A COMPARATIVE STUDY OF DEEP LEARNING METHODS FOR CYBERBULLYING DETECTION

Dang Thi Kim-Ngan, Nguyen Thi Thanh-Thuy, Lam Mai*

The University of Danang - Vietnam-Korea University of Information and Communication Technology, Vietnam

*Corresponding author: mlam@vku.udn.vn

(Received: May 06, 2025; Revised: June 17, 2025; Accepted: June 18, 2025)

DOI: 10.31130/ud-jst.2025.23(6A).257E

Abstract - This paper conducts a comparative study of machine learning and deep learning approaches for cyberbullying detection on social networking platforms. The evaluated models include traditional classifiers such as Logistic Regression and Support Vector Machine (SVM), as well as deep learning architectures including LSTM, BiLSTM, CNN, and a hybrid CNN-BiLSTM model. Experimental results indicate that while SVM and Logistic Regression achieve competitive performance among traditional methods, the proposed CNN-BiLSTM model consistently outperforms others by effectively capturing both local and sequential text features. These findings demonstrate the effectiveness of integrating convolutional and recurrent neural networks in improving the accuracy and robustness of automated cyberbullying detection systems.

Key words - Cyberbullying detection; social network; natural language processing; machine learning; CNN-biLSTM

1. Introduction

Social networks have become an integral part of modern life, offering diverse benefits such as communication, information sharing, and social interaction. According to Statista, over 4.89 billion people worldwide actively use social media platforms [1]. However, alongside these advantages, the issue of cyberbullying has emerged as a serious social problem. A study by Patchin and Hinduja revealed that approximately 59% of teenagers in the United States have experienced some form of cyberbullying [2]. This form of online harassment - involving the use of digital platforms to threaten, insult, or humiliate others - poses significant risks to users' mental health, especially among young people.

The increasing prevalence of cyberbullying highlights the urgent need for effective detection and prevention solutions. Traditional approaches, which largely rely on user reports and manual moderation, have proven to be time-consuming and often ineffective in identifying subtle or context-dependent abusive behavior [3]. In response, automated detection systems leveraging advances in Natural Language Processing (NLP) have demonstrated promising potential in supporting online content moderation efforts [4].

Detecting cyberbullying through automated systems is challenging due to factors like sarcasm, slang, code-switching, and the need for contextual understanding. Sarcasm and slang often blur the line between offensive and non-offensive comments. Code-switching, where multiple languages are mixed in a single post, adds complexity to text classification. Furthermore, the meaning and intent of a comment can vary depending on the context, making accurate detection even more difficult.

Figure 1. *Cyberbullying in Social Networks*

While numerous studies have proposed diverse machine learning and NLP-based methods for cyberbullying detection, there remains a lack of comprehensive, systematic comparison between these approaches. Such a comparison is essential to understand the relative strengths, weaknesses, and practical applicability of different models in real-world scenarios. In this work, we conduct a comparative analysis of traditional machine learning methods and advanced deep learning architectures, notably a hybrid CNN-BiLSTM model, to strengthen the performance and reliability of cyberbullying detection systems.

This research aims to explore the following core questions:

- In detecting cyberbullying on social media, how do conventional machine learning techniques measure up against deep learning models in terms of accuracy, computational cost, and flexibility?

- Additionally, does integrating a hybrid CNN-BiLSTM framework lead to notable enhancements compared to using standalone models, particularly in recognizing both contextual and sequential characteristics of toxic online comments?

2. Related Work

2.1. Cyberbullying Detection with Conventional Machine Learning Approach

Cyberbullying detection has attracted significant research attention, with numerous studies exploring both machine learning and deep learning methods to tackle its complexities. Initial methods predominantly utilized conventional machine learning algorithms alongside simple textual features like Bag of Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF).

BoW combined with Logistic Regression has been one of the most common baseline methods in text classification, including cyberbullying detection, due to its simplicity and surprisingly effective performance on smaller datasets [5]. Similarly, TF-IDF paired with Support Vector Machine (SVM) has demonstrated strong

performance, leveraging the ability of SVM to handle high-dimensional feature spaces effectively [6]. Random Forest classifiers using TF-IDF vectors have also been explored, benefiting from ensemble learning to improve classification stability and reduce variance [7].

