International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Enhanced Security and Reliability in Messaging Systems through Real-Time SMS Spam Filtering with Machine Learning

[1]Surabatthini Mounika ,[2]Uppala Ravi teja ,[3]Bala vijetha Reddy Badveli,[4]Sadhasivam Vamsy , [5]Amanpreet singh

Author Affiliations
[1, 2, 3, 4] School of Computer Application, Lovely Professional University, Jalandhar, Punjab, India
[5]Assistant Professor, School of Computer Application, Lovely Professional University, Jalandhar, Punjab, India
Author Emails
[1] Corresponding author: surabatthini99999gmail.com
[2] ravitejauppala77@gmail.com
[3] vijethabadveli03@gmail.com
[4] vamsysadhasivam@gmail.com
[5] apsj24@gmail.com

***Abstract*—** SMS spam has emerged as a major issue for cellular users, which leads to annoyance and inconvenience. Machine learning has been effective in filtering out spam SMS. Yet, applying these techniques in actual real-time situations poses special challenges. A newly published study seeks to tackle these challenges by building an efficient real-time. SMS spam filtering system based on machine learning. The main contribution of this work is to improve the performance of the system in real-time classification by focusing on data preparation, feature engineering, algorithm selection, and model deployment.

***Keywords*—**SMS spam filtering, Real-time classification, Machine Learning

INTRODUCTION

SMS remains a popular means of communication in the modern world of ongoing connectivity. The ubiquity of spam messages, however, taints the usefulness of SMS. Spam is not just a nuisance; it can include malicious links, phishing or fake content, exposing consumers to harm. Machine learning programs provide a powerful alternative to developing smart spam filters, but adapting these systems to the instantaneous needs of SMS filtering requires design and implementation caution. Amid the continuously changing terrain of communication technology, Short Message Service (SMS) is still a reliable performer, even amidst the spread of instant messaging software and social networks. Its simplicity, its universality across mobile phones, and its reliability in areas with limited internet access makes it a preferred method of communication for a wide range of personal and business communications. But the effectiveness of SMS as a mode of communication is threatened by the widespread invasion of spam messages. Spam via SMS is not just an irritation but a major issue with far-reaching ramifications. In addition to clogging inboxes and interfering with the user experience, spam messages typically come with malicious intent. They can include malicious links that point to phishing sites, malware downloads, or requests for sensitive information under the guise of legitimacy. These threats not only violate personal privacy and security but also erode confidence in SMS as a trustworthy form of communication. Resolution of the problem of SMS spam demands a comprehensive approach that includes technical solutions, awareness among users, and regulation. Of these solutions, machine learning (ML) algorithms have become a significant resource for designing intelligent spamfilters.

LITERATURE SURVEY

Pavas Navaney; Gaurav Dubey "SMS Spam Filtering Using Supervised Machine Learning Algorithms", suggested a system in 2018. Spam and Ham message detection based on different supervised machine learning algorithms such as naive Bayes Algorithm, support vector machines algorithm, and the maximum entropy algorithm and compares their performances in filtering the Ham and Spam messages. As individuals engage more in Web-based activities, and with increased sharing of private-data by firms, SMS spam is highly prevalent. SMS spam filter inherits a lot of functionality from E-mail Spam Filtering. Comparing the performance of different supervised learning algorithms we find the support vector machine algorithm provides us with the most precise result.In the Internet developing era, people are more and more in free online services. Pradeep kumar, jyothi "Deep learning to filter SMS Spam", suggested a system in 2020.The usage of short message service (SMS) has been increasing in the past decade. For companies, these messages are more efficient than even emails. This is due to the fact that although 98% of mobile users read their SMS at the end of the day, roughly 80% of the emails go unopened. Its popularity has also resulted in SMS Spam, which is any worthless text messages sent via mobile networks. They are

extremely bothersome to users. Most of the currently existing research that has tried to screen out SMS Spam has used manually extracted features. Expanding on existing literature, this paper applies deep learning to differentiate between Spam and Not-Spam text messages. In particular, Convolutional Neural Network and Long Short-Term Memory models were used. The new models were trained using text data alone, and self- derived the feature set. On a benchmark dataset with 747 Spam and 4,827 Not-Spam text messages, an impressive accuracy of 99.44% was attained.

