ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-1/438-444
B.Siddarth Reddy *et. al.,* / International Journal of Engineering & Science Research

Full Length Article

Machine Learning Enhanced By Sentiment Analysis For Cyberbullying Detection Using NLP

K.Anil Kumar[1],B.Siddarth Reddy[2],M.Havish Reddy[3],M.Nikhitha[4]
[1]Assistant Professor; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.
[2,3,4]B.Tech Students; Department Of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, India.
Mail Id;miryalanikhitha@gmail.com[4]
Accepted 25-03-2026

Abstract
Cyberbullying has become a significant concern in the modern digital environment, negatively impacting individuals and society at large. Detecting harmful interactions on social media platforms is therefore essential, as these platforms represent a major channel of online communication. Conventional detection techniques relying on machine learning methods and pre-trained language models often struggle with computational overhead and limited capability to interpret subtle linguistic variations. To address these limitations, this study introduces an enhanced cyberbullying detection framework that integrates Natural Language Processing (NLP) techniques with Long Short-Term Memory (LSTM) networks.The proposed approach incorporates comprehensive text preprocessing procedures, including tokenization, stop-word elimination, stemming, and lemmatization, to generate clean and meaningful input data. Semantic and sentiment-based features are extracted using embedding strategies that preserve contextual relationships among words. These representations are subsequently processed by an LSTM model, which is capable of learning sequential dependencies and temporal patterns in textual data, thereby improving the detection of complex cyberbullying expressions.Furthermore, to mitigate class imbalance in multi-class classification scenarios, appropriate resampling strategies are applied, enhancing model stability and reducing bias. Experimental observations indicate that the integration of deep learning with structured NLP preprocessing improves both accuracy and contextual understanding, making the proposed framework effective for identifying cyberbullying content in online communications.

Keywords— *Cyberbullying Detection, Natural Language Processing (NLP), Long Short-Term Memory (LSTM), Text Classification, Sentiment Analysis, Deep Learning.*

INTRODUCTION

The rapid expansion of digital communication technologies and social media platforms has transformed how individuals interact, share opinions, and exchange information. While these advancements have improved connectivity, they have also contributed to the emergence of harmful online behaviors, particularly cyberbullying. Cyberbullying refers to the intentional use of digital platforms to intimidate, harass, threaten, or humiliate individuals. Such behavior often results in serious psychological, emotional, and social consequences for victims. The anonymity and widespread reach of online platforms make it difficult to identify and control abusive interactions effectively.Conventional cyberbullying detection approaches mainly rely on traditional machine learning algorithms. These techniques often depend on manually engineered features and predefined rules, which restrict their ability to adapt to evolving linguistic styles and contextual meanings in online communication. Social media text frequently includes slang, sarcasm, abbreviations, and implicit expressions, making it challenging for traditional models to achieve high detection accuracy.

Recent developments in Natural Language Processing (NLP) and deep learning have enabled the creation of more intelligent and adaptive text classification systems. Among various deep learning architectures, Long Short-Term Memory (LSTM) networks have demonstrated strong performance in processing sequential textual data. LSTM models are capable of capturing contextual dependencies and understanding relationships between words across a sequence, making them suitable for identifying complex cyberbullying patterns.

This study proposes an advanced cyberbullying detection framework that integrates comprehensive NLP preprocessing techniques—such as tokenization, stopword removal, stemming, and lemmatization—with an LSTM-based deep learning model. Additionally, word embedding methods are employed to preserve semantic relationships and improve contextual understanding. To address class imbalance commonly observed in social media datasets, resampling techniques are incorporated to ensure balanced learning. The proposed system aims to improve detection accuracy, enhance contextual interpretation, and provide a reliable solution for identifying harmful content, thereby contributing to a safer digital ecosystem.

