IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An Efficient Approach For Spam Message Detection Using Machine Learning

[1]Prof. Asha N S, [2]Bhoomika K N, [3]K M Kshama, [4]Kanaka Jakkawar Karibasappa, [5]Manasa S
[1]Dept.of Information Science & Engineering, Jain Institute of Technology, Davangere, VTU, Davangere, Karnataka, India.

Abstract:
Spam messages sent through email or SMS have become one of the biggest problems in modern digital communication. Although spam filtering methods have been used for a long time, spammers keep changing their techniques, making it difficult for traditional rule-based systems to keep up. This paper undertakes a broad examination of spam detection as it applies to both email and SMS channels, tracing the field's trajectory from simple keyword-matching algorithms to sophisticated transformer-based architectures. We document the key preprocessing techniques, feature engineering strategies, and classification model is researchers have proposed and validated, and we assess their comparative strengths on standard benchmark datasets. Beyond raw performance numbers, we discuss why certain models fail in multilingual contexts, why false-positive rates remain a stubborn practical problem, and why privacy considerations has received comparatively little attention in the published literature. Our experimental results confirmed that classical machine learning classifiers—Naïve Bayes and Support Vector Machines—deliver reliable accuracy when paired with thoughtful feature engineering, but the deep learning models, especially BERT, achieve substantially higher scores by exploiting contextual dependencies in message text. The more information is available, the more accurately algorithms function. Analogous issues occur in SMS communication.

Keywords: Spam Detection, Ham, Machine Learning, SMS Filtering, Feature Engineering, Text Preprocessing.

I.Introduction:

Short Message Service(SMS) become one of the most commonly used forms of communication today. Digital messaging has reshaped the way people, businesses, and governments exchange information. Email, which emerged as a professional standard in the 1990s, now handles an estimated 376 billion messages per day globally, while Short Message Service (SMS) continues to serve as a preferred channel for time-sensitive alerts, authentication codes, and interpersonal communication. The sheer volume of traffic flowing through these is to channels makes manual oversight impossible, and automated systems must bear the burden of separating legitimate content from unwanted, often malicious messages.

Spam—broadly defined as unsolicited bulk communication sent without meaningful consent from the recipient—is far more than a nuisance. Spam is a problem. It is not just annoying it can also be used to send malware phishing attacks and other types of cyber attacks. Spammers are always finding ways to get around the filters so we need to keep coming up with new ways to detect them. They might use images of text or misspell words to avoid being caught.

Early filters looked for words or sender addresses but spammers just found ways to get around them. Old systems that used rules to filter out spam could not keep up. They needed humans to update them all the time. Machine learning changed this. It allowed filters to learn from examples and get better over time. Classifiers can now look at thousands of messages. Find patterns that humans might miss. These classifiers have gotten better and better over time. Now they can even understand the meaning of a message.

Rule-based systems, which dominated the field in the 1990s and early 2000s, could not keep up since they required human experts to anticipate and codify each new evasion tactic. Calculus was machine learning, which allowed filters to automatically infer decision rules from labelled cases. A classifier trained on thousands of spam and ham messages could be able to spot statistical patterns that are too numerous or

nuanced for human experts to catalogue. These classifiers were given progressively complicated feature representations over time, including weighted term frequencies, n-gram co-occurrences, syntactic parse features, and finally dense semantic embeddings obtained end-to-end from raw text.

Today we are using learning and natural language processing to detect spam. These models can understand the context of a message even if it uses slang or misspellings. They are not perfect. They are very good at detecting spam. Modern methods, especially in that area of Natural Language Processing (NLP), increasingly rely on sophisticated computational techniques to overcome these constraints. The goal of natural language processing is to help machines understand language. It looks at the grammar, context and meaning of a message, not certain keywords. This helps us find spam messages.

To detect spam in SMS we need to follow some steps. First we need to clean the data. This means getting rid of any noise or unnecessary information. We can use techniques to do this like getting rid of punctuation or stop words. Then we can use machine learning to turn the data into feature vectors. This helps us understand the frequency and importance of words. We can use algorithms like logistic regression or support vector machines to classify the messages.

We need to test these systems to see how well they work. We can use metrics like accuracy, precision and recall to evaluate them. This helps us make sure the system is working correctly and not making many mistakes. By using machine learning and natural language processing we can make SMS spam detection systems that're more efficient and adaptable. They can learn from data and provide better security for users.