3. OVERVIEW OF THE SYSTEM

3.1 Existing system

Current SMS spam filtering systems often employ a combination of methodologies, including rule-based filtering, simple machine learning, or a hybrid of both. Rule-based systems are based on pre-defined criteria, like keywords or blacklisted numbers, and thus are simpler to bypass and require regular updating. Machine learning systems employ methods like Naive Bayes or Support Vector Machines, which increase accuracy but at the cost of real- time efficiency. While these systems have advantages, they frequently struggle to successfully filter SMS within the milliseconds required for real-time settings. Furthermore, continuing spammer changes decrease system effectiveness unless datasets and models are updated on a regular basis. Spammers continuously change their messages, using new language, evasion techniques, and changing senders to evade filters. Systems that do not actively update their datasets or retrain their models quickly become ineffective. Limited Accuracy: Less complex systems,

especially those based primarily on rules, often generate both false positives (legitimate messages incorrectly labeled as spam) and false negatives (spam messages that get through). This degrades the user experience. Real-time Performance Challenges: Machine learning models designed for offline analysis might not be sufficiently quick for real-time filtering. Advanced feature engineering and computationally expensive methods restrict the ability to classify communications in milliseconds. Traditional SMS spam filtering systems use several strategies. Some use rule-based methods, where messages are blocked based on pre-defined keywords or blacklists. Though easy to deploy, such systems are simple to bypass by spammers and need to be constantly updated. Other systems employ simple machine learning algorithms such as Naive Bayes or Support Vector Machines.

PROPOSED SYSTEM

The suggested approach identifies spam with a Naive Bayes classifier trained on a well-annotated sample of SMS messages. Preprocessing of text involves traditional cleaning and normalization. The bag-of- words paradigm enables computationally inexpensive feature extraction. Naive Bayes is inherently speedy, highlighting real-time suitability. A server-side deployment method provides more versatility and ability for large data sets. Focused feature selection is expected to enhance accuracy. Periodic updating of datasets and retraining models are designed to counteract the effects of evolving spam tactics. The suggested approach identifies spam via a Naive Bayes classifier that has been trained on a highly selected and annotated set of SMS messages.

3.3 Proposed System

Design In this project work we used four modules and each module has own functions such as:

3.3.1 Scikit-Learn
3.3.2 NLTK (Natural Language Toolkit)
3.3.3 Random Forest
3.3.4 Scikit-Plot

3.3.1 Scikit-Learn

Scikit-learn, usually shortened as sk learn, is a fast and general machine learning library for Python. It provides simple and efficient tools for data analysis, machine learning modeling, and predictive analytics. Scikit learn is founded on top of other scientific computing libraries like NumPy, SciPy, and Matplotlib, utilizing their functionalities to provide a wide range of machine learning algorithms, data preprocessing methods, model evaluation tools, and visualization utilities. One of the most significant advantages of Scikit-learn is its ease of use and simplicity. The library offers a reliable and user-friendly API through which it becomes easy for users, ranging from novices to veteran data scientists, to incorporate multiple machine learning tasks.

3.3.2 NLTK (Natural Language Toolkit)

NLTK, or Natural Language Toolkit, is a Python library that is extensively used for natural language processing (NLP) operations. It offers a rich set of tools, algorithms, and resources for processing and analyzing human language data. NLTK is intended to support a range of NLP operations like tokenization, stemming, lemmatization, part-of-speech tagging, named entity recognition, sentiment analysis, and more. One of NLTK's major strengths is its large library of corpora and lexical resources, such as text collections, word lists, and linguistic resources for various languages. These corpora are useful datasets for training and testing NLP algorithms and models so that researchers and practitioners can try out a variety of techniques and test their performance on real language data. NLTK also offers text preprocessing and normalization functionalities, which facilitate the cleaning and preparation of text data for analysis.

3.3.3 Random Forest

Random Forest is a robust ensemble learning algorithm commonly applied to machine learning both for classification and regression problems. Random Forest relies fundamentally on decision trees. Decision trees are hierarchical models that use predictions by partitioning the data according to features at each node, resulting in leaf nodes being the final prediction or outcome. However, decision trees can suffer from overfitting, particularly when confronted with noisy or intricate datasets.

3.3.4 Scikit Plot

Scikit-plot is a Python package developed on top of the widely used machine learning library Scikit-learn and the Matplotlib visualization library. It aims to make visualization generation easier in machine learning processes, especially in model assessment and data exploration. Using Scikit-plot, machine learning professionals and data scientists are able to produce informative and useful plots within a few lines of code to better understand the model performance, data distribution, and relationships of the dataset.

