Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 7
TPCEE 2024

Enhancing Spam Email Detection with Machine Learning: A Comparative Study of Logistic Regression and Naive Bayes Using Apache Spark

Zhaoyang Ye *

School of Mathematics, South China University of Technology, Guangzhou, China

* Corresponding Author Email: 202130322339@mail.scut.edu.cn

Abstract. The spread of spam emails presents serious problems for both email security and user experience. This research aims to develop an effective spam email classification system utilizing machine learning techniques, specifically Logistic Regression and Naive Bayes, within the Apache Spark framework. The methodology encompasses a thorough preprocessing of the Enron email dataset. This process involves several critical steps: text cleaning to remove irrelevant information, tokenization to break down the text into individual words, removal of stop words to eliminate common but uninformative words, and text feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF) to quantify the importance of terms within the dataset. The study is conducted on a subset of the Enron email dataset, comprising 11,029 emails, with 2,996 labeled as spam. Experimental results demonstrate that the Naive Bayes model outperforms Logistic Regression, achieving higher accuracy and F1 score. This finding underscores the robustness of Naive Bayes in spam email classification, highlighting its potential for enhancing email security by effectively filtering spam.

Keywords: Spam; Machine Learning; Naive Bayes; Logistic Regression; Apache Spark.

1. Introduction

With the proliferation of the internet and rapid advancement in electronic communication technologies, the problem of spam has become increasingly prominent. The word "spam" originated from a pre-cooked canned meat product called "Spam Ham Hock," which was introduced in 1937. Over time, the word was adopted for electronic mail spam [1]. Spam emails, messages, and other forms of unsolicited information not only disrupt user experience but also pose potential security risks such as phishing and malware distribution. These threats can lead to data breaches, system damage, and even financial losses. In 2023, Kaspersky Lab reported that 45.60% of all emails sent globally were spam, with scam spam being the most common type. They noted an increase in "list linking" or "email bombing" DoS attacks, where cybercriminals register victims' emails on numerous legitimate websites, flooding their inboxes with confirmation notifications [2]. This places a heavy burden on mail servers, rendering email accounts unusable. Apache Spark, as a fast and general-purpose big data processing framework, offers significant potential for efficiently handling and analyzing large datasets in a distributed environment. Spark has emerged as an invaluable tool in spam detection, significantly enhancing both the accuracy and efficiency of identifying spam emails, thereby improving email security and user experience. This improvement is achieved through the application of various machine learning and data processing methodologies [3]. Machine learning algorithms such as Logistic Regression, Naive Bayes, Decision Trees, and Support Vector Machines (SVM) have proven effective by learning from data patterns to accurately classify spam. Additionally, recent advancements have seen the integration of ensemble methods and sophisticated deep learning techniques like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which excel in managing complex classification tasks. These innovative approaches collectively contribute to more robust and reliable spam detection systems [4]. For example, Labonne and Moran enhanced spam classification accuracy by employing large language models. Similarly, Dada et al. reviewed numerous machine learning methods, noting that SVM and Random Forests performed

exceptionally well in spam detection [5]. In more recent research, JayaLakshmi and Kishore applied Apache Spark to filter email spam, demonstrating the effectiveness of Deep Neural Networks (DNN) and other machine learning methods on large datasets [6]. Additionally, Park et al. introduced a Term Frequency-Topic Inverse Document Frequency (TF-TIDF) model, which improved upon the TF-IDF model, resulting in a more efficient spam classifier [7]. Despite these advancements, the need for scalable and efficient processing remains, especially given the growing volume of email data. This is where Apache Spark's distributed computing capabilities become highly relevant. Additionally, recent studies like those by Li et al. and Shehaby et al. are introducing novel techniques to enhance the robustness and privacy-preserving aspects of spam detection [8, 9].

The main goal of this study is to create a spam email classification system utilizing machine learning methods. Specifically, first, logistic regression is used to model the relationship between the features extracted from email text and spam classification, leveraging its interpretability and effectiveness in binary classification problems. Second, Naive Bayes is employed because of its straightforwardness and effectiveness in managing extensive text data, making it a suitable comparison to logistic regression. Third, evaluate and contrast these models' prediction abilities to ascertain which spam classification technique is the most effective. In addition, the feature extraction process involves tokenizer, stop words remover, and TF-IDF to ensure that only the most relevant textual information is used for model training. In parallel, the models are rigorously evaluated using a range of metrics, including accuracy, recall, precision, and F1 score. The experimental outcomes reveal that Naive Bayes surpasses logistic regression, particularly in overall accuracy and F1 score, highlighting its robustness and effectiveness in classifying spam emails. This indicates that Naive Bayes is better equipped to handle the nuances of spam detection, making it a more reliable choice for this application. The TF-IDF feature weighting also greatly improves the model's capability to identify ham and spam. The practical significance of this study is that it can accurately identify and filter spam, so as to reduce the risk of malicious attacks and improve user experience. Through this research, a better understanding of spam detection can be gained.