Muneer and Fati [8] compared models based on training time and prediction performance, with Logistic Regression offering the best trade-off. Bozyiğit et al. [9] found AdaBoost to be the most accurate but slowest, while Random Forest achieved the highest recall and was preferred for its ability to reduce overfitting. Thun et al. [10] highlighted the importance of feature selection using the Gini index in Random Forest, which was further supported by other studies [11, 12].

2.2. Cyberbullying Detection with Deep Learning Approach

In recent years, deep learning approaches have significantly advanced the state of cyberbullying detection. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have shown promising results due to their capacity to capture sequential dependencies in text data [4], [13]. Further improvements were observed with the use of Bidirectional LSTM (BiLSTM), which allows information to be processed in both forward and backward directions, thereby better understanding the contextual relationships within comments [14].

Kumar and Sachdeva [15] introduced a hybrid deep learning framework, Bi-GRU-Attention-CapsNet (Bi-GAC), which integrates a Bi-GRU with self-attention mechanisms and a capsule network to efficiently capture both semantic and spatial characteristics in social media content for detecting cyberbullying.

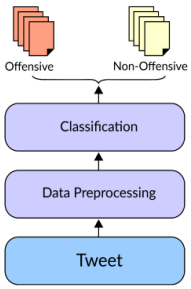

Figure 2. *A General Cyberbullying Detection System*

Convolutional Neural Networks (CNNs), traditionally popular in computer vision, have also been effectively adapted for text classification by capturing local patterns and n-gram-like features in text sequences [16], [19]. Building on these advancements, recent studies have combined CNN with BiLSTM architectures to leverage the strengths of both models-CNN for local feature extraction and BiLSTM for sequential context modeling. This hybrid CNN-BiLSTM model has been found to improve detection performance in cyberbullying classification tasks, demonstrating superior ability to capture both complex textual patterns and contextual dependencies [17]. Dadvar and Eckert [18] investigated various deep learning architectures, including CNN, LSTM, BiLSTM, and attention-enhanced BiLSTM, demonstrating that these models outperform conventional machine learning approaches when applied to the same YouTube dataset.

3. Methodology

The system's overall workflow involves several consecutive stages, such as data preprocessing, feature extraction, model training, and performance assessment. This structured pipeline allows for systematic handling of input data and consistent assessment of different models. The detailed architecture and interaction between various components within the system are illustrated in Figure 3, providing a clear visualization of how data flows through the processing and learning stages, ultimately supporting effective cyberbullying detection.

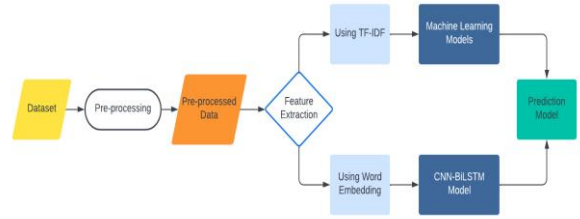

Figure 3. *Propose Model*

3.1. Dataset

In this research, we used multiple datasets for detecting cyberbullying, combining both publicly available and custom-collected data. The primary dataset is the "Cyberbullying Detection" dataset from Kaggle [20], which contains approximately 30,000 English-language comments labeled as either cyberbullying or non-cyberbullying. The data was collected from social media platforms such as Facebook, Instagram, and online forums, reflecting real-world informal language, slang, and harassment expressions.

In addition, we performed web scraping and crawling to collect supplemental publicly available text data from social media and forums. All collected samples were manually annotated by human reviewers based on predefined criteria, such as the presence of insults, threats, or targeted harassment.

The final dataset is stored in a structured format and shows a relatively balanced distribution between classes, as illustrated in Figure 4, ensuring fairness and improving model generalization. Prior to training, all text data were preprocessed through normalization, including lowercasing, stop-word removal, and punctuation filtering.