Architecture

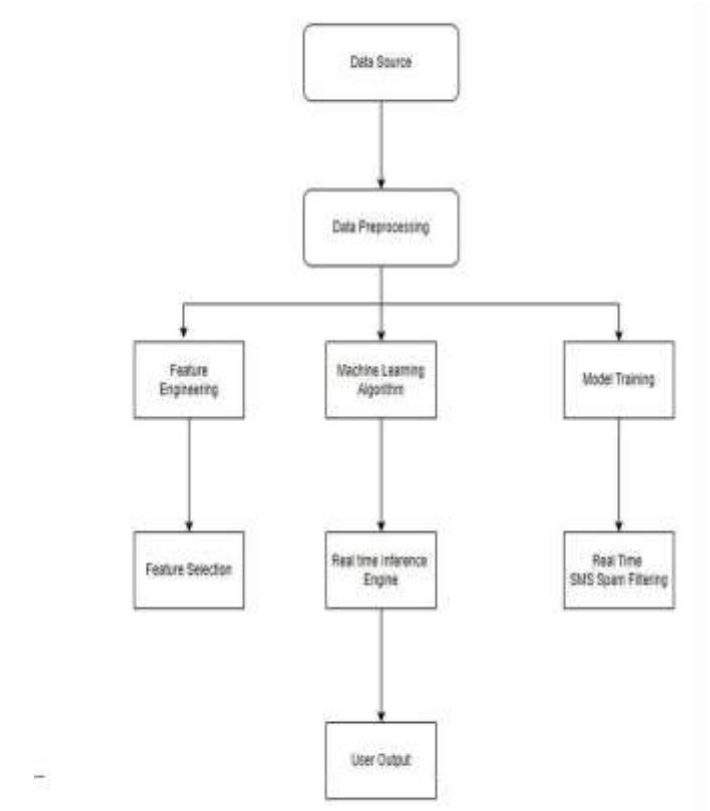


Fig 1: System Architecture

Economical Feasibility:

The real-time SMS spam filtering project is economically viable as it can significantly reduce the costs incurred by mobile users and service providers in terms of spam-related issues. The objective of the project is to minimize the ill effects of SMS spam on customer experience by adopting a powerful machine learning-based approach, which will enhance customer satisfaction and loyalty. The cost of developing and deploying the system can be recovered in the long term by reduced spam-related grievances, reduced support costs, and the potential revenue losses from dissatisfied customers. Additionally, by optimizing computational capacity and processing time, the low latency and high accuracy of the system minimize the costs of operating. Generally, the economic sustainability of the project is supported by its ability to generate tangible benefits in terms of reduced costs, improved user experiences, and higher operational efficiency in the battle against SMS spam.

Technical Feasibility

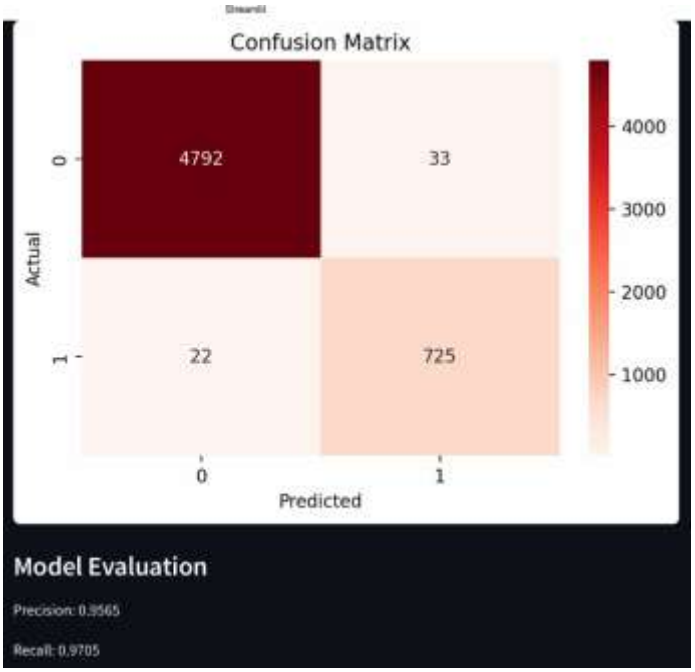
Advances in machine learning techniques, cloud computing technology, and natural language processing (NLP) methods bring technical support to the initiative for real-time filtering of SMS spam. Constructing models and training them become simplified by such libraries as TensorFlow and PyTorch, while cloud infrastructure makes scalable processes possible for rapid handling of huge quantities of SMS content. Advances in NLP enhance feature extraction capabilities, raising standards for precision of spam detection. Real-time APIs enable seamless interaction between the filtering system and the mobile networks to ensure instant classification of incoming communications. These technology components integrate with efficient data pipelines and integration protocols to generate a robust and secure system that can combat real-time SMS spam with low interference to user experience and high accuracy. The combination of real- time APIs and messaging protocols allows easy communication between the spam filtering system and mobile networks, allowing for timely classification of incoming SMS messages.