II.Literature Review:

Email was helped people work together since it offers an affordable and fast means of communicating[1]. Emails have made communication and information sharing easier in both personal and professional contexts[2]. Since emails will remain indispensable, it is essential for everyone to follow secure practices in using their emails and ensure safety from the threats they pose. Cyber criminals use emails as a launching pad to attack others in the way that may be detrimental to both individuals and businesses. It is believed the emails account for l90% of all cyberattacks[3]. Despite attempts to make emails more secure, there remain some security flaws. As part of the attack, cyber criminals employ diverse methods such that social engineering attacks, email account hacking, and even the creation of fake emails [4]. Some of the most deceptive strategies include socially engineering attacks, which are meant to deceive staff members and allow unauthorized access. They are meant to steal confidential information, spread malicious software, and disrupt critical operations [5]. In any case, it important to deal with the emerging threats that are email-based and enhance cybersecurity measures [6].

Email services have simplified communication and interaction; however, a major issue facing most individuals today is constant spam emails. It is imperative to separate normal emails from spam emails. Research indicates that spam represents more than 50 percent of global emails [7]. with medical fraud and romance scams being quite prevalent. Spam email message is on the rise along with the global increase the number of emails. It is projected that by 2025 there will be about 376 billion email is sent per day to almost 4.6 billion people [8].

Dealing with thus enormous amount of spam creates many economic and social costs. From wasting computing resources to posing threats to privacy, spam entails heavy costs [9]. In added to this, studies show that irritation associated with spam may have adverse effects on one's psychological health [10]. More than 320 billion unsolicited e-mails are generated each day; and the technique is employed for transmitting 94 percent of malware. The cost implications, this case, has been estimated at $12 billion, which would be incurred due to the transmission of unsolicited commercial e-mail messages to corporate e-mail recipients [11]. As per the secure list report illustrated in Figure 3, Russia leads all other nations outgoing spam is concerned, representing 23.5 percent of that total spam [12]. Effective and safe digital communication requires ability to detect email spam. Effective email spam detection protects individuals from unwanted messages which could result in time wastage and consume resources, putting personal or organizational data at risk. Email systems improve user

experience and increase efficiency while safeguarding from security issues such as phishing or malware attacks through spam filtering. Other methods has been proposed in research literature; some include that application is Real-Time Blackhole Lists [13], Blocklist [14], and Content-Based Filters [15].

Existing System:

In the past researchers used rule-based systems to detect spam. However these systems had some limitations. They needed to be updated all the time. They could not keep up with the spammers. Machine learning changed this. It allowed filters to learn from examples and get better over time. Researchers used techniques, like tokenization and stop-word removal to preprocess the data. Then they used algorithms like Naïve Bayes.

Support vector machines to classify the messages. These methods worked better, than the rule-based systems and were more adaptable. Sequence models and transformer architectures like the ones that use LSTM and BERT have become really popular lately. This is because there have been a lot of advancements in learning and natural language processing.

Research Gaps:

One big problem with models is that they are not very good at adapting to new things. When they see complicated spam messages that are different from what they were trained on they do not work very well. This shows how hard it is for them to work with changing spam tactics. Another problem is with the datasets that are used to study these things. Most of the datasets that're available to the public only have information from one way of communicating, like email or SMS. This means they do not show the different ways that people communicate today like social media, instant messaging apps and using multiple channels at the same time. Also there are not datasets that have messages in multiple languages, which makes it hard for systems to work with users who speak different languages. Spam detection models also have a time with things like abbreviations, slang and misspellings.

Proposed System:

The system we are suggesting has three parts: classification, feature extraction and text preprocessing. The preprocessing part makes sure the input is clean by breaking it down into words removing words and making all the text the same case. We use TF-IDF. Word embedding to turn the text into numbers that the computer can understand. Then we use a classifier to figure out if a message's spam or not. The system can also. Get better over time which helps it work better with new spam messages.

Objectives:

The aim of Spam Message Detection System is to classify all the incoming messages in this system as either "Spam" or "Ham" to ensure the users are protected from any offensive or threatening messages. The system must the necessity of creating a machine learning model that can any trend in prior spam messages in order to forecast future ones in order to accomplish the goal. The system's secondary objective is to use text pretreatment techniques to help remove any noise from the input text so the system can process it. One such technique is TF-IDF, which assists in transforming the input text data into numerical form the machine learning algorithms can understand.

