Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 4 Iss. 4 October 2024, pp: 1533-1538
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Spam Detection in YouTube Comments Using Deep Learning Models: A Comparative Study of MLP, CNN, LSTM, BiLSTM, GRU, and Attention Mechanisms

Gregorius Airlangga

Information System Study Program, Atma Jaya Catholic University of Indonesia, Indonesia

E-Mail: gregorius.airlangga@atmajaya.ac.id

Received Aug 18th 2024; Revised Sept 30th 2024; Accepted Oct 6th 2024
Corresponding Author: Gregorius Airlangga

Abstract

This study explores the effectiveness of various deep learning models for detecting spam in YouTube comments. Six models were evaluated: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Attention mechanisms. The dataset consists of 1,956 real comments extracted from popular YouTube videos, representing both spam and legitimate messages. The preprocessing phase involved tokenization and padding of text sequences to prepare them for model input. Results reveal that the LSTM model achieved the highest test accuracy of 95.65%, outperforming other models by capturing sequential dependencies and context within comments. The CNN model also demonstrated high accuracy, underscoring the importance of local pattern recognition in text classification. While BiLSTM and Attention models offered comparable performance, their marginal improvement over LSTM indicates that sequential modeling plays a crucial role in this task. The GRU model, despite being computationally efficient, showed slightly lower accuracy compared to LSTM and BiLSTM. The MLP model, serving as a baseline, exhibited limited performance, emphasizing the need for advanced architectures in spam detection. These findings suggest that combining sequential modeling with local feature extraction could lead to more robust spam detection systems. Future research may focus on hybrid models and ensemble methods to further enhance spam detection capabilities in user-generated content on social media platforms.

Keyword: Deep Learning, LSTM, Spam Detection, Text Classification, YouTube Comments

1. INTRODUCTION

The increasing volume of user-generated content on platforms like YouTube has led to a corresponding rise in spam, which threatens the quality of online interactions and the integrity of information shared [1]–[3]. YouTube, as one of the largest video-sharing platforms, hosts millions of comments daily, providing a rich medium for genuine interaction as well as an attractive target for spammers [4]–[6]. These spam comments range from promotional messages to harmful content, often misleading viewers or attempting to redirect them to malicious websites [7]. Given the scale of YouTube's content, manually filtering out spam is infeasible, underscoring the need for automated methods to detect and manage spam comments efficiently [8]. Traditional approaches to spam detection have primarily focused on rule-based systems and simple machine learning models [9]. Rule-based systems, while straightforward, rely on predefined patterns and keywords to identify spam [10]. They suffer from rigidity and lack adaptability to the evolving tactics of spammers. Machine learning techniques, particularly those based on statistical models like Naive Bayes and Support Vector Machines (SVM), have offered more flexibility and improved performance [11]. However, these models still face challenges in handling the nuanced and often sophisticated nature of spam comments, especially when contextual understanding is required [12]. The shift towards more complex machine learning techniques, including deep learning, represents a significant advancement in spam detection.

Recent literature has explored the use of deep learning models, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), for text classification tasks, including spam detection [13]. CNNs, originally designed for image processing, have been adapted to text classification due to their ability to capture local features in the data [14]. They excel at identifying n-gram patterns and are useful for recognizing commonly used spam phrases. RNNs, particularly Long Short-Term Memory (LSTM) networks, are adept at modeling sequential data, making them suitable for understanding the context within a comment [15]. The combination of these models with techniques like attention mechanisms has shown promise in enhancing model performance by focusing on the most relevant parts of the input data [16]. Despite these advancements,

the challenge remains to develop a model that balances accuracy, generalizability, and computational efficiency [17]. Current state-of-the-art models often require extensive computational resources and large datasets to achieve optimal performance, limiting their practical application. Furthermore, while existing models have demonstrated high accuracy in spam detection, they may still struggle with edge cases such as ambiguous comments or those employing novel spam tactics [18]. This necessitates ongoing research to refine and enhance spam detection models, ensuring they remain effective in dynamic and diverse online environments like YouTube [19].

