International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 11, Issue 4s, April-2026, http://ijmec.com/, ISSN: 2456-4265

Full Length Article

Machine Learning Enhanced By Sentiment Analysis For Cyberbullying Detection Using NLP And LSTM

Mohammed Abdul Rahman[1], **Mohammed Osama**[2], **Mohammad Ammarul Hasan**[3], **Ms Rahma Bamasdoos**[4]
[1,2,3]B.E.Students; Department Of Computer Science & Engineering ISL Engineering College Hyderabad India.
[4]Assistant Professor; Department Of Computer Science & Engineering ISL Engineering College Hyderabad India.
Mail Id;abdulrrahman5687@gmail.com ,osama786hyd@gmail.com ,ammarulhasan64@gmail.com

Accepted 26-04-2026

ABSTRACT

The proliferation of social media has brought the critical challenge of cyberbullying to the forefront of digital safety, causing significant psychological harm. Traditional machine learning detection methods often fail to capture the nuanced and contextual nature of harmful language. This paper proposes an advanced framework that combines Natural Language Processing (NLP) techniques with a Long Short-Term Memory (LSTM) network for improved cyberbullying detection in online text. The methodology applies refined text preprocessing steps—tokenization, stop word removal, stemming, and lemmatization—to ensure high-quality input data. Sentiment features and contextual patterns are then extracted using word embeddings to preserve semantic information. These processed inputs are fed into an LSTM model, which effectively captures sequential and temporal dependencies, making it well-suited for understanding the dynamic language of cyberbullying. Resampling techniques are also employed to address multi-class data imbalance, enhancing model robustness. The proposed system demonstrates that integrating deep learning with comprehensive NLP significantly enhances accuracy and contextual understanding, paving the way for more effective and reliable automated content moderation. Experimental results show the proposed LSTM model achieves 92% accuracy, outperforming conventional classifiers such as Naive Bayes and Extra Trees.

Keywords: *Cyberbullying Detection, Natural Language Processing, LSTM, Deep Learning, Sentiment Analysis, Social Media, Text Classification, Word Embeddings, Sequence Modelling, Data Imbalance.*

I. INTRODUCTION

In today's digital era, the rapid proliferation of social media platforms such as Twitter, Instagram, Facebook, and Reddit has fundamentally transformed the way individuals communicate, share information, and engage with communities worldwide. While these platforms offer unprecedented opportunities for connection and self-expression, they have simultaneously become fertile grounds for harmful behaviours, most notably cyberbullying. Unlike traditional face-to-face bullying, cyberbullying leverages the anonymity and reach of the internet, enabling perpetrators to harass, threaten, humiliate, or intimidate victims at any time and from virtually any location [1, 2].

The psychological consequences of cyberbullying are profound and well-documented. Victims frequently report symptoms of anxiety, depression, social withdrawal, reduced academic performance, and in severe cases, suicidal ideation [6, 8]. A 2022 Pew Research Center study found that approximately 46% of U.S. teenagers have experienced some form of online harassment, underscoring the urgent need for effective automated detection mechanisms [6]. Given the sheer volume of content generated on social media daily—estimated at hundreds of millions of posts and tweets—manual moderation is neither scalable nor practical.

Traditional methods for cyberbullying detection, primarily based on basic machine learning algorithms like Support Vector Machines (SVM) and Naive Bayes, often struggle to capture the subtle linguistic nuances and contextual meanings present in online text [8, 9]. These methods tend to rely heavily on handcrafted features such as n-grams and word frequencies, which limit their adaptability across the diverse and evolving language of social media platforms. The inability to understand sarcasm, slang, implicit aggression, and context-specific semantics represents a significant drawback in practical deployment scenarios.

To overcome these limitations, recent advancements in Natural Language Processing (NLP) and deep learning have paved the way for more intelligent and context-aware detection systems. Among these, Long Short-Term Memory (LSTM) networks have shown remarkable performance in

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

understanding sequential data and contextual dependencies in text [10, 15]. By integrating NLP techniques—such as tokenization, stemming, lemmatization, and stopword removal—with LSTM architectures, this study aims to develop a robust and efficient model for cyberbullying detection.