Scope of the Project

The scope of this project centers on designing and implementing an intelligent cyberbullying detection system using advanced NLP and LSTM techniques. The proposed framework focuses on identifying and classifying abusive or harmful textual content obtained from social media platforms. It is designed to process diverse linguistic variations, including informal language, slang, and contextual expressions commonly used in online communication.The system emphasizes efficient text preprocessing, semantic embedding, and sequential analysis to enhance detection performance. The proposed architecture can be adapted for multiple social media platforms and supports extension to different languages. Furthermore, the framework provides a foundation for integrating real-time monitoring, automated reporting, and content moderation mechanisms. Overall, the project contributes to improving online safety by reducing harmful interactions in digital communication environments.

Objective

The main objective of this study is to develop a deep learning-based cyberbullying detection model capable of accurately identifying abusive content in online text. The proposed system aims to integrate NLP preprocessing techniques with LSTM networks to capture contextual and emotional characteristics of social media messages.

Specific objectives of the project include:

- To preprocess textual data using tokenization, stemming, lemmatization, and stopword removal
- To extract semantic features using word embedding techniques
- To design an LSTM-based architecture for sequential text classification
- To detect multiple forms of cyberbullying, including insults, harassment, and threats
- To reduce noise and ambiguity through comprehensive text cleaning methods
- To address class imbalance using resampling strategies
- To improve classification accuracy while maintaining computational efficiency
- To evaluate the system using benchmark datasets
- To enhance contextual understanding of abusive language patterns
- To incorporate sentiment-related features for emotional tone analysis
- To create a scalable model adaptable to evolving linguistic trends
- To contribute toward safer social media environments
- To provide a foundation for future research in automated content moderation

Existing System

Existing cyberbullying detection systems primarily employ traditional machine learning algorithms such as Support Vector Machines (SVM), Naïve Bayes, Random Forest, and Extra Trees classifiers. These approaches rely heavily on handcrafted features derived from textual data, including n-grams, word frequency counts, and sentiment scores. While such models can achieve moderate performance, they often fail to capture deeper contextual and semantic relationships within social media text.

Another limitation of traditional systems is their inability to adapt to dynamic language usage, including sarcasm, abbreviations, and implicit forms of aggression. Additionally, many existing approaches suffer from class imbalance problems, particularly in multi-class classification scenarios where certain categories of cyberbullying occur less frequently. The absence of sequential modeling further restricts their ability to understand relationships across words and sentences.

Although conventional machine learning models offer faster training and lower computational requirements, they lack the capability to model temporal dependencies in text. Consequently, their performance in detecting complex cyberbullying patterns remains limited. These shortcomings highlight the need for more advanced deep learning approaches capable of understanding contextual and sequential information in online communication.

Proposed System

The proposed framework utilizes a deep learning approach that integrates advanced NLP preprocessing with an LSTM-based architecture for cyberbullying detection. Initially, the system performs data cleaning using tokenization, stopword removal, stemming, and lemmatization. These steps normalize textual inputs and remove noise, ensuring consistent data representation.

The processed text is then converted into dense vector representations using word embedding techniques. These embeddings preserve semantic relationships among words, enabling better contextual understanding. The LSTM network processes these sequences and captures long-term dependencies within text, making it effective for analyzing conversational patterns where cyberbullying may appear across multiple sentences.

To address dataset imbalance, resampling strategies are applied during training to ensure fair representation of all classes. The integration of NLP preprocessing, embedding methods, and LSTM modeling improves classification accuracy and contextual awareness. The proposed system provides a scalable and intelligent solution suitable for real-world cyberbullying detection applications.

Proposed System Advantages

- Efficient sequential data modeling using LSTM
- Improved contextual understanding through NLP preprocessing
- Automatic feature extraction via deep learning
- Higher accuracy in cyberbullying detection
- Balanced classification using resampling techniques
- Robust performance across diverse textual inputs

PROJECT DESCRIPTION

The proposed project focuses on developing an intelligent cyberbullying detection system by integrating Natural Language Processing (NLP) techniques with a Long Short-Term Memory (LSTM) deep learning architecture. The increasing usage of social media platforms has amplified the occurrence of cyberbullying, which can negatively impact individuals' emotional and psychological well-being. To address this issue, the system aims to automate the identification of harmful textual content generated through online communication such as posts, comments, and messages.