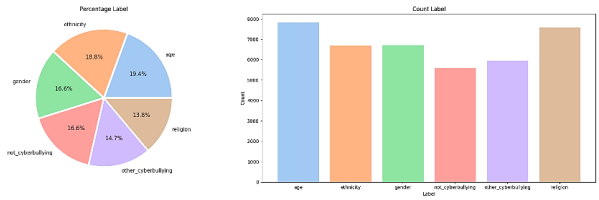

Figure 4. *Proportions of Cyberbullying Categories in the Research*

3.2. Data-preprocessing

The preprocessing procedure follows a structured, multi-step pipeline designed to clean and transform raw

social media text data into a format suitable for cyberbullying detection models. The steps are outlined as follows:

Step 1: Text Normalization

All text is converted to lowercase to reduce case sensitivity and ensure uniform representation. For example, “Hate” and “hate” are treated identically.

Step 2: Data Cleaning

Non-informative elements such as punctuation marks, special characters, and URLs are removed. This reduces noise and prevents these tokens from distorting the learning process.

Step 3: Stopword Removal

Frequent words that contribute little to the meaning (such as "the", "is", "at") are removed to reduce dimensionality, enabling the model to concentrate on more significant terms.

Step 4: Lemmatization

Words are transformed into their base or dictionary forms (lemmas). For instance, “calling” becomes “call”, and “better” becomes “good”. This process helps reduce vocabulary size and unify word variations.

Step 5: Tokenization

Text is split into individual tokens (usually words). Each token is mapped to a unique integer ID to create a numeric representation of the input.

Step 6: Padding

To ensure consistent input length across all samples, shorter sequences are padded (typically with zeros) to a fixed length required by deep learning models.

Step 7: Feature Extraction and Selection

Features like Term Frequency-Inverse Document Frequency (TF-IDF), sentiment scores, and n-grams are extracted. A feature selection technique, such as Random Forest importance ranking, is then used to keep the most relevant and distinguishing features.

3.3. Building deep learning models for fault prediction and performing comparative performance analysis

We applied different preprocessing steps according to model type. For traditional machine learning models (Logistic Regression, SVM), we normalized the text and used TF-IDF to obtain fixed-length feature vectors. For deep learning models (CNN, LSTM, BiLSTM, CNN-BiLSTM), we used pre-trained word embeddings to represent each input as a sequence of vectors. CNN processed these as 2D matrices to capture local patterns, while LSTM and BiLSTM preserved the sequential structure to model contextual dependencies. The hybrid CNN-BiLSTM combined both approaches. All input sequences were padded or truncated to a fixed length.

3.3.1. TF-IDF

TF-IDF is one of the fundamental techniques in natural language processing to evaluate the importance of a word in a document.

TF (Term Frequency): The frequency of a term in a document.

$$TF(t,d) = \frac{Number\ of\ times\ term\ t\ appears}{Total\ number\ of\ terms}$$

IDF (Inverse Document Frequency): Used to assess the importance of a term in a document. When calculating TF, the importance of terms is equal. However, some unimportant terms frequently appear, such as conjunctions, prepositions, and articles. IDF reduces the importance of such terms.

$$IDF(t,D) = log_e^{\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ term\ t}\right)}$$

TF-IDF: $TF - IDF(t,d,D) = TF(t,d) * IDF(t,D)$

Words with high TF-IDF scores are those that appear frequently in one document but less frequently in others. This helps filter out common words and retain valuable words in the document (keywords).

3.3.2. Word Embedding

This method transforms words or phrases from raw text into fixed-size and trainable numerical vectors, enabling computers to understand and process natural language more effectively. In this paper, Word Embedding is used to represent each word in the text as a multi-dimensional feature vector. Each vector not only contains information about the word’s meaning but also includes contextual information and relationships with other words in the same text. This enables the model to recognize and classify text based on the semantics and context of each word.

We used the Embedding layer from Keras to create Word Embedding vectors. This layer automatically learns embedding weights during model training, allowing the model to optimize word representation for better classification performance. As a result, the model can understand complex meanings and dependencies between words, enhancing text classification accuracy.

a. Training and Hyperparameter Tuning

Text data is preprocessed by tokenizing, removing stop words, and applying techniques as stemming or lemmatization. For deep learning models, word embeddings or one-hot encoding are used to represent text numerically.

Hyperparameter tuning is essential for optimizing model performance. For models like Logistic Regression and SVM, hyperparameters such as the regularization parameter (C), kernel type, and gamma are tuned. For deep learning models, essential parameters include learning rate, size of batch, number of training epochs and the number of hidden units in LSTM or BiLSTM layers.