2. Method

2.1. Dataset Description and Preprocessing

The Enron email dataset, which is available to the general public on Kaggle, served as the study's dataset. This dataset contains emails from Enron Corporation employees, consisting of approximately 500,000 messages [10]. It includes both ham (non-spam) and spam emails, with labels indicating their respective categories. The dataset is extensively used for research in machine learning, specifically for spam detection. This study only used part of it. This subset contains a total of 11,029 emails, with 2,996 labeled as spam and 8,033 labeled as ham (non-spam). Preprocessing steps include reading the emails, converting the emails into Dataframe, then transforming all text to lower case, removing punctuation and stop words, and tokenizing the text. Additionally, TF-IDF is applied to transform the textual data into numerical features suitable for model training.

2.2. Proposed Approach

The research methodology entails an extensive preprocessing of the email dataset, followed by the training and evaluation of two machine learning models: Logistic Regression and Naive Bayes. The primary objective of the study is to ascertain which model excels in accurately classifying spam and non-spam emails. The overall approach encompasses data preprocessing, feature extraction using TF-IDF, model training, and performance evaluation. The workflow of this study, which illustrates the essential steps, is depicted in Fig.1.

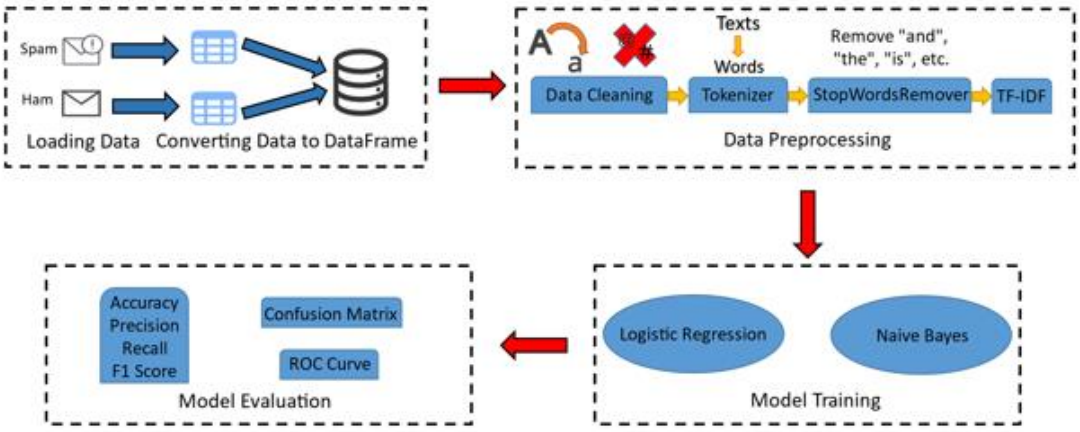

Figure 1. An Overview of the Model Architecture.

2.2.1. Logistic Regression

One effective machine learning technique for addressing classification issues is logistic regression. This linear model is designed for binary classification tasks, assessing the likelihood of an input being spam or ham through the use of the logistic function. By converting input features into a probability value ranging between 0 and 1, the model estimates the likelihood that the given input corresponds to a specific class. This probabilistic approach facilitates a more nuanced understanding of classification, allowing for better differentiation between categories based on the calculated likelihood. The logistic regression model is notable for its robustness, ease of interpretation, and simplicity. Additionally, its ability to provide clear probabilistic outcomes makes it a valuable tool for binary classification problems. Large datasets can be effectively handled by it because of its computing efficiency and ease of implementation. Moreover, it provides a clear probabilistic interpretation of classification outcomes and handles high-dimensional data effectively. The significance of the Logistic Regression model in spam email classification lies in its ability to predict the probability of an email being spam based on the features extracted from its content. Its linear nature allows for straightforward decision boundaries and interpretable results, which are crucial for understanding and improving the model.