Feasibility

Since the real-time SMS spam filtering project can enhance user enjoyment, faith, and overall mobile communication experience, it is socially viable. The project enables mobile customers' text messaging to be more fun and safer through effectively blocking SMS spam. Mobile users may engage more with mobile services because of this enhanced quality, and they might get annoyed less with unsolicited messages. Moreover, the project meets social norms regarding data protection and privacy since it protects users from phishing scams and fraudulent activities, which are often associated with spam messages. In addition, the minimal effect of the system on user experience ensures that legitimate messages are not filtered out erroneously, maintaining seamless communication. Overall positive impact of the project on user well-being, trust in mobile services, and overall Messaging platform satisfaction definitively proves its social viability.

RESULTS SCREENSHOTS:

The following diagram illustrates how message lengths are distributed within a spam detection dataset. The x-axis is the message length, and the y-axis is how many messages have that length. Blue bars indicate normal (non-spam) messages, and orange bars indicate spam messages. Non-spam messages tend to be short, typically less than 100 characters, whereas spam messages tend to be longer, typically around 150 to 200 characters. This indicates that message length can be used to identify spam since spam messages tend to have more text, e.g., offers or links. This renders message length a good feature to use when training machine learning algorithms to identify spam.



RESULT

The model was tested against a confusion matrix and performance metrics. The confusion matrix is an overall count of the correct and incorrect predictions by the model for each class. The model is highly performing, as illustrated below:

•True Negatives (4792): 4792 instances were correctly labeled as class 0 by the model.

•True Positives (725): 725 instances were correctly labeled as class 1 by the model.

•False Positives (33): 33 instances were wrongly predicted as class 1, but were class 0.

•False Negatives (22): 22 instances were wrongly predicted as class 0, but were class 1.

Distribution of Message Length

• Plot Type: Histogram with KDE (Kernel Density Estimation) curves

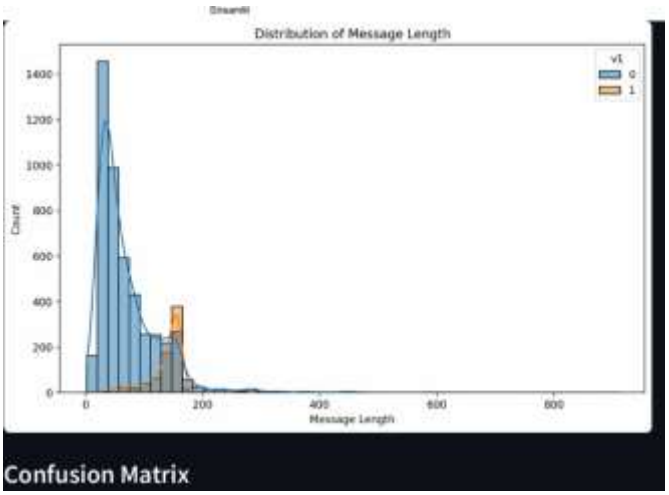
• X-axis: Message Length (number of characters or words in the messages)

• Y-axis: Count (number of messages for each length range)

• Legend:

• v1 = 0 (Blue): Normal (non-spam) messages.

• v1 = 1 (Orange): Represents spam messages.