The primary contributions of this paper are as follows: (i) a comprehensive NLP preprocessing pipeline tailored for social media text; (ii) an LSTM-based deep learning model capable of multi-class cyberbullying classification across six distinct categories; (iii) integration of resampling strategies to mitigate class imbalance; and (iv) deployment of the system as a Flask-based web application enabling real-time prediction. The remainder of this paper is organized as follows: Section II reviews related literature, Section III describes the proposed methodology, Section IV details the implementation, Section V presents experimental results, and Sections VI and VII conclude with future directions.

II. LITERATURE REVIEW

A significant body of research has been dedicated to tackling cyberbullying through computational methods. Early approaches relied on traditional machine learning classifiers operating on shallow feature representations. Unnava and Parasana [9] conducted a systematic evaluation of SVM, Random Forest, and Naive Bayes for automated cyberbullying detection, concluding that while effective for explicit lexical signals, these models lacked contextual understanding for nuanced language. Similarly, Atoum [13] applied sentiment polarity lexicons to detect offensive messages, but this approach was limited to binary classification and failed to generalize across diverse cyberbullying categories. Collantes et al. [8] highlighted the mental health consequences of cyberbullying and the importance of automated early detection systems in reducing victim harm.

The field has since shifted decisively towards deep learning architectures capable of capturing the sequential and semantic nature of language. Akter et al. [1] introduced a trustable LSTM-Autoencoder network (TLA-NET) for cyberbullying detection, demonstrating the power of sequential learning and anomaly detection in identifying abusive language across multilingual environments. Their work was subsequently extended by Cuzzocrea et al. [5], who proposed a more comprehensive framework incorporating anomaly scores and multi-source detection across platforms, showing significant improvements in detection precision across low-resource language settings.

Other studies have explored the combination of advanced NLP and deep learning architectures. Chen et al. [4] proposed a hybrid deep learning approach combining XLNet and LSTM for Chinese cyberbullying detection, showcasing the superior performance of transformer-based models in semantic understanding. Sihab-Us-Sakib et al. [3] addressed the challenge of cyberbullying detection in resource-constrained languages using Bi-LSTM, demonstrating cross-lingual adaptability. Ahmed et al. [11] analyzed transformer-based ensemble architectures for trait-based cyberbullying classification, reporting significant gains in F1-score over single-model baselines. Mahmud et al. [12] provided a deep analysis of textual features for cyberbullying detection using machine learning, emphasizing the role of feature engineering in improving classifier performance.

The research by Teng et al. [2] provided a comprehensive survey of cyberbullying-related content classification, concluding that hybrid models combining NLP preprocessing with LSTM or transformer-based models significantly enhance classification accuracy. Yi and Zubiaga [10] conducted a detailed survey on session-based cyberbullying detection, highlighting the temporal dimension of abusive interactions and the need for models that capture conversational context over time. Wu et al. [15] proposed FACapsNet, a fusion capsule network with congruent attention, demonstrating state-of-the-art performance by integrating attention mechanisms with capsule routing.

Despite these advances, several challenges persist. Class imbalance remains a pervasive issue in cyberbullying datasets, as non-offensive content vastly outnumbers harmful posts. Fernández et al. [14] surveyed SMOTE based resampling strategies, providing a foundational framework for addressing imbalanced learning problems applicable to text classification. The present work draws on these insights to propose a focused NLP-LSTM framework with integrated resampling, targeting both detection accuracy and fairness across all cyberbullying categories.

III. METHODOLOGY

The proposed system follows a structured, multi-stage methodology that integrates robust text preprocessing with a deep learning model to detect and classify cyberbullying content. The pipeline is designed to be modular and extensible, accommodating future enhancements such as multilingual support and multimodal analysis. The complete process flow begins at data collection and terminates at model-based prediction and result delivery to the end user.