The development process begins with collecting labeled textual datasets containing examples of both cyberbullying and non-cyberbullying content. These datasets serve as the foundation for training and evaluating the model. During preprocessing, multiple NLP techniques including tokenization, stopword removal, stemming, and lemmatization are applied to clean and normalize the text. This ensures that irrelevant noise is removed and meaningful features are retained.

To represent textual data numerically, embedding techniques such as Word2Vec or GloVe are employed. These embeddings preserve semantic relationships and contextual information within the text. The generated feature vectors are then provided as input to an LSTM network, which is capable of learning sequential dependencies and capturing emotional tone within sentences. This property makes LSTM particularly effective in identifying subtle patterns in abusive language.

The model is trained and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. To address class imbalance in the dataset, resampling techniques such as SMOTE are applied to improve fairness in classification. The architecture is designed to support real-time prediction, allowing the system to analyze live text inputs efficiently. The final objective is to develop a scalable solution capable of automatically detecting cyberbullying content and promoting safer online interactions. Additionally, the framework can be extended to support multiple languages and social media platforms.

Methodologies

Existing Technique

The Extra Trees Classifier, also known as Extremely Randomized Trees, is an ensemble learning algorithm based on multiple decision trees. It improves prediction accuracy by combining outputs from several trees. Unlike Random Forests, Extra Trees introduces additional randomness by selecting feature thresholds randomly during splitting. This reduces overfitting and enhances generalization.

In cyberbullying detection, Extra Trees is used with features such as bag-of-words, TF-IDF, and sentiment scores. Although the algorithm is efficient and easy to implement, it does not capture sequential relationships in text. Consequently, it may fail to recognize contextual meaning, sarcasm, or implicit aggression. These limitations reduce its effectiveness in detecting complex cyberbullying patterns.

Proposed Technique Used or Algorithm Used

Long Short-Term Memory (LSTM) is an advanced Recurrent Neural Network architecture designed for sequential data modeling. Traditional RNNs often suffer from vanishing or exploding gradient problems, which limit their ability to learn long-term dependencies. LSTM addresses this limitation through memory cells and gating mechanisms, including input, forget, and output gates, which regulate information flow.

In the proposed cyberbullying detection system, LSTM analyzes social media text sequences to capture contextual relationships among words. This allows the model to understand tone, sentiment, and intent behind messages. When combined with NLP preprocessing techniques such as tokenization, stopword removal, stemming, and word embeddings, LSTM provides improved classification performance. The deep learning-based approach enables accurate and real-time detection of harmful content across diverse communication platforms.

REQUIREMENTS ENGINEERING

Requirements engineering defines the necessary specifications for designing and implementing the proposed cyberbullying detection system. The performance of the system depends on the quality of extracted textual features and the learning capability of the classification model. By leveraging semantic representations and deep learning techniques, the proposed framework aims to minimize classification errors and improve detection accuracy. The integration of Natural Language Processing (NLP) and Long Short-Term Memory (LSTM) networks enhances contextual understanding, resulting in competitive performance compared to existing approaches. The requirements outlined in this chapter provide a structured foundation for system development and deployment.

Hardware Requirements

Hardware requirements specify the minimum computational resources needed for system development, training, and testing. These specifications ensure that the system performs efficiently during model training and prediction tasks.

- **Processor** : Dual Core / Intel Core 2 Duo or higher
- **RAM** : 4 GB or above
- **Hard Disk** : 250 GB minimum storage
- **System Type** : 64-bit architecture recommended

These hardware configurations support dataset processing, model training, and execution of machine learning libraries required for cyberbullying detection.

Software Requirements

Software requirements define the platform and tools necessary for implementing the proposed system. These components provide the environment for data preprocessing, model development, and evaluation.

- **Operating System** : Windows 7 / 8 / 10 or later
- **Programming Language** : Python
- **Development Environment** : Spyder / Jupyter Notebook
- **Libraries Used** : NumPy, Pandas, Scikit-learn, TensorFlow/Keras, NLTK
- **Front-End Interface** : Jupyter Notebook / Spyder IDE

The selected software stack supports efficient data manipulation, NLP preprocessing, and deep learning model implementation.

