for natural language processing tasks [22]. Unlike Logistic Regression, which treats each word as an independent feature, LSTMs are designed to capture the relationships between words and their context within the entire sequence, enabling them to better understand the flow of conversations and detect toxicity that may not be immediately obvious. Although the LSTM model demonstrated great potential in capturing the sequential nature of language, it faced challenges in generalizing to simpler cases where toxicity is more direct. However, its ability to grasp context-dependent toxicity in longer and more complex messages proved beneficial, highlighting its strength in identifying nuanced forms of harmful language that may require a deeper understanding of word sequences.

IV. RESULTS AND DISCUSSION

The performance of the proposed system was evaluated using two models: Logistic Regression and Long Short-Term Memory (LSTM). Their effectiveness was assessed using metrics such as accuracy, precision, recall, and F1-score, and the results were further analyzed using visual representations. Figures 2 and 4 illustrate the performance metrics of Logistic Regression and LSTM, respectively, while Figure 3 presents the confusion matrix for Logistic Regression. Figure 5 provides a side-by-side performance comparison of both models.

A. Logistic Regression

The Logistic Regression model showcased impressive performance in classifying messages as toxic or non-toxic. As seen in **Fig. 2**, the model achieved an overall accuracy of 79.9%. It demonstrated excellent results in identifying non-toxic messages (Class 0), with a precision of 0.80, recall of 0.93, and F1-score of 0.86. This success highlights the model's ability to effectively classify the majority class, benefitting from the dataset's class distribution.

```
Accuracy:  0.7990515708358032
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.93      0.86      4448
           1       0.79      0.55      0.65      2300

    accuracy                           0.80      6748
   macro avg       0.80      0.74      0.76      6748
weighted avg       0.80      0.80      0.79      6748
```

Fig.2. Performance metrics of Logistic Regression

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The confusion matrix for Logistic Regression, depicted in **Fig. 3**, provides a detailed visualization of the model's performance across different classes. The matrix highlights the model's ability to classify non-toxic messages (Class 0) with a high degree of accuracy, as evidenced by a significant number of true positives. This indicates that the model was highly effective in identifying the majority class, which is consistent with its high precision (0.80), recall (0.93), and F1-score (0.86) for non-toxic messages, as seen in Figure 2. Additionally, the matrix shows a low rate of false positives for non-toxic messages, further reinforcing the model's reliability in correctly identifying harmless content. However, the model's performance on toxic messages (Class 1) was comparatively less robust. The recall for toxic messages was 0.55, indicating that nearly half of the toxic messages were misclassified as non-toxic. This resulted in an F1-score of 0.65, reflecting a balanced measure of the model's precision and recall for the minority class. The confusion matrix underscores this challenge, revealing a noticeable number of false negatives for toxic messages, which points to the model's difficulty in recognizing subtle or context-dependent expressions of toxicity. These cases often involve nuanced language or implicit cues, which can be challenging for traditional machine learning models to interpret. Despite these limitations, Logistic Regression remains a computationally efficient and reliable model, particularly for handling high-dimensional data such as text represented by TF-IDF features [23]. Its straightforward implementation and quick training make it a valuable baseline model for tasks involving large datasets. The insights derived from the confusion matrix and performance metrics emphasize the importance of addressing the challenges associated with class imbalance and the detection of nuanced toxicity, which could further enhance the model's effectiveness.