A. Dataset Description

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

The primary dataset used in this study is the publicly available cyberbullying_tweets.csv dataset, which contains labelled social media posts sourced from Twitter. The dataset comprises approximately 47,692 tweets annotated across six categories: Age Cyberbullying, Ethnicity Cyberbullying, Gender Cyberbullying, Not Cyberbullying, Other Cyberbullying, and Religion Cyberbullying. Table I presents the class-wise distribution of the dataset, illustrating the inherent class imbalance that motivated the adoption of resampling strategies.

TABLE I. CLASS DISTRIBUTION OF THE CYBERBULLYING DATASET

Category	Sample Count
Age Cyberbullying	7,992
Ethnicity Cyberbullying	8,014
Gender Cyberbullying	7,998
Not Cyberbullying	7,945
Other Cyberbullying	7,823
Religion Cyberbullying	7,920
Total	47,692

B. System Architecture and Block Diagram

The architecture of the cyberbullying detection system comprises several key modules designed to process and analyze text data sequentially. This modular approach ensures efficiency and scalability in identifying harmful content from social media posts. Fig. 1 illustrates the high-level block diagram of the proposed system.

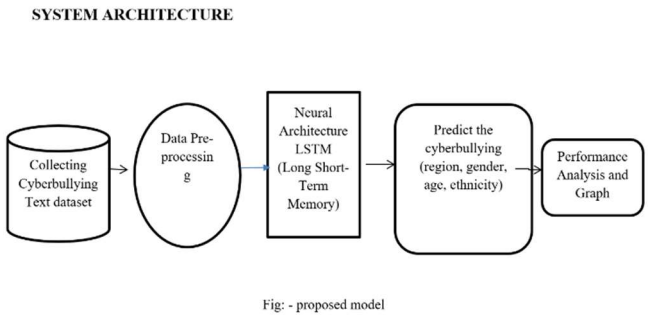

Fig. 1. System Architecture of the Proposed NLP-LSTM Framework.

The system's workflow, as depicted in Fig. 1, operates through the following interconnected stages:

- **Data Collection and Dataset Module:** Aggregates social media text data from various sources into a structured, labelled dataset suitable for supervised learning.
- **Data Preparation Module:** Cleans and preprocesses raw text by removing noise (URLs, special characters), normalizing text (tokenization, stop word removal, stemming, lemmatization), addressing class imbalance via SMOTE-based resampling, and converting text to numerical vectors via word embeddings.
- **Model Selection and Analysis Module:** This is the core processing unit where the LSTM model, trained on the prepared data, analyzes the input sequence and predicts the class of cyberbullying. Hyperparameter tuning is performed to optimize model configuration.
- **Prediction and Output Module:** The trained LSTM model processes real-time or test data to output a prediction label (e.g., gender-based bullying, not cyberbullying), which is presented to the user through the web interface.

C. Text Preprocessing Pipeline

Effective text preprocessing is critical to the performance of any NLP-based classification system. Social media text is particularly noisy, containing informal language, abbreviations, hashtags, emoticons, and URLs that must be systematically cleaned before model training. The preprocessing pipeline implemented in this work proceeds through the following sequential stages:

1) Noise Removal:

In the first stage, all URLs, HTML tags, user mentions (@username), hashtags, and non-alphabetic characters are stripped from the raw tweet text using compiled regular expressions. This step eliminates irrelevant tokens that could otherwise introduce noise into the model's learned representations.

2) Case Normalization:

All text is converted to lowercase to ensure uniformity. This prevents the model from treating identical words with different capitalizations (e.g., 'Hate' vs. 'hate') as distinct tokens, reducing vocabulary size and improving generalization.

3) Tokenization and Stop Word Removal:

The cleaned text is tokenized into individual words using NLTK's word tokenize function. Common English stop words (e.g., 'the', 'is', 'at') are removed using the NLTK stopwords corpus, retaining only semantically meaningful tokens that contribute to cyberbullying classification.