To enhance model robustness, K-fold cross-validation approach is employed. The dataset is divided into K equally sized folds. This iterative validation approach provides a more reliable assessment of model performance and mitigates the probability of overfitting, thereby improving the model’s overall to unobserved data.

b. Evaluation Metrics

After training, each method's performance is evaluated using standard metrics (accuracy, precision, recall, F1-

score, and confusion matrix). Subsequently, the performance of machine learning and deep learning models is evaluated comparatively to identify the most effective approach.

Accuracy is the ratio of correctly predicted instances to the total instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP (True Positives) refers to instances where the model accurately identifies positive cases, while TN (True Negatives) denotes correctly recognized negative instances. FP (False Positives) occurs when negative cases are mistakenly classified as positive, and FN (False Negatives) represents positive cases that the model fails to detect.

Precision measures the proportion of true positive predictions out of all the predicted positives.

$$Precision = \frac{TP}{TP + FN}$$

Recall quantifies the ability of a model to correctly identify positive instances, calculated as the ratio of true positives to the total number of actual positive cases.

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

F1-Score represents the harmonic average of precision and recall, offering a single metric that balances both measures, especially useful when dealing with imbalanced classes.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

A confusion matrix offers a structured overview of classification outcomes, presenting the counts of correct and incorrect predictions across different classes in a tabular format.

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

This matrix serves as a visual tool to derive key performance metrics such as accuracy, precision, recall, and the F1-score, offering insights into the model's predictive effectiveness.

c. Model Selection

In this study, we employ a range of traditional machine learning models and deep learning models to detect cyberbullying. The selected models include BoW + Logistic Regression, TF-IDF + SVM, as well as deep learning models like CNN, LSTM, and the proposed model CNN-BiLSTM. Each of these models has unique characteristics and is used to analyze the complex relationships within the textual data. The following section provides an in-depth overview of each model under consideration.

d. BoW + Logistic Regression

BoW (Bag of Words) is a standard text feature extraction method with each word in the document is treated as a feature, represented as a column in the feature matrix, indicating the frequency of the word. Logistic Regression is a binary classification model that identifying whether a comment is toxic or non-toxic. It learns a classification function to separate the two classes. First of all, BoW features are extracted then Logistic Regression is applied directly to these features for classification.

e. TF-IDF + SVM

TF-IDF (Term Frequency-Inverse Document Frequency) helps assess the importance of words in a document by reducing the impact of frequently occurring words. TF-IDF enables the model to focus on more informative terms. While SVM is a powerful classification model that finds the optimal hyperplane to separates the two classes. The data is first processed using TF-IDF, and then fed into the SVM for classification of comments as toxic or non-toxic.

f. CNN (Convolutional Neural Network)

CNN uses convolutional layers to learn local features from the text, allowing the model to getting relationships between words within a small window of the sentence. These features are then connected through fully connected layers. CNN applies a series of convolution and pooling layers to extract strong features from the text data, followed by fully connected layers for classification.

g. LSTM

LSTM consists of input, forget, and output gates that maintain the information state across time that can learn semantic relationships between words in a sentence, making it ideal for sequential data tasks like text analysis.

h. CNN-BiLSTM

CNN-BiLSTM is combining the strengths of both CNN and BiLSTM, aiming to leverage both local features and sequential relationships in the text. Input Layer is a pre-processed text sequence, CNN is used to extract local features of the text, especially important patterns or phrases. BiLSTM is the model learns that not only from past information but also from future context, enhancing the understanding of the semantic meaning in the text.

4. Experiment result

In this paper, experiments were conducted on a labeled cyberbullying dataset collected from social networking platforms. The dataset was divided into 80:20 split ratio training and testing sets. Text preprocessing were applied to clean the raw text data. For deep learning models, text sequences were converted into word embeddings using pre-trained GloVe vectors, while traditional models used BoW and TF-IDF features for representation.

All models were implemented using Python with libraries including Scikit-learn, Keras, and TensorFlow. Experiments were executed on a workstation (Intel Core i7 processor, 32GB RAM, and an NVIDIA RTX 2060 GPU), ensuring efficient model training and evaluation.