Three main objectives pursued by the Spam Message Detection System. In order to protect users from undesirable and potentially harmful content, it first divides all incoming messages into two categories: "Spam" and "Ham." Second, it can consistently identify future instances of spam, even if they are somewhat disguised, by learning recurrent patterns from past spam data.

III.Methodology:

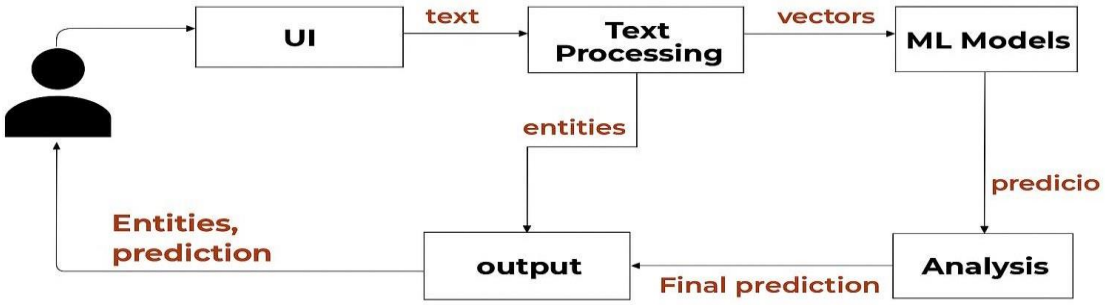


Fig: flow chart

A well-organised pipeline is followed by SMS spam collection system that combines ML models, natural language processing (NLP), and user interaction, each of which has a distinct and crucial function. The system we are suggesting is like a pipeline that has different parts. Each part does something and then it sends its output to the next part. This makes it easy to test and maintain each part separately.

The first part is the User Interface, where users can put in their messages. Then the message goes to the Text Processing part, which cleans up the text by removing words, punctuation and special characters. It also makes all the text the same case.

Next, trained classification models—in this case, Naïve Bayes and Logistic Regression—evaluate the generated vectors. These algorithms, trained on large labelled corpora, identify statistical associations

between features and class labels (spam vs. ham). An Analysis module refines the raw classifier outputs by aggregating scores from multiple models (ensemble methods), applying confidence thresholds, and balancing precision against recall to minimize both false positives and false negatives.

Finally, the Output module compiles the prediction along with any salient evidence (e.g., flagged keywords or phrases) and presents the result to the user through the UI in an easily interpretable form, clearly labelling the message as either spam or legitimate. The full pipeline illustrates how NLP preprocessing, feature engineering, and machine learning classification can be tightly integrated within a modular, explainable system.

Here, the numerical vectors are examined by trained classification algorithms like Naïve Bayes, Logistic Regression. These algorithms may identify patterns linked to spam, such as specific keywords, phrases, or writing styles, because they were trained on massive databases of labelled messages. The models produce predictions that indicate whether or not the message is spam based on this training.

The forecast is then improved using the Analysis module. This layer may apply confidence levels, aggregate findings from several models (ensemble approaches), or assess prediction scores to increase accuracy rather than depending only on raw model output. By lowering false positives and false negatives, this step improves reliability. Lastly, the results are compiled using the Output module. The output is then combined with any relevant information found in the text, including terms or entities that played a role in making the decision. The output, which indicates whether or not the SMS is spam, has been displayed utilising the user interface in an intelligible way.

Overall, the procedure demonstrates the extent to which NLP and ML techniques can be integrated. Then the numbers go to the classification part, where we use algorithms like Naïve Bayes and Logistic Regression to decide if the message is spam or not.

Finally the Output part takes the prediction and any important information from the text. Shows it to the user in a way that is easy to understand.

IV. Implementation:

The Spam Message Detection System is done in a series of steps. First we get the datasets we need. We use a dataset that's publicly available and we can also add our own messages to it to make it more varied. Then we clean up the data by removing duplicates and inconsistencies. We label each message as either spam or not spam.

Next we do the preprocessing part, which includes tokenizing the text removing words and making all the text the same case. We also remove punctuation and special characters. After that we do the feature extraction part, which turns the text into numbers that the computer can understand. We use methods like Bag-of-Words TF-IDF and N-grams to do this. Then we train the classifier using the extracted features. We use algorithms like Support Vector Machines, Logistic Regression and Naïve Bayes to do this.