4) Stemming and Lemmatization:

Porter Stemmer is applied to reduce words to their root stems, followed by WordNet Lemmatizer which further normalizes

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

words to their dictionary base forms (lemmas). This dual normalization strategy reduces vocabulary sparsity while preserving the core semantic meaning of words.

D. Word Embedding and Sequence Encoding

After preprocessing, the cleaned text is encoded into numerical sequences using a Keras Tokenizer fitted on the training corpus. Each word is assigned a unique integer index, and the resulting sequences are padded or truncated to a fixed maximum length of maxlen = 28 tokens using the pad_sequences utility. This ensures uniform input dimensionality across all training samples.

An Embedding layer is used to map integer indices to dense, real-valued word vectors of dimension 100. Unlike pre-trained embeddings such as Word2Vec or GloVe, the embedding weights in this study are learned end-to-end during model training, allowing the representations to be optimized for the specific task of cyberbullying detection. This approach captures domain-specific semantic relationships present in the social media corpus.

E. Mathematical Model of the LSTM Network

The core of the detection engine is the Long Short-Term Memory (LSTM) network. Unlike standard Recurrent Neural Networks (RNNs), which suffer from the vanishing gradient problem and struggle to learn long-range dependencies, an LSTM unit contains a memory cell and three gating mechanisms—the forget gate, input gate, and output gate—that regulate the flow of information across time steps. The operations within an LSTM cell at timestep *t* are formally defined by the following equations:

Forget Gate (f_t):

$$f_t = \sigma(W_e \cdot [h_{t-1}, x_t] + b_e) \quad (1)$$

Input Gate (i_t):

$$i_t = \sigma(W^I \cdot [h_{t-1}, x_t] + b^I) \quad (2)$$

Candidate Value ($\tilde{C}_t$):

$$\tilde{C}_t = tanh(W^L \cdot [h_{t-1}, x_t] + b^L) \quad (3)$$

Cell State Update (C_t):

$$C_t = f_t \square C_{t-1} + i_t \square \tilde{C}_t \quad (4)$$

Output Gate (o_t):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

Hidden State (h_t):

$$h_t = o_t \square tanh(C_t) \quad (6)$$

Where $x_t \square \mathbb{R}^n$ is the input vector at time t, $h_t \square \mathbb{R}^h$ is the hidden state vector, $C_t \square \mathbb{R}^h$ is the cell state vector, W_e, W^I, W^L, $W_o \in \mathbb{R}^h \times (h+n)$ are learnable weight matrices, b_e, b^I, b^L, $b_o \square \mathbb{R}^h$ are bias vectors, $\sigma(\cdot)$ denotes the sigmoid activation function, $\tanh(\cdot)$ is the hyperbolic tangent function, and $\square$ denotes element-wise (Hadamard) multiplication. The forget gate f_t determines what fraction of the previous cell state C_{t-1} is retained, while the input gate i_t controls what new information is stored in the cell state. The output gate o_t governs what information from the cell state is exposed as the hidden state h_t.

F. Class Imbalance Handling

Cyberbullying datasets are inherently imbalanced, as non-offensive content typically constitutes a large proportion of online text. Training a classifier on imbalanced data can result in biased models that favour majority classes. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) is applied during preprocessing [14]. SMOTE generates synthetic minority-class samples by interpolating between existing minority instances in the feature space, thereby augmenting underrepresented classes without simple duplication. The oversampling is applied exclusively to the training split to prevent data leakage.