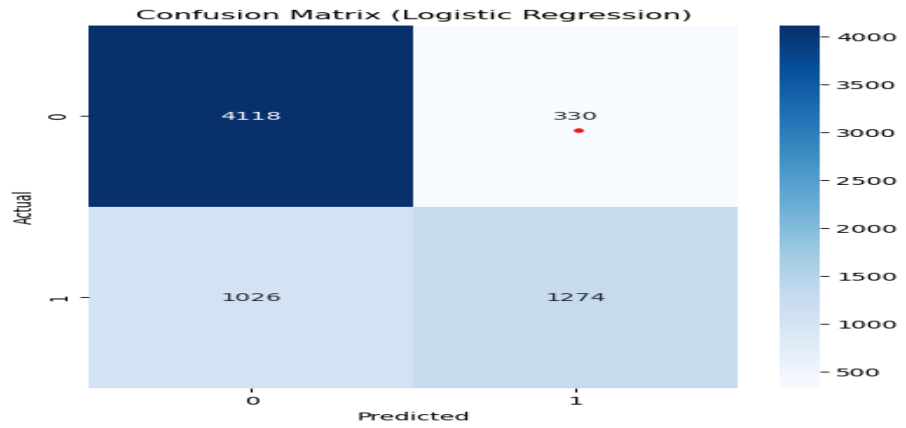


Fig.3. Confusion matrix for Logistic regression

B. Long Short-Term Memory(LSTM)

The LSTM model demonstrated its potential by leveraging its sequential data processing capabilities. As illustrated in fig.4, the model achieved an overall accuracy of 66%. Its performance for non-toxic messages (Class 0) was notable, with a precision of 0.66, recall of 1.00, and an F1-score of 0.79. These results highlight the LSTM's ability to capture patterns in sequential data, effectively reducing false negatives for non-toxic messages.

```
Accuracy:  0.6591582691167753
Classification Report:
                precision    recall  f1-score   support

           0       0.66      1.00      0.79      4448
           1       0.00      0.00      0.00      2300

    accuracy                           0.66      6748
   macro avg       0.33      0.50      0.40      6748
weighted avg       0.43      0.66      0.52      6748
```

Fig.4 Performance metrics of Logistic Regression

While the LSTM model struggled with toxic messages (Class 1), where precision, recall, and F1-score were 0.00, this outcome primarily reflects the challenge posed by class imbalance in the dataset. Although the performance for toxic

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

messages was limited, the model's design for understanding context and dependencies provides a strong foundation for improvement. Refining the model with advanced embeddings or balancing the dataset could unlock its full potential.

C. Comparative Analysis

The comparison between the two models is thoroughly illustrated in Figure 5, which presents their performance metrics side by side, offering a clear perspective on their respective strengths and weaknesses. Logistic Regression emerged as a reliable baseline model, excelling in the classification of non-toxic messages and demonstrating effective handling of class imbalance within the dataset [24]. With a macro-average F1-score of 0.76, Logistic Regression significantly outperformed the LSTM model, which achieved a macro-average F1-score of 0.40. Additionally, the weighted F1-scores, which account for the dataset's imbalanced distribution, further solidified the superiority of Logistic Regression, with a weighted F1-score of 0.79 compared to the LSTM model's 0.52. This indicates that Logistic Regression consistently delivered strong performance across all evaluation metrics, particularly for the majority class of non-toxic messages

However, despite these differences, the LSTM model showcased its unique strengths, particularly in understanding sequential patterns and contextual relationships within the data [25]. This capability is reflected in its high recall for non-toxic messages, which underscores its ability to capture the temporal and relational aspects of textual data. This contextual learning strength positions the LSTM model as a promising candidate for further improvements. Adjustments such as addressing class imbalance through techniques like oversampling the minority class, undersampling the majority class, or incorporating contextual embeddings like BERT could help the LSTM model realize its full potential [26].Such refinements may enable the LSTM model to achieve a better balance between precision and recall, especially for the minority class of toxic messages.