A logistic function is applied after a linear combination of input characteristics in the Logistic Regression model to provide a probability. This probability is then used to classify the email as spam or ham based on a threshold, typically 0.5. The logistic function is used in logistic regression:

$$P(z = 1|x) = \frac{1}{1+e^{-(\gamma_0+\gamma_1 x_1+\gamma_2 x_2+\cdots+\gamma_n x_n)}} \quad (1)$$

Where γ_0, γ_1,…, γ_n are the model parameters learned from the data. This function ensures the output is between 0 and 1, representing a probability.

2.2.2. Introduction of Spark

Naive Bayes, a traditional classification method, is grounded in Bayes' theorem and operates under the assumption that features are conditionally independent. This classifier excels in tasks like text classification, including spam detection, because of its straightforward nature and high efficiency. Bayes' theorem offers a mechanism to compute hypothetical probabilities using existing knowledge, making Naive Bayes both practical and effective for these applications. Its simplicity allows for quick implementation and processing, further enhancing its utility in various text-related classification problems. The theorem is stated as follows:

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)\cdot P(\alpha)}{P(\beta)} \quad (2)$$

Where P (α|β) represents the posterior probability of class α when predictor β is given. P (β|α) is the probability of predictor β when class α is given. P (α) represents the initial likelihood of class α occurring. P (β) represents the initial likelihood of predictor β occurring.

In the context of email spam classification, α represents the class (spam or ham), and β represents the features derived from the email content. First, Emails are transformed into feature vectors, where each feature represents the presence or frequency of a word. Then the Naive Bayes classifier is trained on a labeled dataset where emails are tagged as spam or ham. The classifier learns the probabilities of words appearing in spam and ham emails. For the test set, the classifier calculates the probability that the email is spam or ham based on the learned probability and predicts the label accordingly.

2.2.3. TF-IDF

TF-IDF is a statistical measure used to evaluate the significance of a word within a document relative to a collection of documents. This metric is calculated by combining two key components: TF and IDF. TF quantifies how frequently a word appears in a specific document, providing insight into its prevalence within that context. Conversely, IDF gauges the importance of the word across the entire corpus, diminishing the influence of terms that are common across many documents while emphasizing those that are distinctive to particular documents. This dual approach allows TF-IDF to effectively balance term frequency with term rarity, making it a powerful tool for text analysis and information retrieval. By integrating TF and IDF, TF-IDF effectively balances term frequency and rarity, providing a nuanced understanding of word relevance within the context of text analysis, as:

$$TF(x,b) = \frac{f_{x,b}}{\sum_{x' \in b} f_{x',b}} \tag{3}$$

Where $f_{x,b}$ denotes how often term "x" appears in document b. $\sum_{x' \in b} f_{x',b}$ indicates the whole number of all terms within document b. IDF is as follows.

$$IDF(x,B) = log(\frac{N}{1+|b \in B: x \in b|}) \tag{4}$$

Where N signifies the overall quantity of documents within the corpus. $|b \in B: x \in b|$ is the quantity of documents that include the term "x". TF and IDF combine to form TF-IDF.

2.2.4. Apache Spark

Apache Spark is a highly versatile, open-source, and robust unified analytics engine that has been specifically optimized for big data processing. It features a powerful and efficient architecture designed to handle large-scale data computations with ease. Spark's compatibility with multiple programming languages, including Java, Scala, Python, and R, enhances its flexibility and usability across diverse applications. This adaptability, combined with its ability to perform both batch and real-time processing, makes Apache Spark an invaluable tool for data scientists and engineers working on complex data analysis tasks. This study will utilize PySpark, the Python Application Programming Interface (API) for Spark, to conduct data processing and analysis, leveraging Spark's capabilities to handle extensive datasets with ease and speed.

2.2.5. Spark's MLlib

Apache Spark's MLlib is a distributed machine learning library designed for large-scale data processing. It provides a range of machine learning algorithms, including clustering, regression, classification, and collaborative filtering. In the context of spam classification, MLlib can handle vast amounts of email data, processing and analyzing it to build robust spam filters. This integration enhances the progress of scalable and powerful solutions for spam classification.

3. Discussion

The analysis is structured into three main sections: Model Performance Metrics, Confusion Matrix Analysis, and Receiver Operating Characteristic (ROC) Curve Evaluation. Each section discusses different aspects of the models' performance and insights derived from the evaluation metrics.

3.1. Model Performance Metrics

Both the logistic regression and Naive Bayes models are evaluated using the Multiclass Classification Evaluator, focusing on critical metrics such as accuracy, recall, precision, and F1 score. On the test dataset, the logistic regression model achieves an accuracy of 96.18%, whereas the Naive Bayes model reaches a higher accuracy of 98.73%, demonstrating superior performance in classifying emails as spam or ham. To provide a comprehensive understanding of the models' performance, recall, precision, and F1 score are also calculated. The detailed results are presented in Table 1.