Figure 5 shows the top 15 words in texts related to 'Ethnicity' and 'Gender' cyberbullying for analysis.

Figure 6 shows the top 15 words in texts related to 'Other Cyberbullying' and 'Not Cyberbullying' categories for comparison.

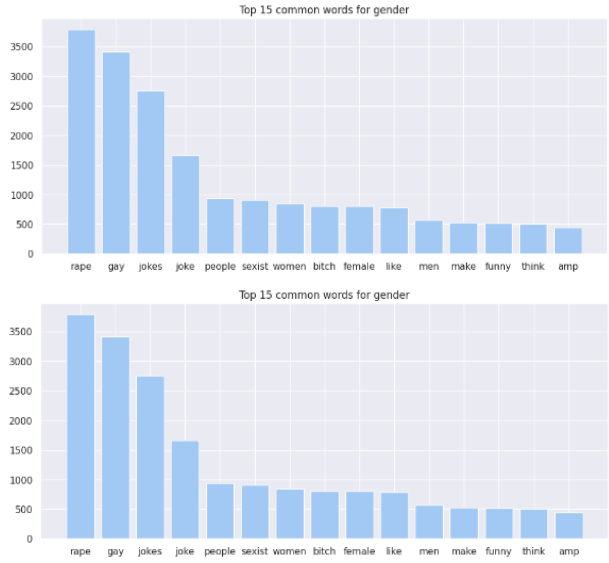

Figure 5. *Top 15 Words in Texts Related to a) 'Ethnicity' Cyberbullying; b) 'Gender' Cyberbullying*

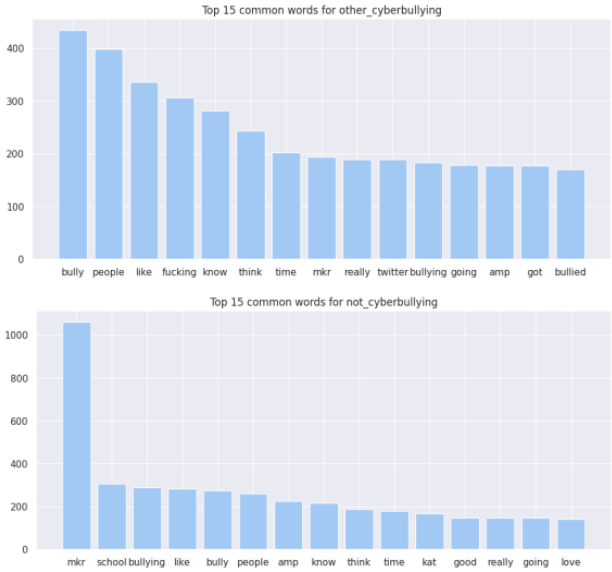

Figure 6. *Top 15 Words in Texts Related to a) 'Other Cyberbullying'; b) 'Not Cyberbullying'*

4.1. Experimental setup

After training the proposed models, the prediction matrices are as shown in Figure. Figure illustrating the prediction matrix for various machine learning models. The results show a generally high number of correct predictions across different categories of cyberbullying. However, the prediction accuracy for the 'other cyberbullying' and 'not cyberbullying' categories is comparatively lower across all models used.

4.2. Performance of Classification Models

This section reports and interprets the experimental results. Initially, Table 1 presents and compares the performance of each classifier based on important evaluation metrics such as accuracy, precision, recall, F1-score, and prediction time. Subsequently, Table 2 illustrates the training time complexity associated with each algorithm. A detailed analysis of these findings is provided in the subsequent subsections.

4.3. Performance of Classification Models

This section reports and interprets the experimental results. Initially, Table 1 presents and compares the performance of each classifier based on important evaluation metrics such as accuracy, precision, recall, F1-score, and prediction time. Subsequently, Table 2 illustrates the training time complexity associated with each algorithm. A detailed analysis of these findings is provided in the subsequent subsections.

Table 1 presents the performance evaluation of the models in cyberbullying detection. Among all the models tested, the CNN-BiLSTM hybrid model demonstrated the best overall performance across all evaluation metrics. Notably, it attained the best performance among all models, with an accuracy of 91.05%, precision of 90.50%, recall of 89.95%, and an F1-score of 90.22%. This indicates its strong ability to both identify and generalize across instances of cyberbullying texts.