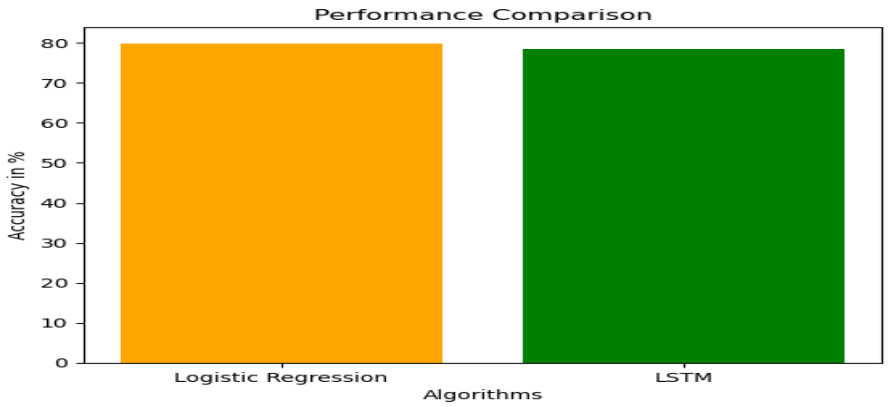


Fig. 5. Performance Comparison of Algorithms

D. Visual Analysis

The visualizations offer crucial insights into the performance of both models**.** Fig. 2 highlights the robust performance of Logistic Regression for non-toxic messages, showcasing its ability to classify the majority class with high precision and recall. This visualization aligns with the confusion matrix in Fig. 3 which reveals Logistic Regression's success in minimizing false positives. However, the matrix also points to the model's limitations in identifying toxic messages, as evidenced by a noticeable number of false negatives. These false negatives suggest that while Logistic Regression excels in classifying straightforward cases, it struggles with the nuanced expressions of toxicity often present in the minority class.

Fig. 4 emphasizes the LSTM model's potential for learning sequential patterns and capturing contextual relationships in the data. This capability, while not fully optimized in the current implementation, highlights the model's suitability for tasks requiring a deeper understanding of language flow and conversational context. Although the LSTM model achieved lower overall performance metrics compared to Logistic Regression, its high recall for non-toxic messages indicates its ability to generalize effectively for the majority class. This strength underscores the importance of leveraging sequential learning models like LSTM in scenarios where context plays a critical role in classification.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Finally, Fig. 5 provides a comprehensive comparative overview of the two models, clearly outlining their respective strengths and limitations. While Logistic Regression delivered superior overall performance, the LSTM model demonstrated complementary potential by excelling in areas that Logistic Regression struggled with, such as contextual understanding. This comparison suggests that combining the strengths of both models such as using Logistic Regression for baseline classification and augmenting it with the LSTM model's sequential learning capabilities could lead to a more robust and nuanced classification system [27]. These insights underscore the need for tailored approaches to improve both models, ensuring they perform optimally in detecting toxicity in varied and complex datasets.

V. FUTURE WORK

To enhance the performance and effectiveness of the model in addressing toxic content, future work should focus on several key areas. One of the main challenges in toxicity detection is dealing with imbalanced datasets, which can be mitigated through techniques such as oversampling, under sampling, or synthetic data generation [28-30]. These strategies help ensure the model receives a balanced distribution of toxic and non-toxic samples, which is crucial for effective training. Additionally, fine-tuning the hyper parameters of the model can have a significant impact on its performance. Using optimization techniques like grid search or Bayesian optimization to adjust parameters such as the learning rate, number of layers, dropout rates, and batch size can improve the model's ability to generalize and prevent over fitting, which can lead to better results on unseen data [31-33].

Furthermore, incorporating advanced contextual embeddings such as BERT into the model can provide a deeper understanding of the nuances in text, allowing the model to better capture both explicit and implicit forms of toxicity. BERT's ability to process entire sentences, considering both left and right contexts, makes it especially valuable in distinguishing between subtle and more overtly harmful content [34]. In addition, a hybrid approach combining the strengths of both LSTM and Logistic Regression models could provide a more robust solution. While LSTM networks excel at processing sequences and capturing long-term dependencies, Logistic Regression offers computational efficiency and effectiveness in linear classification. By combining these models, the system can take advantage of LSTM's deep contextual understanding and Logistic Regression's quick decision-making capabilities.

Domain-specific model customization is also an important consideration. Toxicity manifests differently across various platforms, such as social media, online forums, and gaming communities. Fine-tuning the model to accommodate platform-specific language, jargon, and communication patterns could significantly improve its performance [35]. Another promising avenue is transfer learning, where pre-trained models like BERT are fine-tuned on a specific toxicity detection dataset. This approach leverages the existing knowledge in these pre-trained models, reducing the need for large amounts of training data and enabling the model to quickly adapt to new tasks.

As the model is deployed in real-world scenarios, ensuring its decisions are interpretable becomes crucial. Techniques for model explain ability, such as attention mechanisms within the LSTM or using SHAP values, can help clarify the reasons behind the model's classification, fostering trust and transparency [36]. Finally, continuous evaluation of the model on real-world data will be essential to adapt to evolving communication patterns and new forms of toxic content. Regular retraining with newly labeled data, as well as measuring performance using various metrics like precision, recall, and F1-score, will help strike the right balance between sensitivity and specificity, ensuring the model remains effective over time. Through these strategies, the model's ability to detect both explicit and subtle toxic content will improve, making it more adaptable and reliable in a wide range of contexts.