The urgency of addressing spam on YouTube is underscored by its potential impact on user experience and platform credibility [20]. Spam comments can distort genuine discussions, mislead viewers, and, in some cases, pose security risks through phishing or malware links [21]. For content creators and platform administrators, spam not only affects viewer engagement but can also result in a loss of trust in the platform [1]. Given the critical role that social media and content-sharing platforms play in information dissemination, effective spam detection mechanisms are essential to maintaining a safe and trustworthy digital environment [22]. A significant gap in current research lies in the need for more comprehensive evaluations of deep learning models on smaller, real-world datasets. While previous studies have demonstrated the effectiveness of deep learning in spam detection, they often rely on large-scale, synthetic, or highly curated datasets [23]. This research contributes to the field by utilizing a dataset composed of real YouTube comments, which presents the challenges of noisy, varied, and unbalanced data. By examining how different deep learning architectures perform on this dataset, the study aims to provide insights into the practical application of these models in real-world scenarios. Additionally, this research will focus on evaluating the models not only based on accuracy but also on precision, recall, and F1 score, providing a more nuanced understanding of their effectiveness in detecting spam.

This study aims to develop a comprehensive spam detection system for YouTube comments using an ensemble of deep learning models, including Multilayer Perceptrons (MLP), CNNs, LSTMs, Bidirectional LSTMs (BiLSTM), Gated Recurrent Units (GRU), and Attention-based mechanisms. By leveraging the strengths of these diverse models, the research seeks to identify a robust approach to classify comments accurately as spam or not. The ensemble approach is expected to outperform individual models by capturing various linguistic and contextual features present in the comments. Furthermore, the study will explore the effectiveness of using a relatively small dataset of 1,956 real messages, addressing the common limitation of requiring extensive labeled data for training deep learning models. The contribution of this research is threefold. First, it provides a comparative analysis of various deep learning models for the task of spam detection, highlighting their strengths and limitations in handling user-generated content on social media platforms. Second, the study introduces an ensemble approach that combines different architectures to enhance spam detection performance, offering a potential solution that can adapt to the evolving nature of spam. Third, it addresses the challenge of working with a relatively small, real-world dataset, providing insights into the models' robustness and generalizability. These contributions not only advance the current understanding of spam detection in the context of social media but also offer practical implications for the deployment of spam detection systems on platforms like YouTube.

The remaining structure of this journal article is organized as follows. The Literature Survey section reviews existing methods and models used for spam detection, emphasizing the transition from traditional machine learning approaches to deep learning techniques. This section also discusses the challenges associated with spam detection in social media environments. The Methodology section details the dataset preparation, model architectures, and evaluation metrics employed in this study. It provides a comprehensive explanation of how the various deep learning models were implemented and combined into an ensemble. The Results and Discussion section presents the experimental findings, comparing the performance of individual models and the proposed ensemble approach. This section also discusses the implications of the results, addressing the practical considerations for implementing spam detection systems on platforms like YouTube. The Conclusion section summarizes the key findings, outlines the limitations of the current study, and proposes directions for future research, including the potential for extending the models to other social media platforms and languages.

2. MATERIALS AND METHOD

This study employs a multi-stage approach to develop and evaluate deep learning models for classifying YouTube comments as spam or not spam. The methodology encompasses data collection and preprocessing, model development, model training, and evaluation, followed by performance analysis as presented in the figure 1. Each stage is elaborated with a focus on mathematical rigor and technical specificity to ensure the clarity and reproducibility of the study. The dataset consists of 1,956 real comments extracted from five YouTube videos, representing a mixture of spam and legitimate messages. It contains several features, including *comment_id*, *author*, *date*, *content*, *video_name*, and *class*. The *content* field serves as the primary input, containing the text of each comment, while the *class* field indicates whether a comment is spam (1) or not spam (0). Given the natural imbalance in the dataset, where most comments are legitimate, special

consideration was given to ensure that the preprocessing and model training steps accounted for this discrepancy to avoid model bias.

In the preprocessing phase, text normalization was the first step to prepare the comment content for model ingestion. Each comment was converted to lowercase to eliminate the variability introduced by case sensitivity. Punctuation marks and special characters, often irrelevant to the semantic understanding of the text, were removed. This process aimed to distill the comments down to their core informative components. Subsequently, tokenization was performed to split each comment into individual words or tokens, which were then mapped to unique integer indices. This mapping facilitated the conversion of text into numerical representations that the models could process. Padding was applied to the tokenized sequences to standardize their length, a crucial step when dealing with neural network models that require inputs of uniform dimensions. The length (n) was determined based on the distribution of comment lengths in the dataset, selecting a maximum sequence length that captures the essential information without excessive truncation or unnecessary padding. Formally, let (X_{tokens}) represent the tokenized sequence of a comment, the padding operation can be defined as $X_{\text{padded}} = \text{pad}(X_{\text{tokens}}, n)$, ensuring each sequence (X_{padded}) has a fixed dimensionality of (n).