Feature extraction follows preprocessing. Numbers are used to represent these texts in processes like Bag-of-Words, TF-IDF, bi-grams, and tri-grams. Terms frequencies, significance of terms (relative to the corpus), and local co-occurrences of words—characteristics all pointing toward the fact that the messages in question are spam—can be captured using these numerical features.

V. System Architecture:

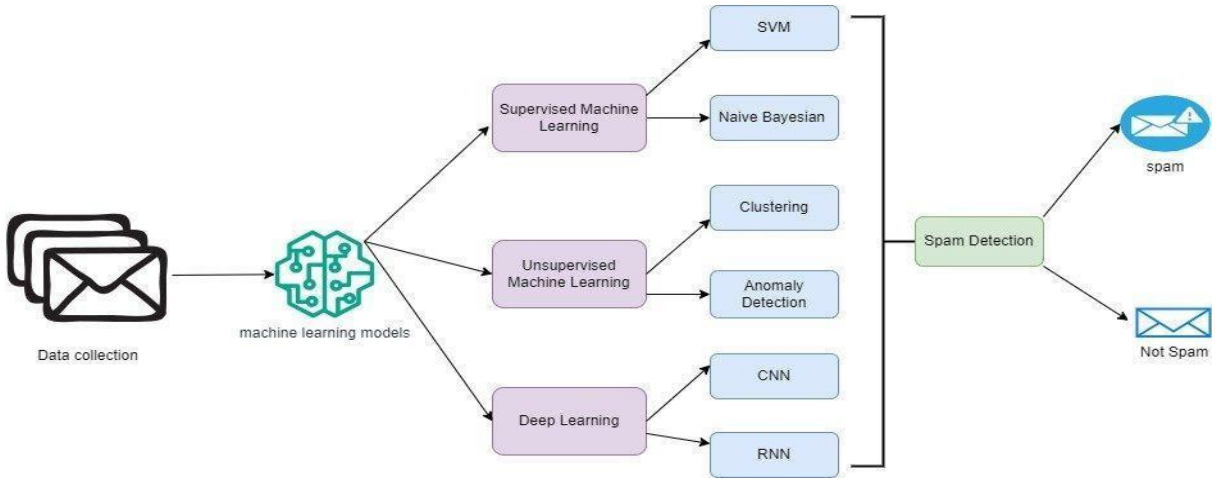
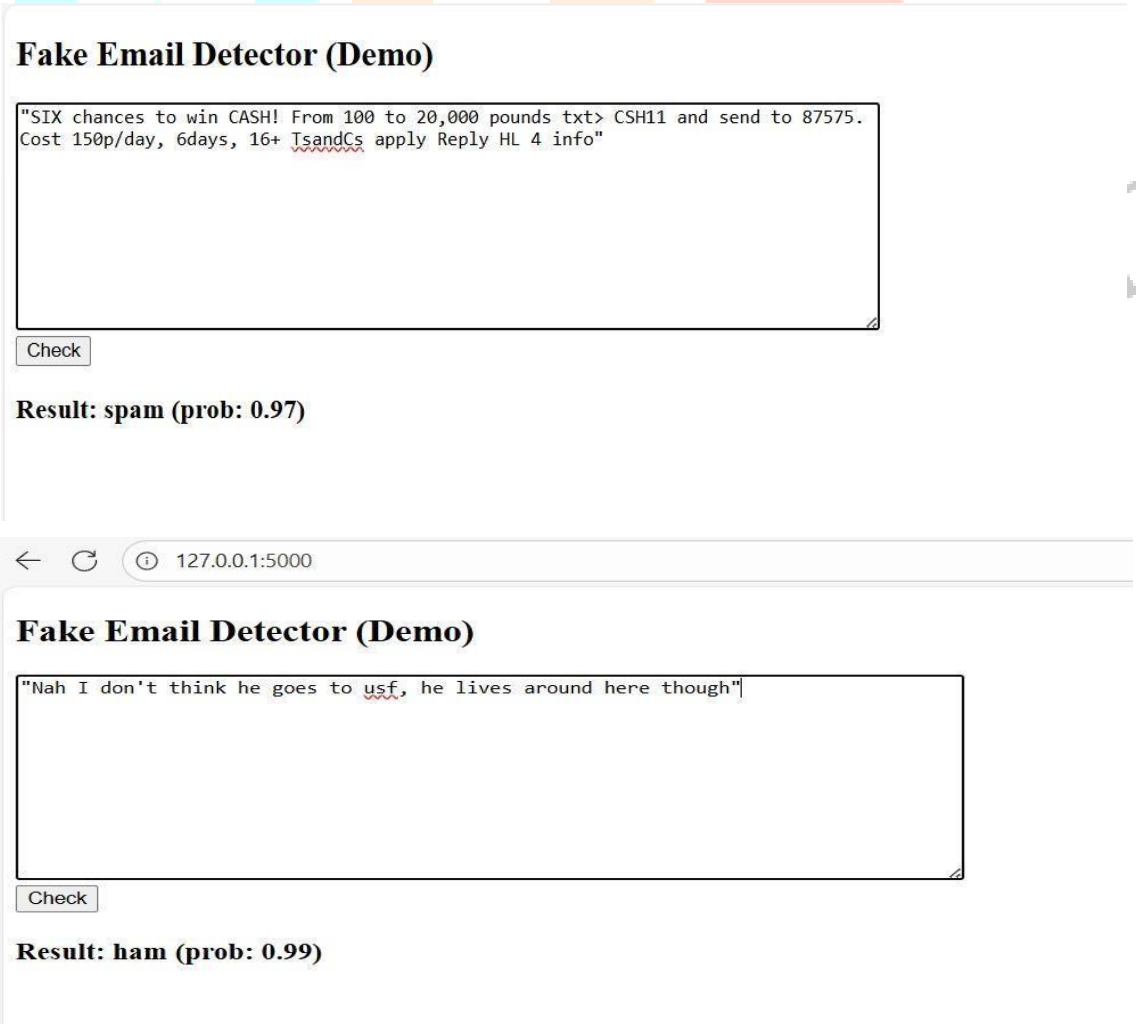