VI. CONCLUSION

In conclusion, detecting toxic content on online platforms is a complex challenge that demands a balanced and multi-faceted approach. This paper has demonstrated the potential of combining sequence-based models like LSTM with computationally efficient classifiers such as Logistic Regression to enhance detection capabilities. While LSTM models effectively capture contextual and sequential nuances, Logistic Regression provides reliability and speed for high-dimensional data. Together, these approaches form a promising foundation for tackling both explicit and subtle forms of toxicity. The insights gained underscore the importance of addressing key challenges, such as class imbalance and the detection of nuanced toxic expressions, to further enhance model performance. Incorporating techniques such as

hyper parameter fine-tuning, domain-specific adaptations, and feedback loops can help ensure these systems remain robust and relevant in diverse applications. Equally critical is the emphasis on model explain ability and transparency, which are vital for fostering trust and fairness. Continuous evaluation and adaptation to real-world data will further solidify the effectiveness of toxicity detection systems. By combining technological advancements with ongoing refinements, it is possible to develop solutions that not only detect harmful content with greater precision but also contribute to creating safer and more inclusive online environments. The future of toxicity detection lies in sustaining these efforts and evolving alongside the changing dynamics of online communication.

REFERENCES

1. A. Ostayeva, Z. Kozhamkulova, Y. Aimakhanov, D. Abylkhassenova, A. Serik, and Y. Tenizbayev, "Utilizing machine learning and deep learning approaches for the detection of cyberbullying issues," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 6, 2024.
2. M. S. Islam, A. N. Orno, and M. Arifuzzaman, "Approach to social media cyberbullying and harassment detection using advanced machine learning," SSRN, 2024. [Online]. Available: https://ssrn.com/abstract=4705261
3. S. Mallappa, M. A. N. Saif, and H. D. E. Al-Ariki, "DEA-RNN: A hybrid deep learning approach for cyberbullying detection in Twitter social media platform."
4. V. Shah, A. Sinha, N. Navalkar, S. Gupta, P. Gonsalves, and A. Malik, "ML and natural language processing: Cyberbullying detection system for safer and culturally adaptive digital communities," J. Smart Internet Things, vol. 2023, no. 2, pp. 193–205, 2023.
5. S. Kaur, S. Singh, and S. Kaushal, "Deep learning-based approaches for abusive content detection and classification for multi-class online user-generated data," Int. J. Cogn. Comput. Eng., vol. 5, pp. 104–122, 2024.
6. J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on Twitter," arXiv preprint arXiv:1706.01206, 2017.
7. Z. Pitenis, M. Zampieri, and T. Ranasinghe, "Offensive language identification in Greek," arXiv preprint arXiv:2003.07459, 2020.
8. R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on Bengali social media comments using machine learning," Int. J. Cogn. Comput. Eng., vol. 4, pp. 21–35, 2023.
9. A. Ostayeva, Z. Kozhamkulova, Y. Aimakhanov, D. Abylkhassenova, A. Serik, and Y. Tenizbayev, "Utilizing machine learning and deep learning approaches for the detection of cyberbullying issues," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 6, 2024.
10. K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," Eur. Phys. J. Spec. Top., vol. 228, no. 10, pp. 2313–2324, 2019.
11. K. Alemerien, A. Al-Ghareeb, and M. Z. Alksasbeh, "Sentiment analysis of online reviews: A machine learning based approach with TF-IDF vectorization," J. Mobile Multimedia, pp. 1089–1116, 2024.
12. S. Farley, I. Coyne, and P. D'Cruz, "Cyberbullying at work: Understanding the influence of technology," Concepts, Approaches and Methods, pp. 233–263, 2021.
13. V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 4, pp. 42–47, 2012.
14. A. Deekshith, "Data engineering for AI: Optimizing data quality and accessibility for machine learning models," Int. J. Manag. Educ. Sustain. Dev., vol. 4, no. 4, pp. 1–33, 2021.
15. P. A. Brown and R. A. Anderson, "A methodology for preprocessing structured big data in the behavioral sciences," Behav. Res. Methods, vol. 55, no. 4, pp. 1818–1838, 2023.
16. B. Aklouche, I. Bounhas, and Y. Slimani, "Query expansion based on NLP and word embeddings," in Proc. TREC, Nov. 2018.
17. F. Matteucci, V. Arzamasov, and K. Böhm, "A benchmark of categorical encoders for binary classification," Adv. Neural Inf. Process. Syst., vol. 36, 2024.
18. L. Havrlant and V. Kreinovich, "A simple probabilistic explanation of term frequency-inverse document frequency (TF-IDF) heuristic (and variations motivated by this explanation)," Int. J. Gen. Syst., vol. 46, no. 1, pp. 27–36, 2017.
19. M. I. Alfarizi, L. Syafaah, and M. Lestandy, "Emotional text classification using TF-IDF (term frequency-inverse document frequency) and LSTM (long short-term memory)," JUITA: J. Informatika, vol. 10, no. 2, pp. 225–232, 2022.