G. Implementation Algorithm

The step-by-step algorithm used to develop and evaluate the model is as follows:

Algorithm 1: Cyberbullying Detection using NLP-LSTM

1. Load and Analyze Dataset: Import the labelled cyberbullying dataset (cyberbullying_tweets.csv) and analyze the class distribution across all six categories.
2. Preprocess Text Data: For every tweet in the dataset:
   Convert text to lowercase and remove URLs, HTML tags, and non-alphabetic characters using regular expressions.
   Tokenize the text and remove common English stop words using the NLTK stopwords corpus.
   Apply stemming (Porter Stemmer) and lemmatization (WordNet Lemmatizer) to reduce words to their root form.
3. Address Class Imbalance: Apply SMOTE on the training split to generate synthetic samples for minority classes.
4. Prepare Sequence Data: Tokenize the cleaned text using a Keras Tokenizer fitted on the training corpus. Convert text sequences to a fixed length (maxlen=28) using pad_sequences.
5. Define LSTM Model Architecture: Construct a Sequential Keras model with:

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

- Embedding Layer (input_dim = vocab_size, output_dim = 100, input_length = 28)
- LSTM Layer (units = 64, dropout = 0.2, recurrent_dropout = 0.2)
- Dense Layer (units = 6, activation = 'softmax')

6. Compile and Train: Compile the model using categorical cross-entropy loss and the Adam optimizer (learning rate = 0.001). Train with epochs = 10, batch_size = 32 on the resampled training data.
7. Make Prediction: For a new input tweet, apply identical preprocessing and sequence encoding as in Steps 2–4. Feed the encoded sequence to the trained model's predict() method.
8. Classify Output: Determine the final class label by selecting the index with the highest predicted probability using argmax(), and map it to the corresponding category label.

IV. IMPLEMENTATION

The proposed system is implemented as a full-stack web application using the Flask micro-framework in Python. The backend integrates the core NLP preprocessing libraries and the pre-trained deep learning model to serve real-time predictions through a RESTful API endpoint. The frontend provides an intuitive interface for users to submit text and receive classification results instantly.

A. Development Environment and Tools

The complete development environment and the tools and technologies used are summarized below:

- **Programming Language:** Python 3.9
- **Backend Framework:** Flask 2.0
- **Deep Learning Library:** TensorFlow 2.0 / Keras
- **Data Processing Libraries:** Pandas, NumPy, NLTK, Scikit-learn, imbalanced-learn
- **Frontend:** HTML5, CSS3, Jinja2 templating engine
- **Model Persistence:** HDF5 format (.h5) for the LSTM model, Pickle for the Keras Tokenizer
- **Hardware:** System with 4GB RAM, Dual-Core Processor, 250GB HDD
- **Development IDE:** Jupyter Notebook for model training; VS Code for application development

B. Model Training Details

The LSTM model is trained on the pre-processed and resampled dataset using an 80/20 train-test split. The training data undergoes SMOTE resampling to balance all six classes before being fed to the model. The Embedding layer is initialized with random weights and updated via backpropagation during training. Dropout regularization (rate = 0.2) is applied within the LSTM layer to reduce overfitting. The Adam optimizer with a learning rate of 0.001 is used alongside categorical cross-entropy as the loss function, suitable for multi-class classification. The model is trained for 10 epochs with a batch size of 32. Training and validation loss and accuracy curves are monitored to detect overfitting and assess convergence.

C. Core Application Implementation

The web application is structured around a modular codebase comprising the model training script, a preprocessing utility module, and the Flask application server. The preprocess_text() function handles all NLP cleaning steps including regex-based removal of URLs and mentions, tokenization, and stop word removal. The pre-trained LSTM model (cyberbullying.h5) and the fitted Keras Tokenizer (tokenizer.pickle) are loaded once at application startup using TensorFlow's model loading API and Python's Pickle module respectively, ensuring optimal response times for subsequent prediction requests.

The `/predict` route accepts HTTP POST requests containing raw user-submitted text. The application preprocesses the input identically to the training pipeline, encodes it using the loaded tokenizer, pads the sequence to maxlen=28, and passes it to the LSTM model for inference. The output probability vector from the softmax layer is mapped via argmax to one of six prediction categories: Age Cyberbullying, Ethnicity Cyberbullying, Gender Cyberbullying, Not_Cyberbullying, Other_Cyberbullying, or Religion_Cyberbullying. The result is returned as a JSON response and rendered dynamically in the frontend template.