The next phase involved the development of six deep learning models, each designed to capture different aspects of the textual data. The Multilayer Perceptron (MLP) model, which serves as a baseline in neural network-based classification tasks, consists of an input layer, multiple hidden layers with ReLU activation functions, and an output layer that employs a softmax activation function to produce a probability distribution over the two classes. Given the input (X), the output of the MLP is given by $y = \text{softmax}(W_h \cdot \sigma(W_{h-1} \cdot ... \sigma(W_1 \cdot X + b_1) + b_{h-1}) + b_h)$, where ($W_i$) and \(b_i \) represent the weights and biases of the \(i \)-th layer, and \(\sigma \) denotes the non-linear ReLU activation function. The Convolutional Neural Network (CNN) was adapted for text classification by leveraging its ability to capture local feature patterns within the comments. The model structure includes an embedding layer that transforms the padded sequences into dense vector representations, followed by convolutional layers equipped with multiple filters of varying kernel sizes to detect n-gram patterns. The convolution operation, central to this model, is expressed mathematically as $f_{ij} = \sigma\left(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X_{i+m,j+n} \cdot K_{m,n} + b\right)$, where (X) represents the input, (K) the convolution kernel, (b) the bias, and (σ) the activation function. The CNN architecture also incorporates max-pooling layers to down-sample the feature maps, followed by a fully connected layer that integrates the extracted features for final classification.

Long Short-Term Memory (LSTM) networks were included to model the sequential dependencies in the comments, crucial for understanding context in natural language. The LSTM architecture consists of an embedding layer followed by LSTM units that process the input sequence one element at a time, maintaining a hidden state and a cell state across the sequence. The computations within an LSTM cell involve three gates: the forget gate, the input gate, and the output gate. At each time step (t), the LSTM updates its states using the following equations $f\_t = \sigma(W\_f \cdot [h_{t-1}, X\_t] + b\_f), i\_t = \sigma(W\_i \cdot [h_{t-1}, \text{X\_t}] + \text{b\_i})$, $\tilde{C}\_t = \tanh(W\_C \cdot [h_{t-1}, \text{X\_t}] + \text{b\_C})$, $C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}$, $o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$, $h_t = o_t \cdot \tanh(C_t)$, where ($f_t$), ($i_t$), ($\widetilde{C_t}$), ($C_t$), ($o_t$), and ($h_t$) represent the forget gate, input gate, cell state update, final cell state, output gate, and hidden state, respectively. To capture contextual information from both past and future elements within a comment, the Bidirectional LSTM (BiLSTM) model processes the input sequence in both forward and backward directions. The output at each time step is the concatenation of the hidden states from both directions, given by $h_t = \left[h_t^{\text{forward}}, h_t^{\text{backward}}\right]$. This bidirectional approach provides a more comprehensive understanding of the comment's content by incorporating information from both the preceding and succeeding contexts.

The Gated Recurrent Unit (GRU) model, a simplified variant of the LSTM, was also implemented to capture sequential dependencies with fewer computational resources. The GRU cell combines the input and forget gates into an update gate, reducing the number of parameters. The GRU's computations involve the update gate and reset gate as follows $z_t = \sigma(W_z \cdot [h_{t-1}, X_t] + b_z)$, $r_t = \sigma(W_r \cdot [h_{t-1}, X_t] + b_r)$, $\widetilde{h_t} = \tanh(W \cdot [r_t \cdot h_{t-1}, X_t] + b)$, $h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \widetilde{h_t}$, where ($z_t$), ($r_t$), and ($\widetilde{h_t}$) represent the update gate, reset gate, and candidate hidden state.

To further enhance the models' ability to focus on the most informative parts of the input, an Attention-based model was introduced. This model applies an attention mechanism that assigns different weights to each element of the input sequence, allowing the model to prioritize the most relevant words for classification. The attention mechanism calculates a context vector as a weighted sum of the hidden states $\alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^{T} \exp(e_j)}$ and $c_t = \sum_{t=1}^{T} \alpha_t \cdot h_t$, where ($\alpha_t$) is the attention weight for the (t)-th element, and \(c_t \) is the context vector. The training process involved feeding the preprocessed sequences into each model, using categorical cross-entropy as the loss function and the Adam optimizer to adjust the model parameters. Early stopping was employed to monitor the validation loss and prevent overfitting, restoring the model weights corresponding to the best performance on the validation set. During training, a custom callback was utilized to compute the validation precision, recall, and F1 score at the end of each epoch, offering a comprehensive evaluation of the