CONCLUSION

The constant fight against SMS spam has been a recurring problem for mobile consumers, leading to frustration and inconvenience. Conventional spam filtering methods tend to be inadequate in real-time scenarios, requiring the application of sophisticated machine learning methods. This project explores the creation of a real-time SMS spam filtering system that utilizes machine learning to effectively solve this problem. One of the main areas of concern in this venture is data preparation, where data is gathered and preprocessed from large sets of SMS to pull out relevant features for training models. Feature engineering is one of the critical tasks in turning raw data into meaningful features that reveal the character of spam messages, including keyword frequency, message length, and sender reputation. These engineered attributes are subsequently employed to train machine learning models that can differentiate between valid messages and spam in real time. Algorithm choice is another essential factor taken into account in this research, as various machine learning algorithms have different performance profiles in terms of accuracy, speed, and resource utilization. By extensive experimentation and testing, the most appropriate algorithms for real-time SMS spam classification are determined and incorporated into filtering system. Model deployment is an important step in the development process, during which the learned machine learning models are rolled out to production systems to process arriving SMS messages in real-time. The deployment stage includes tuning the system for low latency and as little interruption as possible to the user experience. Effective model deployment techniques, e.g., containerization or cloud centric design. User feedback

channels and usability testing are integrated to measure the effectiveness of the system from the end user's point of view. This people-focused approach guarantees that not only does the SMS spam filtering system meet its technical obligations, but it also makes the overall messaging experience better for mobile consumers. based solutions are investigated in order to guarantee smooth integration with current infrastructure. Also, continuous monitoring and updating of models are provided to cope with the dynamics of changing spam

patterns and ensure that high filtering accuracy is sustained over time. During the project, attention is not just placed on technical excellence but also on user.

FUTURE ENHANCEMENT

The future of this real-time SMS spam filtering system lies in its ability to improve efficiency and adaptability to emerging challenges. This can be achieved through an ongoing process of improving data preparation methods, feature engineering, and algorithm choice. Further exploration of more advanced machine learning models, like deep learning architectures, can similarly revolutionize the system's accuracy as well as its capacity for complex spam patterns. Integration of user feedback and preferences to make the filtering system personalized will also be important in improving user experience. Additionally, extending the system's applicability to other messaging platforms and languages can extend its reach in spam fighting across multiple communication channels. Another potential direction for future research is the use of natural language processing (NLP) techniques to better analyze the content of SMS messages. This can assist in detecting fine-grained spam traits, like the employment of misleading language or manipulative strategies, which might be presently escaping the system. Finally, investigating the possibility of edge computing and distributed systems for real-time SMS spam filtering can assist in minimizing latency and enhancing the overall performance of the system, making it more scalable and responsive to the dynamic nature of spam threats. Adding natural language processing (NLP) methods to the system can also enhance its spam filtering abilities by examining the semantic meaning and context of SMS messages. By being able to recognize the nuances of language use, the system can detect misleading or manipulative spam messages that can avoid conventional keyword-based filtering. Given developments in edge computing and distributed systems, investigating these technologies for real-time SMS spam filtering can result in lower latency and greater scalability. Edge computing makes it possible to process data nearer the source, reducing communication overhead as well as response time, while distributed systems are capable of efficiently spreading the computing load across nodes, making their performance very resilient against increased loads. Personalization is yet another major element in future development, whereby incorporating user feedback and personal preferences can customize the filtering system to the individual needs and desires of users. This personalized mechanism not only increases the user's experience but also the system's efficiency by responding to users' changing communication habits and spam tolerance levels. Broadening the system's coverage to othermessaging platforms to accommodate the specifics of each platform and language. beyond SMS, including instant messaging software or social media websites, can greatly extend its reach in fighting spam in various channels of communication. This can involve tailoring the system's features and algorithms

ACKNOWLEDGMENT

In the digital age, spam messages—ranging from unsolicited advertisements to phishing attacks—pose a significant challenge to online communication platforms, including email services, SMS gateways, and social media networks. These unwanted messages not only clutter inboxes but also pose serious security threats such as identity theft, malware distribution, and financial scams. Traditional spam filters, often based on manually crafted rules and blacklists, struggle to keep pace with the rapidly evolving tactics used by spammers. Hence, there is a growing need for adaptive, intelligent systems capable of learning from patterns in data. This project investigates the application of machine learning (ML) techniques for efficient and accurate spam detection. We employed algorithms such as Naive Bayes, Support Vector Machines (SVM), Logistic Regression, Decision Trees, and ensemble methods like Random Forests to build predictive models that classify messages as spam or non-spam (ham). To develop and test these models, we utilized publicly available datasets such as the SMS Spam Collection Dataset and Kaggle's Spam Email datasets. These datasets provided diverse samples of legitimate and spam messages, allowing us to train models that generalize well across different types of spam content. Preprocessing steps—including text cleaning, tokenization, stop- word removal, and lemmatization—were applied to normalize the input data. We also implemented feature extraction techniques like Bag of Words and TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into a structured format that machine learning algorithms can understand. Furthermore, we integrated natural language processing (NLP) techniques to capture contextual and semantic nuances in message content. Performance evaluation was carried out using metrics such as accuracy, precision, recall, F1-score, and confusion matrices. Our models demonstrated strong performance in detecting spam, with some achieving precision and recall scores exceeding 95%, depending on the algorithm and feature set used.