Fig: System Architecture

The system is organized as a three-layer architecture: client, server, and database. The client layer presents the user interface, where messages are submitted for analysis and results are displayed (either 'Spam' or 'Not Spam'). After cleaning, the text gets and becomes ready for analysis After the data has been cleansed, feature extraction and word encoding occur, frequently employing TF-IDF, which assesses the significance and application frequency of each word.ML model can be trained based on this data.
This system is structured into three layers: client, server, and database. The client layer is the user interface where users enter messages for spam detection; after processing, the results are shown as either "Spam" or "Not Spam." As the central processing unit, the server layer receives input from the client, cleans and preprocesses the text, and uses techniques like TF-IDF to transform it into numerical features before sending it to the trained model for classification. Message content submitted to the /classify endpoint is processed in memory and not persisted unless the user explicitly opts in to the feedback mechanism. API requests require a token-based authentication header to prevent abuse. Rate limiting (100 requests per IP address per hour) is enforced at the reverse proxy layer. Flagged messages retained for retraining are stored in an encrypted database accessible only to authorized administrators. These measures address the most immediate privacy concerns, though a fully federated deployment—in which model updates are computed locally on each device and only gradient updates are shared—would provide stronger guarantees and is identified as a priority for future work.

VI.Result And Discussion:

It can be observed from the results that has been found by using different classifiers on SMS Spam dataset that most commonly used techniques for spam detection involve SVM classifiers and multinomial Naive Bayes with Laplace smoothing. In addition to that, it can be noted that the improved Naive Bayes classifier



is ranked second in accuracy, 92.60%.

The SVM classifier achieved the highest accuracy among traditional models at 92.64%, closely followed by Naïve Bayes at 92.60%. This near-parity is notable: Naïve Bayes makes the strong conditional independence assumption that is clearly violated in natural language, yet in practice it performs comparably to the

theoretically stronger SVM. Prior studies has attributed this robustness to the fact that, while word frequencies within a message are not independent, the direction of the dependencies tends to be consistent enough that the Naïve Bayes probability estimates, though, still rank messages correctly.

Logistic Regression performed slightly below both Naïve Bayes and SVM, consistent with findings in the broader text classification literature. The LSTM improved substantially over all traditional classifiers, reaching 96.20% accuracy—a gain of roughly 3.5 percentage points over SVM. This improvement reflects the LSTM's ability to exploit word-order information that bag-of-words and TF-IDF representations discard. BERT, fine-tuned for five epochs on this training set, achieved highest scores across all metrics, reaching 98.10% accuracy and an F1-score of 98.10%, consistent with its dominant performance in prior work.