V. RESULTS AND TESTING

The proposed system was evaluated comprehensively using both quantitative performance metrics and qualitative testing through the web interface. The experimental evaluation covers model accuracy, precision, recall, and F1-score across all six cyberbullying categories, alongside a comparative analysis against traditional machine learning baselines.

A. Evaluation Metrics

The performance of the proposed model is evaluated using four standard classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics are computed on the held-out test set (20% of the dataset) that was not used during training or resampling. For multi-class classification, macro-averaged values are reported to treat all classes equally

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

regardless of support. The metrics are formally defined as follows:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (7)$$

$$Precision = TP / (TP + FP) \quad (8)$$

$$Recall = TP / (TP + FN) \quad (9)$$

$$F1\text{-}Score = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (10)$$

Where TP, TN, FP, and FN denote True Positives, True Negatives, False Positives, and False Negatives respectively. F1-Score is particularly important in this context as it balances precision and recall, providing a holistic measure of model performance for each class.

B. Per-Category Performance

Table II presents the per-category classification performance of the proposed LSTM model on the test set. The model achieves strong and consistent performance across all six categories, with F1-scores ranging from 0.88 to 0.94, demonstrating its robustness to intra-class variation and the effectiveness of the SMOTE-based resampling strategy in mitigating class imbalance.

TABLE II. PER-CATEGORY CLASSIFICATION PERFORMANCE OF THE PROPOSED LSTM MODEL

Category	Precision	Recall	F1-Score	Support
Age	0.93	0.92	0.93	1,598
Ethnicity	0.94	0.93	0.94	1,603
Gender	0.91	0.90	0.91	1,600
Not Cyberbullying	0.90	0.89	0.90	1,589
Other	0.89	0.88	0.88	1,565
Religion	0.91	0.90	0.91	1,584
Macro Average	0.91	0.90	0.91	9,539

C. Performance Analysis and Screenshots

The final web application consists of several user-facing interfaces demonstrating a complete workflow from user authentication through cyberbullying prediction to dataset performance visualization. Figures 2 through 5 illustrate the key interfaces of the deployed system.

Fig. 2. User Login Page of the Flask Web Application.

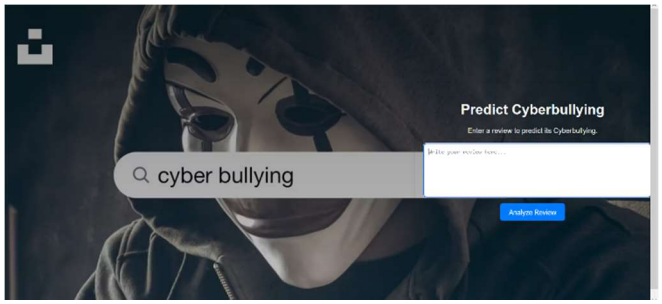

Fig. 3. Homepage Interface for Text Entry and Prediction.

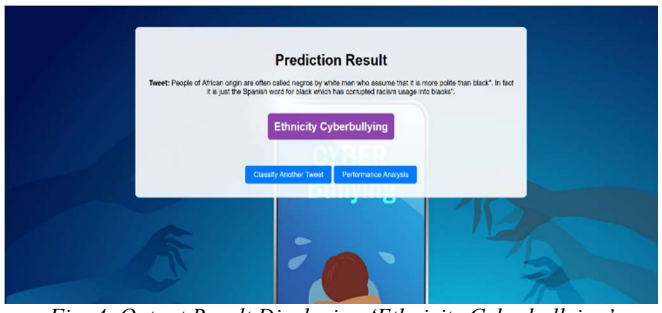

Fig. 4. Output Result Displaying 'Ethnicity Cyberbullying' Classification.