While BiLSTM slightly outperformed CNN in terms of precision and recall, CNN demonstrated a shorter prediction time (2.20ms) that can be applied efficient in real-time applications.

Logistic Regression and SVM lagged behind in overall accuracy and F1-score, though they maintained very low inference times (0.35ms and 0.40ms), which might still be beneficial in resource-constrained settings. The LSTM model showed moderate performance, surpassing classical models but underperforming compared to its bidirectional counterpart and the CNN-BiLSTM model.

Table 1. *Summary of Evaluation Metrics for Models*

No.	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Prediction Time (ms)
1	Logistic Regression	85.32	84.76	83.95	84.35	0.35
2	SVM	86.78	86.12	85.30	85.70	0.40
3	LSTM	88.45	87.90	87.30	87.60	2.80
4	BiLSTM	89.30	88.65	88.10	88.37	3.10
5	CNN	88.92	88.20	87.70	87.95	2.20
6	**CNN-BiLSTM**	**91.05**	**90.50**	**89.95**	**90.22**	**3.50**

The BiLSTM and CNN models followed closely, with accuracy scores of 89.30% and 88.92%, respectively.

In terms of prediction time, while deep learning models generally require more computation than classical methods, the slight increase in time (e.g., 3.50ms for CNN-BiLSTM) is justifiable given the significant improvement in classification performance.

Overall, the findings indicate that the combination of convolutional layers and bidirectional recurrent architectures enhances the model's ability to extract local patterns while simultaneously modeling long-range dependencies, thereby improving the accuracy and robustness of cyberbullying content detection.