Beyond classification, this project also explores the interpretability of models and highlights the importance of balancing performance with explainability, especially when deploying these systems in real-world applications. The study concludes that machine learning provides a robust, scalable, and adaptable framework for combating spam, with potential for integration into email servers, messaging platforms, and cybersecurity systems. Future work may include exploring deep learning architectures like LSTM, BERT, or transformers, and developing real-time detection systems that can handle multilingual or adversarial spam content effectively.

REFERENCES

1. Sharma, S., & Bhondekar, A. P. (2020). Spam email detection using machine learning approach. Procedia Computer Science, 167, 370–378.

2. Kim, B., Abuadbba, S., & Kim, H. (2020). DeepCapture: Image Spam Detection Using Deep Learning and Data Augmentation. arXiv preprint arXiv:2006.08885.

3. Fattahi, J., & Mejri, M. (2020). SpaML: A Bimodal Ensemble Learning Spam Detector based on NLP Techniques. arXiv preprint arXiv:2010.07444.

4. Agboola, O. S. (2020). Spam Detection Using Machine Learning. Computer Engineering and Intelligent Systems, 11(3).

5. Vashisth, S., Dhall, I., & Aggarwal, G. (2020). An Approach to Automated Spam Detection Using Deep Neural Network and Machine Learning Classifiers. In Micro-Electronics and Telecommunication Engineering (pp. 151-159). Springer, Singapore.

6. Authors Unknown. (2020). Using machine learning to deal with Phishing and Spam Detection. In Proceedings of the 3rd International Conference on Networking, Information Systems & Security (pp. 1-6). ACM.

7. Authors Unknown. (2021). Evaluating the Effectiveness of Machine Learning Methods for Spam Detection. Procedia Computer Science, 190, 479-486.

8. Shaaban, M. A., Hassan, Y. F., & Guirguis, S. K. (2021). Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. arXiv preprint arXiv:2110.15718.

9. Guo, Y., Mustafaoglu, Z., & Koundal, D. (2022). Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms. Journal of Computational and Cognitive Engineering, 2(1), 5-9.

10. Zhang, Z., Damiani, E., Al Hamadi, H., Yeun,

11. C. Y., & Taher, F. (2022). Explainable Artificial Intelligence to Detect Image Spam Using Convolutional Neural Network. arXiv preprint arXiv:2209.03166.

12. Authors Unknown. (2022). Spam detection on social networks using deep contextualized word representation. Multimedia Tools and Applications, 82, 3697–3712.

13. Authors Unknown. (2020). Spam and phishing in Q3 2020. Securelist by Kaspersky.

14. Authors Unknown. (2020). Gmail Is Catching More Malicious Attachments With Deep Learning. WIRED.

15. Zhang, Z., Damiani, E., Al Hamadi, H., Yeun, C. Y., & Taher, F. (2022). A Late Multi-Modal Fusion Model for Detecting Hybrid Spam E-mail.

16. Uddin, M. A., Islam, M. N., Maglaras, L., Janicke, H., & Sarker, I. H. (2024). ExplainableDetector: Exploring Transformer-based Language Modeling Approach for SMS Spam Detection with Explainability Analysis.

17. Wani, M. A., ElAffendi, M., & Shakil, K. A. (2024). AI- Generated Spam Review Detection Framework with Deep Learning Algorithms and Natural Language Processing. Computers, 13(10), 264.

18. Zhang, Z., Damiani, E., Al Hamadi, H., Yeun, C. Y., & Taher, F. (2022). Explainable Artificial Intelligence to Detect Image Spam Using Convolutional Neural Network.

19. Guo, Y., Mustafaoglu, Z., & Koundal, D. (2022). Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms. Journal of Computational and Cognitive Engineering, 2(1), 5-9.

20. Danilchenko, K., Segal, M., & Vilenchik, D. (2022). Opinion Spam Detection: A New Approach Using Machine Learning and Network-Based Algorithms. Proceedings of the International AAAI Conference on Web and Social Media, 16(1), 125-134.