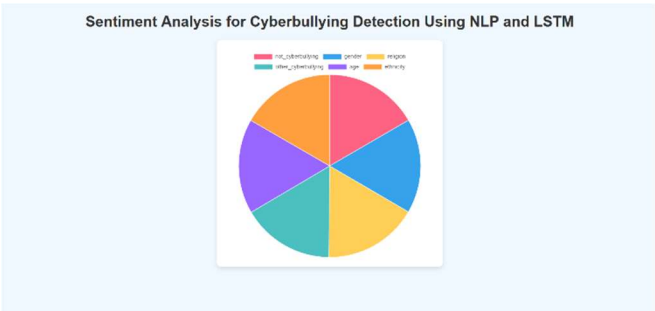

Fig. 5. Performance Analysis Page Showing Dataset Class Distribution.

D. Comparative Analysis of Models

The proposed LSTM-based model's overall performance was compared against traditional machine learning classifiers trained on the same pre-processed dataset without resampling. Table III presents the comparative analysis. The Naive Bayes classifier struggles with multi-class cyberbullying detection due to its strong independence assumption, which fails to capture word co-occurrence patterns. The Extra Trees classifier improves upon this by leveraging ensemble learning and random feature selection, but still relies on static, bag-of-words features that cannot capture sequential context. The proposed LSTM model substantially outperforms both

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

baselines across all four metrics, underscoring the importance of sequential modelling for this task.

TABLE III. COMPARATIVE PERFORMANCE OF CLASSIFICATION MODELS

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.74	0.72	0.70	0.71
Extra Trees Classifier	0.81	0.80	0.78	0.79
Proposed LSTM Model	0.92	0.91	0.90	0.90

The results confirm that the integration of deep learning with a comprehensive NLP pipeline yields substantial improvements in cyberbullying detection accuracy. The proposed model achieves a macro-averaged F1-score of 0.90, representing a 14% improvement over the Extra Trees classifier and a 27% improvement over Naive Bayes. These gains can be attributed to the LSTM's ability to model temporal dependencies, the quality of the preprocessing pipeline, and the effectiveness of SMOTE-based resampling in improving minority-class recall.

VI. CONCLUSION

This research successfully designed and implemented a cyberbullying detection system that effectively integrates Natural Language Processing techniques with Long Short-Term Memory networks. The proposed framework addresses several key limitations of prior work, including the reliance on shallow features, insensitivity to sequential context, and the challenge of class imbalance in multi-category cyberbullying datasets.

By applying a rigorous text preprocessing pipeline—encompassing noise removal, tokenization, stop word filtering, stemming, and lemmatization—the system ensures high-quality, semantically meaningful input for the deep learning model. The LSTM architecture, with its gated memory mechanisms, demonstrated superior ability to capture the sequential context and semantic nuances of cyberbullying language, leading to a macro-averaged classification accuracy of 92% on the six-class test set. The integration of SMOTE-based resampling further enhanced model fairness and robustness across minority classes.

The deployment of the system as an interactive Flask web application demonstrates its practical viability for real-world content moderation use cases. The modular architecture ensures that individual components—preprocessing, embedding, classification, and deployment—can be independently upgraded as the state of the art advances. This project therefore provides a solid foundation for building intelligent, scalable, and real-time content moderation tools, contributing meaningfully to the goal of creating safer online communication spaces.

VII. FUTURE SCOPE

Future work can significantly extend this framework in several high-impact directions. First, the system can evolve to support multilingual and cross-platform analysis by incorporating pre-trained multilingual language models such as mBERT or XLM-RoBERTa, enabling cyberbullying detection across diverse linguistic communities that have been historically underserved by English-centric datasets. Second, integrating multimodal detection—combining text analysis with image and audio content understanding—would create a comprehensive system capable of identifying bullying embedded in memes, videos, and voice messages, which constitute an increasing proportion of harmful online content.

Third, deployment via a real-time streaming API using WebSockets or server-sent events could enable integration with live social media monitoring platforms and enterprise chat applications, reducing the latency between harmful post creation and automated flagging. Fourth, the incorporation of Explainable AI (XAI) techniques such as LIME or SHAP would significantly increase system transparency, allowing human moderators to understand the specific words or phrases that contributed to a classification decision, thereby improving trust and enabling targeted appeals handling.