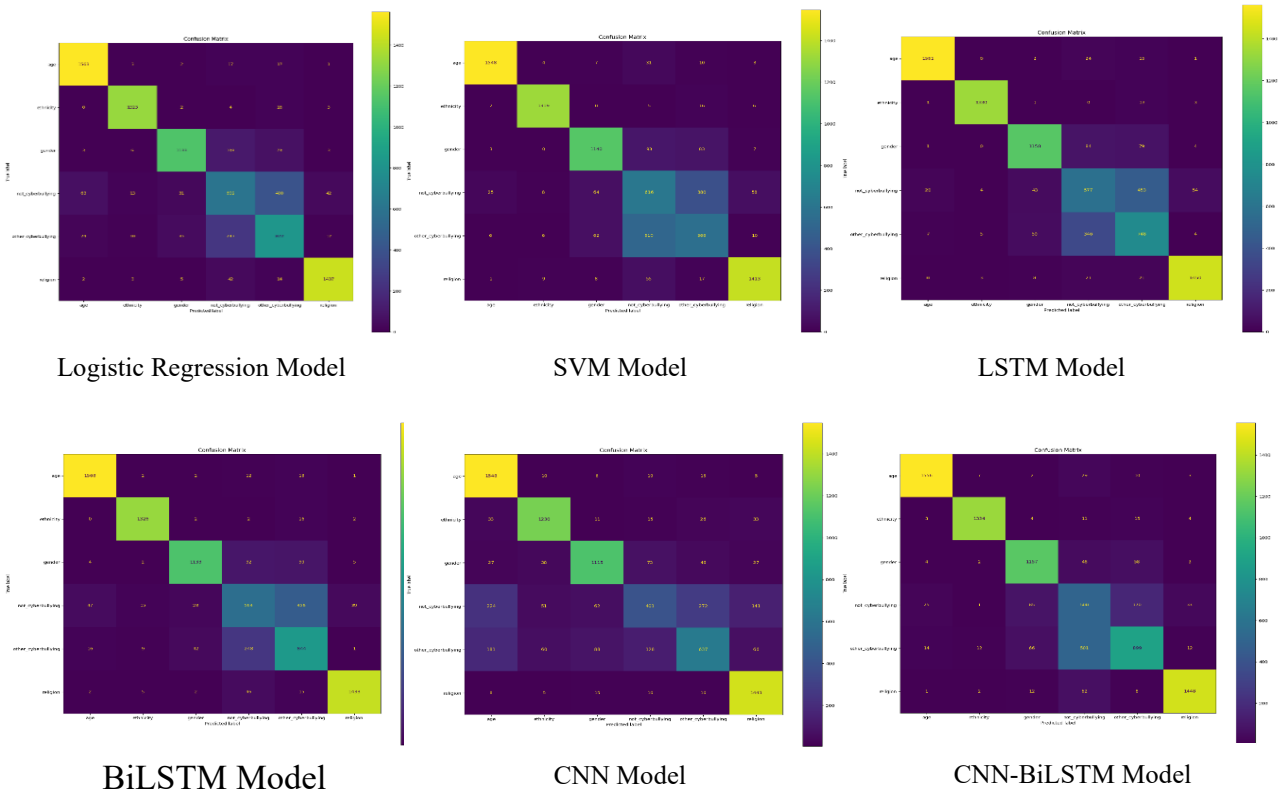

Figure 7. *Confusion Matrix for Model Predictions*

Figure 8. *Performance summary of classification models*

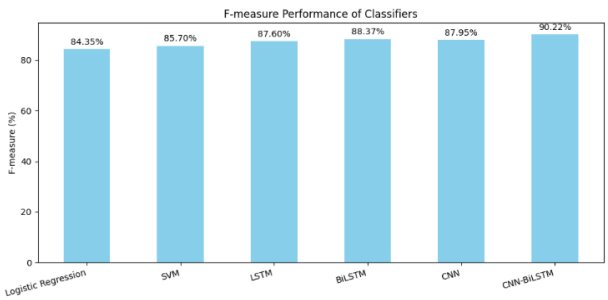

Figure 9. *F-measure performance of classification models*

4.4. Time Complexity of Models

Table 2 summarizes the time complexity of each model, considering both training and inference durations. As shown, Logistic Regression demonstrates the fastest training time at 12.5 seconds, whereas the CNN-BiLSTM model exhibits the highest training time, reaching 185.9 seconds.

In terms of inference time per sample, Logistic Regression again outperforms all other models, with the fastest inference time of 0.35 milliseconds. But then, the CNN-BiLSTM exhibits the slowest inference time at 3.50 milliseconds.

These results highlight that while some models, such as Logistic Regression, provide quicker performance, more complex models like CNN-BiLSTM tend to require longer processing times for both training and inference.

Table 2. *Time Complexity of Models.*

No.	Model	Training Time (seconds)	Inference Time per Sample (milliseconds)
1	**Logistic Regression**	**12.5**	**0.35**
2	SVM	18.7	0.40
3	LSTM	145.3	2.80
4	BiLSTM	162.4	3.10
5	CNN	130.6	2.20
6	CNN-BiLSTM	185.9	3.50

5. Conclusion

This paper presents a comprehensive comparative study of cyberbullying detection approaches on social networking. Through experimental evaluations, traditional models such as Logistic Regression and SVM demonstrated strong baseline performance, with SVM slightly outperforming Logistic Regression in accuracy and F1-score rates.

Among deep learning models, the proposed CNN-BiLSTM architecture achieved the highest performance by effectively capturing both local contextual features and

sequential dependencies in text data. While LSTM and BiLSTM also model semantic information through sequential context, they may overlook critical short-range patterns such as abusive phrases or slang. The convolutional component of CNN-BiLSTM addresses this by extracting salient local features (e.g., n-grams), which complement the semantic modeling of BiLSTM, resulting in richer feature representation and higher classification accuracy.

Additionally, while the CNN-BiLSTM model required higher training and inference times compared to traditional approaches, the significant improvement in detection accuracy and reliability highlights its potential applicability for real-world deployment in social media content moderation systems. In the next future, we may focus on expanding the dataset, integrating multilingual support, and exploring transformer-based models such as BERT to further enhance detection capabilities.

Acknowledgment: This research is funded by the Vietnam-Korea University of Information and Communication Technology under project number ĐHVH-2025-02.

REFERENCES

[1] B. Dudić, A. Mittelman, and J. Vojtechovský, "Social Media and Marketing Worldwide", in *Proc. Int. Conf. New Technologies, Development and Applications*, Cham, Switzerland, 2024, pp. 271-278, Springer Nature.