models' performance beyond mere accuracy. These metrics were defined mathematically as follows $\text{Precision} = \frac{TP}{TP+FP}$, $\text{Recall} = \frac{TP}{TP+FN}$, $\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$, where (TP), (FP), and (FN) denote true positives, false positives, and false negatives, respectively. Each model's final performance was evaluated on the test set, measuring accuracy, precision, recall, and F1 score to ensure a robust assessment of the spam detection capabilities. The results were systematically compiled into a DataFrame to facilitate comparison among the models, with a particular focus on identifying the strengths and weaknesses of each approach. This comprehensive analysis aimed to determine the most effective model or combination of models for the task, taking into consideration the complexity, computational requirements, and classification performance. The insights derived from this analysis offer valuable guidance for the practical deployment of spam detection systems in dynamic and large-scale environments such as YouTube.

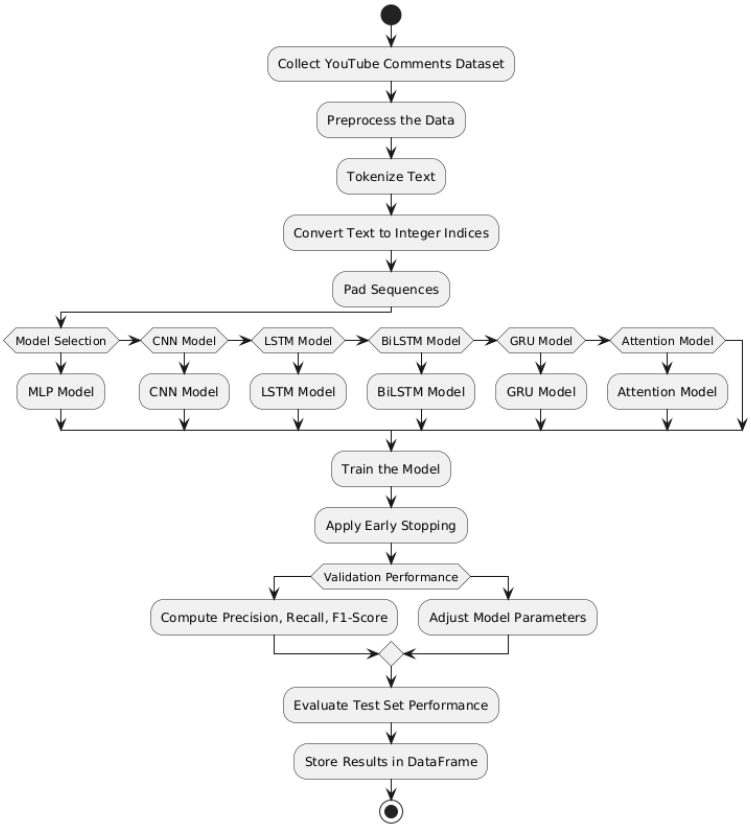

Figure 1. Research Methodology

3. RESULTS AND DISCUSSION

The results of the deep learning models employed for spam classification are presented in the table, showcasing the performance metrics: test accuracy, validation precision, validation recall, and validation F1 score. These metrics provide a comprehensive evaluation of each model's ability to classify YouTube comments as spam or not spam, highlighting their strengths and limitations. The Multilayer Perceptron (MLP) model serves as the baseline in this study, achieving a test accuracy of 54.99%. The model's validation precision and recall are 52.53% and 54.99%, respectively, with an F1 score of 50.86%. These results indicate that the MLP struggled to learn complex patterns in the text data, resulting in relatively low performance across all metrics. The underwhelming performance of the MLP can be attributed to its relatively shallow architecture, which may not effectively capture the intricate relationships and sequential dependencies present in the comments.

In contrast, the Convolutional Neural Network (CNN) model demonstrates a significant improvement, achieving a test accuracy of 94.37%. The CNN outperforms the MLP by a substantial margin, with validation precision, recall, and F1 score all around 94.88%. This improvement can be ascribed to the CNN's ability to extract local patterns from the input text, such as n-grams and key phrases, which are crucial for distinguishing between spam and non-spam comments. The use of multiple filters and kernel sizes in the CNN architecture allows it to identify various textual features, contributing to its enhanced performance. However, despite its strong results, the CNN might still struggle with longer dependencies and contextual understanding, as it primarily focuses on localized features. The Long Short-Term Memory (LSTM) model achieves the highest test accuracy at 95.65%, with validation precision, recall, and F1 score close to 94.98%. The LSTM's superior

performance underscores its capability to capture sequential dependencies within the text, which is essential for understanding the context of comments. By maintaining long-term dependencies, the LSTM can effectively process comments that exhibit complex linguistic structures or require contextual comprehension. The marginal gain in accuracy over the CNN indicates that sequential modeling plays a vital role in this classification task, especially when dealing with longer comments or those with nuanced language.