20. H. Gasmi, J. Laval, and A. Bouras, "Information extraction of cybersecurity concepts: An LSTM approach," Appl. Sci., vol. 9, no. 19, p. 3945, 2019.
21. H. Gonaygunta, "Machine learning algorithms for detection of cyber threats using logistic regression," Dept. Inf. Technol., Univ. Cumberlands, 2023.
22. F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," J. Mach. Learn. Res., vol. 3, pp. 115–143, Aug. 2002.
23. S. S. Noureen, S. B. Bayne, E. Shaffer, D. Porschet, and M. Berman, "Anomaly detection in cyber-physical system using logistic regression analysis," in Proc. IEEE Texas Power Energy Conf. (TPEC), Feb. 2019, pp. 1–6.
24. M. Goswami and P. Sajwan, "A comparative analysis of sentiment analysis using RNN-LSTM and logistic regression," in Trends Wireless Commun. Inf. Secur.: Proc. EWCIS 2020, pp. 165–174, 2021.
25. H. Gasmi, J. Laval, and A. Bouras, "LSTM recurrent neural networks for cybersecurity named entity recognition," arXiv preprint arXiv:2409.10521, 2024.
26. J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying detection using pre-trained BERT model," in Proc. IEEE ICESC, Jul. 2020, pp. 1096–1100.
27. Z. M. ALbazzaz and O. B. Shukur, "Using LSTM network based on logistic regression model for classifying solar radiation time series," in Proc. Int. Conf. Explainable AI Digital Sustainability, Jun. 2024, pp. 375–388.
28. M. S. Shelke, P. R. Deshmukh, and V. K. Shandilya, "A review on imbalanced data handling using undersampling and oversampling technique," Int. J. Recent Trends Eng. Res., vol. 3, no. 4, pp. 444–449, 2017.
29. A. Y. C. Liu, "The effect of oversampling and undersampling on classifying imbalanced text datasets," 2004.
30. B. Santoso, H. Wijayanto, K. A. Notodiputro, and B. Sartono, "Synthetic over sampling methods for handling class imbalanced problems: A review," in IOP Conf. Ser.: Earth Environ. Sci., vol. 58, no. 1, p. 012031, Mar. 2017.
31. S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, "Applying Grid Search, Random Search, Bayesian Optimization, Genetic Algorithm, and Particle Swarm Optimization to fine-tune the hyperparameters of the ensemble of ML models enhances its predictive accuracy for mud loss," 2024.
32. W. Chen, T. Paraschivescu, and X. Can, "Practical Bayesian optimization of machine learning algorithms," Adv. Neural Inf. Process. Syst., vol. 4, pp. 2951–2959, 2012.
33. J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," J. Electron. Sci. Technol., vol. 17, no. 1, pp. 26–40, 2019.
34. S. Paul and S. Saha, "CyberBERT: BERT for cyberbullying identification," Multimed. Syst., vol. 28, no. 6, pp. 1897–1904, 2022.
35. B. Bhatia, A. Verma, Anjum, and R. Katarya, "Analysing cyberbullying using natural language processing by understanding jargon in social media," in Sustainable Adv. Comput.: Sel. Proc. ICSAC 2021, pp. 397–406, Singapore: Springer Singapore, 2022.
36. H. Kindbom, "Investigating the attribution quality of LSTM with attention and SHAP: Going beyond predictive performance," [Journal/Conference Name], vol. [X], no. [Y], pp. [Page Range], 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details