Accuracy is calculated by dividing the number of correctly classified instances by the total number of cases. Recall, or sensitivity, measures the proportion of correctly identified positive cases among all actual positive cases, reflecting the model's ability to capture true positives. Precision is the ratio of correctly predicted positive samples to the total number of predicted positive samples, indicating the accuracy of the positive predictions. The F1 score, which is the harmonic mean of recall and precision, provides a single metric that balances these two aspects, particularly useful in scenarios with uneven class distributions. This balanced measure ensures that both false positives and false negatives are taken into account, offering a more holistic view of the model's effectiveness.

Table 1. The Performance Metrics of Two Models

Performance Metrics	Logistic Regression	Naive Bayes
Accuracy	96.18%	98.73%
Precision	90.17%	97.54%
Recall	96.71%	97.86%
F1 score	93.32%	97.70%

Specifically, the Naive Bayes model shows superior precision and recall, resulting in a higher F1 score compared to the logistic regression model. These findings imply that the Naive Bayes model is more effective in handling the classification task, likely due to its probabilistic nature which is well-suited for text classification.

3.2. Confusion Matrix Analysis

The confusion matrices for both models are analyzed to understand the distribution of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

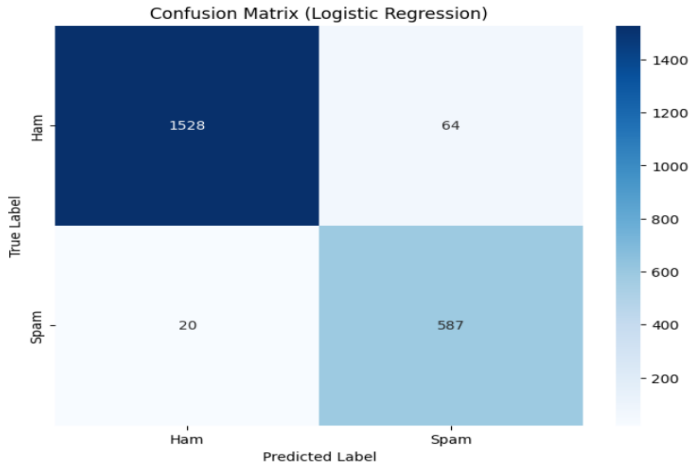

Figure 2. The Result of Confusion Matrix in Logistic Regression.

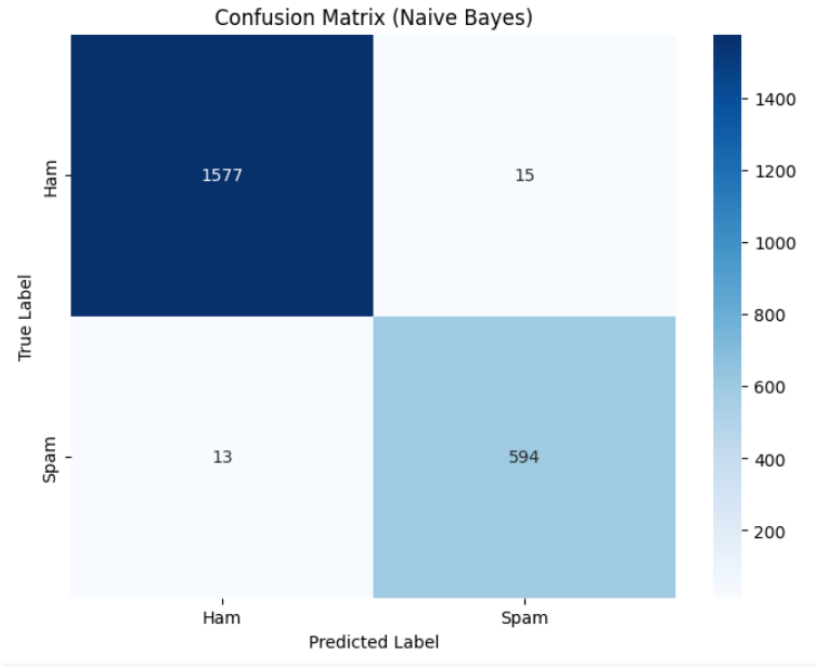

Figure 3. The Result of Confusion Matrix in Naive Bayes.

For the logistic regression model, the confusion matrix (shown in Fig.2) displays 1,528 TNs, 587 TPs, 64 FPs, 20 FNs. This indicates that the logistic regression model has a moderate number of false positives, which are ham emails incorrectly classified as spam.