Functional Requirements

Functional requirements describe the core operations that the cyberbullying detection system must perform. These requirements define the behavior of the system when processing user inputs.

DESIGN ENGINEERING

Design engineering focuses on transforming system requirements into a structured representation that guides implementation. In software development, design acts as an intermediate stage where functional and non-functional requirements are translated into architectural and behavioral models. Unified Modeling Language (UML) diagrams are used to visually represent system components, workflows, and interactions. These diagrams help in understanding system structure, improving maintainability, and ensuring design consistency. The proposed cyberbullying detection system is modeled using various UML diagrams to illustrate data flow, module interaction, and system deployment.

UML Diagrams

Use Case Diagram

Explanation:

The use case diagram illustrates the interactions between external actors and the system. In the proposed model, the primary actor is the user who provides textual input for analysis. The system processes the input, performs preprocessing, applies the trained LSTM model, and returns the prediction result. This diagram highlights the major functionalities such as data input, preprocessing, model training, and prediction.

Class Diagram

Explanation:

The class diagram represents the structural organization of the system by showing classes, attributes, and methods. It includes classes such as Data Collection, Preprocessing, Feature Extraction, LSTM Model, and Prediction Module. Relationships among these classes demonstrate how data flows between components and how the system performs classification tasks.

Object Diagram

Explanation:

The object diagram provides a snapshot of system instances during execution. It shows how objects derived from different classes interact with each other. This diagram helps in visualizing runtime behavior, including how preprocessed text objects are passed to the model and how prediction results are generated.

State Diagram

Explanation:

The state diagram describes the various states of the system and transitions between them. The workflow typically includes states such as data input, preprocessing, feature extraction, model training, prediction, and result display. This representation helps in understanding how the system responds to events and moves from one stage to another.

Activity Diagram

Explanation:

The activity diagram illustrates the step-by-step workflow of the cyberbullying detection process. It begins with data collection, followed by preprocessing, embedding generation, model training, testing, and prediction. Decision nodes are used to represent conditional flows such as whether text contains cyberbullying content. This diagram provides an overview of system operations.

Sequence Diagram

Explanation:

The sequence diagram shows interactions between system components in a time-based order. It demonstrates how the user submits text input, the preprocessing module processes it, the LSTM model performs classification, and the result is returned to the user. The diagram highlights message flow and execution order.

Collaboration Diagram

Explanation:

The collaboration diagram represents

communication among objects in the system. It emphasizes relationships between modules such as user interface, preprocessing unit, feature extractor, and prediction engine. The diagram helps in understanding how components collaborate to complete the detection process.

Component Diagram

Explanation:

The component diagram illustrates the high-level structure of system modules and their dependencies. Major components include Data Input Module, NLP Preprocessing Module, Embedding Generator, LSTM Model, and Prediction Interface. These components interact to process textual data and produce classification output.

DEVELOPMENT TOOLS

Python

Python is a high-level, interpreted programming language widely used for software development, data analysis, and machine learning applications. It emphasizes code readability and simplicity through a clear syntax that resembles natural language. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for developing complex systems such as cyberbullying detection frameworks. Its extensive ecosystem of libraries enables efficient data preprocessing, model training, and evaluation.

History of Python

Python was created by Guido van Rossum in the late 1980s and officially released in 1991. The language was developed at the Centrum Wiskunde & Informatica in the Netherlands. Python was influenced by several programming languages such as ABC, C, C++, and Modula-3. Over time, it evolved into one of the most popular programming languages for scientific computing and artificial intelligence. Python is open-source and maintained by a global community of developers, with contributions coordinated by the Python Software Foundation.

Importance of Python

Python plays a significant role in modern software development due to its flexibility and ease of use. Some important characteristics include:

- **Interpreted Execution:** Python code is executed line by line, eliminating the need for compilation.
- **Interactive Environment:** Developers can test code snippets directly using interactive shells.
- **Object-Oriented Support:** Python allows modular programming using classes and objects.
- **Beginner-Friendly Syntax:** The simple structure makes it easy for new programmers to learn.
- **Extensive Libraries:** Python offers numerous libraries for machine learning, data science, and NLP.