[2] J. W. Patchin, and S. Hinduja, "Measuring Cyberbullying: Implications for Research", *Aggression and Violent Behavior*, vol. 23, pp. 69-74, 2015.

[3] J. M. Nagata *et al.*, "Adverse childhood experiences and early adolescent cyberbullying in the United States", *Journal of Adolescence*, vol. 95, no. 3, pp. 609–616, 2023.

[4] T. Mahmud, M. Ptaszynski, J. Eronen, and F. Masui, "Cyberbullying detection for low-resource languages and dialects: Review of the state of the art", *Inf. Process. Manage.*, vol. 60, no. 5, pp. 103454, 2023.

[5] P. Pranathi, V. Revathi, P. Varshitha, S. Shaik, and S. Bhutada, "Logistic regression based cyber harassment identification", *J. Adv. Math. Comput. Sci.*, vol. 38, no. 8, pp. 76-85, 2023.

[6] I. P. Sari and H. Maulana, "Detecting Cyberbullying on Social Media Using Support Vector Machine: A Case Study on Twitter", *Int. J. Saf. Sec. Eng.*, vol. 13, no. 4, 2023.

[7] M. P. Rao, N. Kota, D. Nidumukkala, M. Madoori, and D. Ali, "Enhancing Online Safety: Cyberbullying Detection with Random Forest Classification", in *Proc. 2024 10th Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2024, pp. 389-393, IEEE.

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

[9] A. Bozyiğit, S. Utku and E. Nasibov, "Cyberbullying detection: Utilizing social media features", *Expert Syst. Appl.*, vol. 179, Oct. 2021.

[10] L. J. Thun, P. L. Teh and C.-B. Cheng, "CyberAid: Are your children safe from cyberbullying?", *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4099-4108, Jul. 2022.

[11] D. Chatzakou, I. Leontiadis, J. Blackburn, E. De Cristofaro, G. Stringhini, A. Vakali, and N. Kourtellis, "Detecting cyberbullying and cyberaggression in social media", *ACM Trans. Web*, vol. 13, no. 3, pp. 1–51, 2019.

[12] V. Balakrishnan, S. Khan, T. Fernandez and H. R. Arabnia, "Cyberbullying detection on Twitter using big five and dark triad features", *Personality Individual Differences*, vol. 141, pp. 252-257, Apr. 2019.

[13] C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, "Cyberbullying detection solutions based on deep learning architectures", *Multimedia Syst.*, vol. 29, no. 3, pp. 1839-1852, 2023.

[14] K. N. Devi, V. Rajasekar, P. Jayanthi, K. Balasubramani, K. Kandasamy, and K. Gowrisankar, "Cyberbullying Detection and Severity Classification Using Bi-LSTM", in *Proc. 2024 9th Int. Conf. Commun. Electronics Syst. (ICCES)*, Dec. 2024, pp. 338-344, IEEE.

[15] A. Kumar and N. Sachdeva, "A bi-GRU with attention and CapsNet hybrid model for cyberbullying detection on social media", *World Wide Web*, vol. 25, no. 4, pp. 1537-1550, Jul. 2022.

[16] A. Kumar and N. Sachdeva, "Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network", *Multimedia Syst.*, vol. 28, no. 6, pp. 2043-2052, 2022.

[17] P. G. Balaji, P. P. Katariya, S. Sruthi, and M. Venugopalan, "Cyberbullying Detection on Multiclass Data Using Machine Learning and A Hybrid CNN-BiLSTM Architecture", in *Proc. 2024 Int. Conf. Knowledge Eng. Commun. Syst. (ICKECS)*, vol. 1, pp. 1-6, Apr. 2024, IEEE.

[18] M. Dadvar and K. Eckert, "Cyberbullying detection in social networks using deep learning based models; a reproducibility study", *arXiv:1812.08046*, 2018.

[19] L. Mai, X. Z. Chen, and Y. L. Chen, "Multi-oriented license plate detection based on convolutional neural networks", in *Proc. 2021 Int. Conf. Syst. Sci. Eng. (ICSSE)*, Aug. 2021, pp. 101-104, IEEE.

[20] Kaggle, "Cyberbullying Classification Dataset", *kaggle.com*, [Online]. Available: https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification. [Accessed: Apr. 22, 2025].