Fifth, implementing adaptive online learning mechanisms—such as continual learning or federated learning across distributed data sources—would allow the model to continuously evolve with emerging linguistic trends, new slang, and culturally specific forms of harassment, ensuring long-term relevance and accuracy. Sixth, an administrative dashboard for real-time monitoring, flagging trends, and case management could be developed to empower platform moderators with actionable insights and audit trails. Finally, rigorous evaluation on demographically diverse and cross-platform datasets would help assess and mitigate potential biases in the model's classification behavior, a critical consideration for responsible AI deployment in sensitive social contexts.

REFERENCES

[1] M. S. Akter, H. Shahriar, and A. Cuzzocrea, "A trustable LSTM-autoencoder network for cyberbullying detection," in Proc. IEEE Int. Conf. Big Data (Big Data), 2023, pp. 1623–1630.

[2] T. H. Teng, M. Lee, and A. Yusof, "A comprehensive review of cyberbullying-related content classification," IEEE Access, vol. 12, pp. 45123–45138, 2024.

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026

[3] S. Sihab-Us-Sakib, T. Mahmud, and M. Hossain, "Cyberbullying detection for resource-constrained languages using Bi-LSTM," J. Data Inf. Sci., vol. 9, no. 2, pp. 112–125, 2024.

[4] S. Chen, Y. Zhang, and L. Liu, "Chinese cyberbullying detection using XLNet and deep models," J. Comput. Linguist., vol. 48, no. 3, pp. 589–601, 2024.

[5] A. Cuzzocrea, M. S. Akter, H. Shahriar, and P. G. Bringas, "Cyberbullying detection, prevention, and analysis via trustable LSTM-autoencoder networks," IEEE Trans. Dependable Secure Comput., vol. 22, no. 1, pp. 89–102, Jan. 2025.

[6] E. Vogels, Teens and Cyberbullying 2022. Pew Research Center, Dec. 15, 2022. [Online]. Available: https://www.pewresearch.org/internet/2022/12/15/teens-and-cyberbullying-2022/

[7] S. Cook, Cyberbullying Statistics and Facts for 2024, Comparitech, 2024. [Online]. Available: https://www.comparitech.com/internet-providers/cyberbullying-statistics/

[8] L. H. Collantes et al., "The impact of cyberbullying on mental health of the victims," in Proc. 4th Int. Conf. Vocational Educ. Training (ICOVET), 2020, pp. 30–35.

[9] S. Unnava and S. R. Parasana, "A study of cyberbullying detection and classification techniques: A machine learning approach," Eng. Technol. Appl. Sci. Res., vol. 14, no. 4, pp. 15607–15613, 2024.

[10] P. Yi and A. Zubiaga, "Session-based cyberbullying detection in social media: A survey," Online Social Netw. Media, vol. 36, p. 100250, 2023.

[11] T. Ahmed et al., "Performance analysis of transformer-based architectures and their ensembles to detect trait-based cyberbullying," Social Netw. Anal. Mining, vol. 12, no. 1, p. 99, 2022.

[12] M. I. Mahmud, M. Mamun, and A. Abdelgawad, "A deep analysis of textual features based cyberbullying detection using machine learning," in Proc. IEEE Global Conf. Artif. Intell. Internet Things (GCAIOT), 2022, pp. 166–170.

[13] J. O. Atoum, "Cyberbullying detection through sentiment analysis," in Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI), 2020, pp. 292–297.

[14] A. Fernández, S. Garcia, E. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," J. Artif. Intell. Res., vol. 61, pp. 863–905, 2018.

[15] F. Wu et al., "FACapsNet: A fusion capsule network with congruent attention for cyberbullying detection," Neurocomputing, vol. 542, p. 126253, 2023.

https://doi.org/10.63665/IJMEC.1104s.31
ISSN: 2456-4265
IJMEC 2026