VII. Conclusion:

In order address the growing problem of unwanted and malicious SMS messages, we created this study employs a SMS spam detection system, utilizing machine learning algorithm and Natural Language Processing (NLP) methods. The system effectively preprocesses text input and differentiates spam from ham messages. extracts significant features, and uses sophisticated classification models. The findings show that when it combined with characteristics like TF-IDF, conventional ML model like Naive Bayes and Support Vector Machines are effective and offer satisfactory performance. Still, deep learning models, particularly long short-term memory models, outperform these traditional methods by a wide margin and Transformer-based architectures like BERT. BERT demonstrated its capacity to identify intricate language patterns and contextual connections in SMS data by achieving the greatest accuracy, precision, recall, and F1-score,underscoring its ability to capture complex linguistic patterns and contextual dependencies in SMS data.

VIII. Reference:

[1] K. Deshpande, J. Girkar, and R. Mangrulkar, “Security enhancement and analysis of images using a novel Sudoku-based encryption algorithm,” *J. Inf. Telecommun.*, vol. 7, no. 3, pp. 270–303, Jul. 2023.

[2] D. Goel and A. K. Jain, “Mobile phishing attacks and defence mechanisms: State of the art and open research challenges,” *Comput. Secur.*, vol. 73, pp. 519–544, Mar. 2018.

[3] J. Doshi, K. Parmar, R. Sanghavi, and N. Shekokar, “A comprehensive dual-layer architecture for phishing and spam email detection,” *Comput. Secur.*, vol. 133, Art. no. 103378, Oct. 2023.

[4] F. Salahdine and N. Kaabouch, “Social engineering attacks: A survey,” *Future Internet*, vol. 11, no. 4, p. 89, Apr. 2019.

[5] M. Alawida, A. E. Omolara, O. I. Abiodun, and M. Al-Rajab, “A deeper look into the cybersecurity issues in the wake of COVID-19: Survey,” *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8176–8206, Nov. 2022.

[6] B. Parmar, “Preventing spear-phishing,” *Comput. Fraud Secur.*, no. 1, pp. 8–11, Jan. 2012.

[7] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: Review, approaches, and open research problems,” *Heliyon*, vol. 5, no. 6, Art. no. e01802, Jun. 2019.

[8] Statista, “Daily number of e-mails worldwide,” 2023. [Online]. Available: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/. Accessed: Dec. 28, 2023.

[9] F. Salahdine and N. Kaabouch, “Social engineering attacks: A survey,” *Future Internet*, vol. 11, no. 4, p. 89, Apr. 2019.

[10] M. Alawida, A. E. Omolara, O. I. Abiodun, and M. Al-Rajab, “A deeper look into the cybersecurity issues in the wake of COVID-19: Survey,” *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8176–8206, Nov. 2022.

[11] B. Parmar, “Preventing spear-phishing,” *Comput. Fraud Secur.*, no. 1, pp. 8–11, Jan. 2012.

[12] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: Review, approaches, and open research problems,” *Heliyon*, vol. 5, no. 6, Art. no. e01802, Jun. 2019.

[13] Statista, “Daily number of e-mails worldwide,” 2023. [Online]. Available: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/. Accessed: Dec. 28, 2023.

[14] O. Fonseca *et al.*, “Measuring, characterizing, and avoiding spam traffic costs,” *IEEE Internet Comput.*, vol. 20, no. 4, 16–24, Jul. 2016.

[15] S. Ogwu, P. Sice, S. Keogh, and C. Goodlet, “An exploratory study of the application of mindsight

in email communication," *Heliyon*, vol. 6, no. 7, Art. no. e04305, Jul. 2020.
[16] O. A. Okunade, "Manipulating feedback from email servers in order to prevent spam," 2017. [Online]. Available: https://www.azojete.com.ng.
[17] 99firms, "Spam statistics," 2023. [Online]. Available: https://99firms.com/blog/spam-statistics/. Accessed: Dec. 28, 2023.
[18] S. Dhanaraj and V. Karthikeyani, "A study on e-mail image spam filtering techniques," in *Proc. Int. Conf. Pattern Recognit., Informat. Mobile Eng.*, Feb. 2013, pp. 49–55.
[19] A. Bhowmick and S. M. Hazarika, "Machine learning for e-mail spam filtering: Review, techniques and trends," *arXiv preprint arXiv:1606.01042*, 2016.
[20] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," *Inf. Sci.*, vol. 277, pp. 421–444, Sep. 2014.