These features make Python highly suitable for developing deep learning-based cyberbullying detection systems.

Libraries Used in Python

Several Python libraries are utilized in the proposed cyberbullying detection system to perform data processing, visualization, and model development:

- **NumPy:** Provides efficient multidimensional array operations and mathematical functions for numerical computations.
- **pandas:** Used for data manipulation and handling structured datasets through DataFrame objects.
- **Matplotlib:** Enables visualization of performance metrics and analysis results.
- **scikit-learn:** Provides tools for preprocessing, model evaluation, and traditional machine learning algorithms.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

SOFTWARE TESTING

Software testing is an essential phase in the software development life cycle aimed at identifying defects and ensuring that the developed system meets both functional and non-functional requirements. The main purpose of testing is to uncover errors and verify that the application behaves as expected under different conditions. It involves evaluating components, subsystems, and the complete software product to ensure reliability and accuracy. Testing also provides confidence that the software satisfies user expectations and performs efficiently without unacceptable failures. Various testing techniques are used to validate different aspects of the system, including functionality, performance, and integration.

System testing is carried out after integrating all modules into a complete application. The purpose of system testing is to validate the entire software system and confirm that it meets overall requirements. This testing examines workflows, data processing, and interactions between different components. It ensures that the integrated system behaves predictably and produces consistent results. System testing also identifies issues related to configuration and communication between subsystems, thereby verifying end-to-end functionality.

Performance testing evaluates the efficiency and responsiveness of the software system. It ensures that outputs are generated within acceptable time limits and that the application handles user requests efficiently. This testing measures response time, processing speed, and system throughput. Performance testing also assesses how the system behaves under different workloads. The objective is to confirm that the software performs reliably without delays or resource limitations.

Integration testing is conducted to verify the interaction between two or more software components. It focuses on identifying defects that occur due to communication between modules. Integration testing is performed incrementally by combining modules step by step. The goal is to ensure that data exchange, control flow, and interface communication function correctly. This testing confirms that integrated components operate smoothly without errors and that the system behaves as expected when modules work together.

Acceptance testing is the final stage of testing and involves participation from end users. This testing ensures that the system satisfies user requirements and is ready for deployment. During acceptance testing, real-world scenarios are used to validate system functionality. For data synchronization, acknowledgment is received by the sender node after packets reach the destination node, route addition is performed only when a route request is generated, and node status information is automatically updated through cache mechanisms. Acceptance testing confirms that the system operates correctly in practical conditions.

The test plan is developed by dividing the project into smaller units and defining testing strategies for each component. Unit testing helps identify defects within individual modules so that they can be corrected before integration. The test plan outlines objectives, scope, testing procedures, and evaluation criteria. A structured testing approach improves software reliability and ensures that the final system is free from major defects. Through systematic testing, the overall quality and performance of the software application are enhanced.

CONCLUSION

The proposed cyberbullying detection framework integrates Natural Language Processing (NLP) techniques with Long Short-Term Memory (LSTM) networks to effectively identify harmful online interactions. The system applies comprehensive preprocessing steps, including tokenization, stemming, lemmatization, and stop-word elimination, to transform raw textual data into structured and meaningful input. These preprocessing techniques reduce noise and enhance the quality of the dataset, allowing the model to learn relevant linguistic patterns. Furthermore, embedding-based feature extraction is employed to preserve semantic relationships among words, enabling the framework to better interpret contextual meaning and subtle variations in language.

The LSTM architecture plays a crucial role in capturing sequential dependencies within text, making it suitable for analyzing emotional tone and contextual flow in user-generated content. By modeling long-term relationships between words, the system improves its ability to detect abusive or offensive expressions that may not be evident through simple keyword-based approaches. Experimental evaluation demonstrates that the proposed model achieves high classification accuracy and outperforms several traditional machine learning algorithms. The incorporation of resampling techniques to address class imbalance further improves model fairness and reduces bias toward majority classes.