The Bidirectional LSTM (BiLSTM) model also performs well, with a test accuracy of 95.14%. Its validation precision, recall, and F1 score are all around 94.92%. The BiLSTM's performance is comparable to that of the LSTM, suggesting that capturing information in both forward and backward directions marginally enhances the model's understanding of the text. This improvement can be particularly beneficial for comments where the meaning is influenced by both preceding and succeeding words. However, the slight difference in performance between LSTM and BiLSTM implies that, in this context, the added complexity of bidirectional processing offers limited additional benefit. The Gated Recurrent Unit (GRU) model achieves a test accuracy of 91.30%, with validation precision, recall, and F1 score around 93.35%. While the GRU performs well, it falls slightly behind the LSTM and BiLSTM models. The GRU, designed to be a more computationally efficient alternative to the LSTM, effectively captures sequential dependencies but with fewer parameters. The slight decrease in performance compared to the LSTM suggests that the task of spam classification for YouTube comments may benefit more from the nuanced control over memory provided by the LSTM architecture.

The Attention-based model achieves a test accuracy of 95.14%, with validation precision, recall, and F1 score all at 94.37%. This model incorporates an attention mechanism that focuses on the most relevant parts of the input sequence, enhancing the model's ability to classify comments correctly. The results indicate that the attention mechanism is beneficial for this task, likely by enabling the model to prioritize certain words or phrases that are more indicative of spam. However, the test accuracy is slightly lower than that of the LSTM, suggesting that while attention improves interpretability, its impact on performance may not surpass that of sequential modeling in this context. In summary, the LSTM model demonstrates the best overall performance, achieving the highest test accuracy, followed closely by the CNN, BiLSTM, and Attention-based models. These results highlight the importance of sequential modeling in the task of spam classification, particularly for capturing context and long-term dependencies within comments. The CNN's high performance further underscores the significance of local pattern recognition, suggesting that combining both local and sequential feature extraction could be a promising direction for future work. The GRU and MLP models, while effective, fall short in comparison, indicating that simpler architectures or those with reduced complexity may not be sufficient for capturing the intricacies of spam comments on YouTube.

4. CONCLUSION

This study investigated the performance of four deep learning models—Deep Multilayer Perceptron (MLP), Deep Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and Long Short-Term Memory with Attention (LSTM with Attention)—in predicting student test scores based on demographic and educational factors. The models were evaluated using multiple metrics, including MAE, RMSE, R^2, MSLE, and MAPE, to provide a comprehensive view of their predictive capabilities. The results indicate that the Deep CNN model consistently outperformed the other models, achieving the lowest MAE and RMSE and the highest R^2, suggesting that its convolutional architecture was effective in capturing relationships within the dataset. The Deep MLP also performed well, though slightly less accurate than the CNN model. The Bidirectional LSTM model, which aimed to capture temporal dependencies, did not show significant improvements and lagged behind the MLP and CNN in accuracy. Finally, the LSTM with Attention model, which was expected to improve performance through its attention mechanism, performed poorly, indicating potential overfitting and an inability to handle the complexity of the task.

One notable observation across all models was the abnormally high MAPE values, which indicate that this metric may not be suitable for datasets where small true values exist. These high percentage errors suggest that alternative metrics such as sMAPE or RMSLE may be more appropriate for future studies dealing with similar datasets. In conclusion, this research underscores the importance of selecting the right model architecture for predictive tasks. Convolutional layers in deep learning models, as seen in the CNN, can capture complex patterns and yield better generalization. However, not all advanced models, such as LSTM with Attention, necessarily improve performance, especially when the task does not heavily depend on sequential or temporal relationships. Future research could explore additional architectures, fine-tune hyperparameters, and assess the robustness of alternative evaluation metrics to improve predictive performance and reduce error sensitivity.

REFERENCES

[1] R. Gorwa, R. Binns, and C. Katzenbach, “Algorithmic content moderation: Technical and political challenges in the automation of platform governance,” *Big Data \& Soc.*, vol. 7, no. 1, p. 2053951719897945, 2020.

[2] G. Jethava and U. P. Rao, “Exploring security and trust mechanisms in online social networks: An extensive review,” *Comput. \& Secur.*, p. 103790, 2024.
[3] H. Jahankhani, S. Kendzierskyj, R. Montasari, and N. Chelvachandran, *Social Media Analytics, Strategies and Governance*. CRC Press, Taylor and Francis Group, 2022.
[4] S. Bayrakdar, I. Yucedag, M. Simsek, and I. A. Dogru, “Semantic analysis on social networks: A survey,” *Int. J. Commun. Syst.*, vol. 33, no. 11, p. e4424, 2020.
[5] A. Puthussery, “Digital marketing: an overview,” 2020.
[6] S. Krüger, *Formative Media: Psychoanalysis and Digital Media Platforms*. Taylor \& Francis, 2024.
[7] S. Rao, A. K. Verma, and T. Bhatia, “A review on social spam detection: Challenges, open issues, and future directions,” *Expert Syst. Appl.*, vol. 186, p. 115742, 2021.
[8] K. Zarei, “Fake identity \& fake activity detection in online social networks based on transfer learning,” Institut Polytechnique de Paris, 2022.
[9] A. Makkar and N. Kumar, “An efficient deep learning-based scheme for web spam detection in IoT environment,” *Futur. Gener. Comput. Syst.*, vol. 108, pp. 467–487, 2020.
[10] J. Gui, Y. Zhou, K. Yu, and X. Wu, “PSC-BERT: A spam identification and classification algorithm via prompt learning and spell check,” *Knowledge-Based Syst.*, vol. 301, p. 112266, 2024.
[11] G. Teles, J. J. P. C. Rodrigues, R. A. L. Rabelo, and S. A. Kozlov, “Comparative study of support vector machines and random forests machine learning algorithms on credit operation,” *Softw. Pract. Exp.*, vol. 51, no. 12, pp. 2492–2500, 2021.
[12] A. K. Mehta and S. Kumar, “Comparative Analysis and Optimization of Spam Filtration Techniques Using Natural Language Processing,” in *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, 2024, pp. 1005–1010.
[13] A. Neisari, L. Rueda, and S. Saad, “Spam review detection using self-organizing maps and convolutional neural networks,” *Comput. \& Secur.*, vol. 106, p. 102274, 2021.
[14] M. Umer *et al.*, “Impact of convolutional neural network and FastText embedding on text classification,” *Multimed. Tools Appl.*, vol. 82, no. 4, pp. 5569–5585, 2023.
[15] J. P. Tan, A. L. A. Ramos, M. V Abante, R. L. Tadeo, and R. R. Lansigan, “A performance review of recurrent neural networks long short-term memory (LSTM),” in *2022 3rd International Conference for Emerging Technology (INCET)*, 2022, pp. 1–5.
[16] K. Cheng, Y. Yue, and Z. Song, “Sentiment classification based on part-of-speech and self-attention mechanism,” *IEEE Access*, vol. 8, pp. 16387–16396, 2020.
[17] W. Liang *et al.*, “Advances, challenges and opportunities in creating data for trustworthy AI,” *Nat. Mach. Intell.*, vol. 4, no. 8, pp. 669–677, 2022.
[18] D. Antonakaki, P. Fragopoulou, and S. Ioannidis, “A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks,” *Expert Syst. Appl.*, vol. 164, p. 114006, 2021.
[19] V. Sharma *et al.*, “FL-XGBTC: federated learning inspired with XG-boost tuned classifier for YouTube spam content detection,” *Int. J. Syst. Assur. Eng. Manag.*, pp. 1–24, 2024.
[20] K. Thomas, P. G. Kelley, S. Consolvo, P. Samermit, and E. Bursztein, “‘It’s common and a part of being a content creator’: Understanding How Creators Experience and Cope with Hate and Harassment Online,” in *Proceedings of the 2022 CHI conference on human factors in computing systems*, 2022, pp. 1–15.
[21] T. Rains, *Cybersecurity Threats, Malware Trends, and Strategies: Learn to mitigate exploits, malware, phishing, and other social engineering attacks*. Packt Publishing Ltd, 2020.
[22] F. Al-Turjman and R. Salama, “Security in social networks,” *Secur. IoT Soc. Networks*, 2020.
[23] A. S. Alhassun and M. A. Rassam, “A combined text-based and metadata-based deep-learning framework for the detection of spam accounts on the social media platform twitter,” *Processes*, vol. 10, no. 3, p. 439, 2022.