This framework contributes to creating safer digital environments by automating the detection of cyberbullying and abusive content. The model can be integrated into social media platforms, online forums, and messaging applications to support real-time moderation and content filtering. The study also highlights the effectiveness of deep learning methods in understanding human language and emotional context. Additionally, the proposed system can be extended to support multilingual datasets and cross-domain applications, increasing its adaptability across different online communities. Overall, the project demonstrates how intelligent text analytics can play a significant role in minimizing cyberbullying and promoting responsible online communication.

REFERENCES

[1] R. Gün and G. G. Akduman, "What is cyberbullying?" in *Bullying in Media and Beyond*. Turkey: IGI Global, pp. 473–485.

[2] Y. Hu, E. M. Clancy, and B. Klettke, "Understanding the vicious cycle: Relationships between nonconsensual sexting behaviours and cyberbullying perpetration," *Sexes*, vol. 4, no. 1, pp. 155–166, 2023.

[3] E. I. Galyashina and V. D. Nikishin, "The concepts of aggressive information impact through the lens of internet users' worldview security," *Journal of Siberian Federal University: Humanities & Social Sciences*, vol. 14, no. 11, pp. 1660–1673, 2021.
[4] S. Joshi, H. G. Nagariya, N. Dhanotiya, and S. Jain, "Identifying fake profile in online social network: An overview and survey," *Communications in Computer and Information Science*, vol. 1240, pp. 17–28, 2020.
[5] E. Vogels, *Teens and Cyberbullying 2022*, Pew Research Center, 2022.
[6] S. Cook, *Cyberbullying Statistics and Facts*, Comparitech, 2024.
[7] L. H. Collantes et al., "The impact of cyberbullying on mental health of victims," in *Proc. International Conference on Vocational Education and Training*, pp. 30–35, 2020.
[8] S. Unnava and S. R. Parasana, "A study of cyberbullying detection and classification techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15607–15613, 2024.
[9] R. Endsuy, "Sentiment analysis between VADER and EDA for the U.S. Presidential Election 2020," *Journal of Applied Data Sciences*, vol. 2, no. 1, pp. 8–18, 2021.
[10] L. Grunin, G. Yu, and S. S. Cohen, "Youth cyberbullying behaviors and parental emotional support," *International Journal of Bullying Prevention*, vol. 3, no. 3, pp. 227–239, 2021.
[11] I. Ali and N. Hameed, "Hybrid tools and techniques for sentiment analysis: A review," *International Journal of Multidisciplinary Sciences and Engineering*, vol. 8, no. 4, pp. 28–33, 2017.
[12] J. O. Atoum, "Cyberbullying detection through sentiment analysis," in *Proc. International Conference on Computational Science and Intelligence*, pp. 292–297, 2020.
[13] P. Yi and A. Zubiaga, "Session-based cyberbullying detection in social media," *Online Social Networks and Media*, vol. 36, 2023.
[14] T. Ahmed et al., "Transformer-based architectures for cyberbullying detection," *Social Network Analysis and Mining*, vol. 12, no. 1, 2022.
[15] UNICEF, "Children at increased risk of harm online during global COVID-19 pandemic," 2024.
[16] M. Humayun et al., "Deep learning based sentiment analysis of COVID-19 tweets," *Computer Systems Science and Engineering*, vol. 47, no. 1, pp. 575–591, 2023.
[17] M. F. Almufareh et al., "Transformer-based model for sentiment analysis," *IEEE Access*, vol. 12, pp. 196803–196817, 2024.
[18] A. Fernández et al., "SMOTE for learning from imbalanced data," *Artificial Intelligence Journal*, vol. 61, pp. 863–905, 2018.
[19] F. Wu et al., "Fusion capsule network for cyberbullying detection," *Neurocomputing*, vol. 542, 2023.
[20] M. I. Mahmud et al., "Textual features based cyberbullying detection," in *Proc. IEEE Global Conference on AI and IoT*, pp. 166–170, 2022.
[21] S. Tambe et al., "High dimensional sparse matrix representations," arXiv preprint arXiv:2202.02894, 2022.