In contrast, the Naive Bayes model's confusion matrix (shown in Fig.3) reveals 1,577 TNs, 594 TPs, 15 FPs, 13 FNs. Compared to logistic regression, Naive Bayes model has significantly fewer false positives and false negatives, highlighting its superior performance in correctly classifying emails. The reduction in false negatives is particularly crucial in spam detection, as it minimizes the risk of spam emails being misclassified as ham, thereby improving email security.

3.3. ROC Curve Evaluation

The ROC curves are plotted and analyzed to evaluate the models' performance in distinguishing between spam and ham emails. The Area Under the ROC Curve (AUC) is a crucial metric that reflects the models' capacity to differentiate between the two classes.

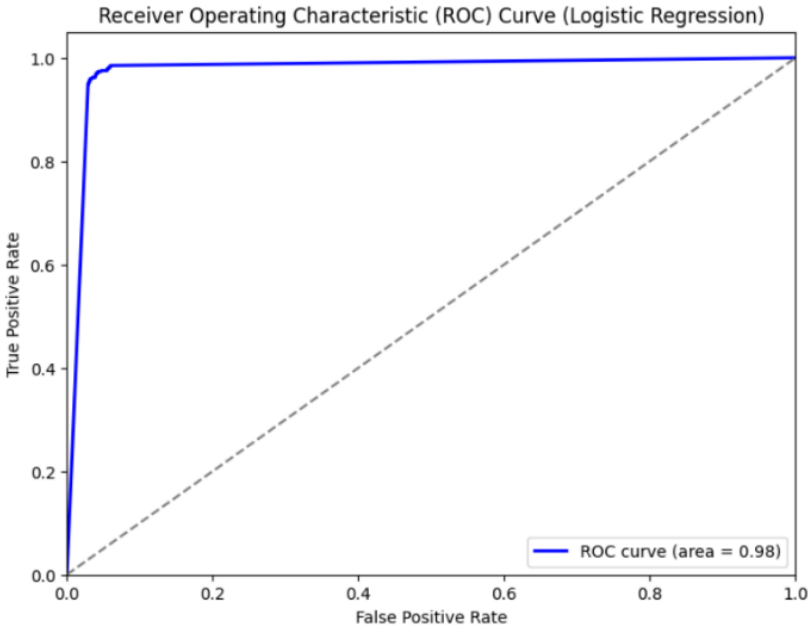

Figure 4. The Curve of ROC with Logistic Regression.

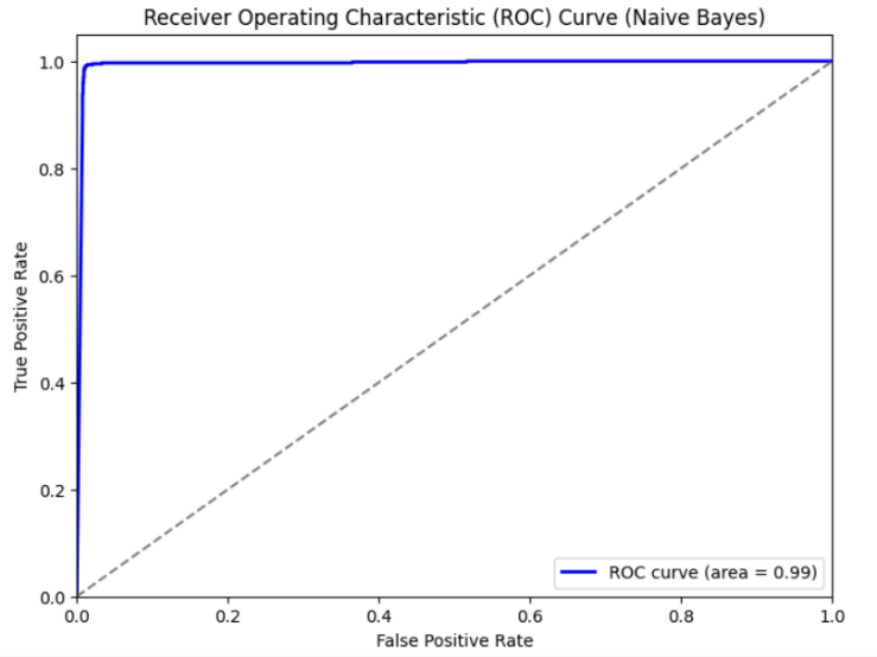

Figure 5. The Curve of ROC with Naive Bayes.

The logistic regression model has an AUC of 0.98, as shown in Fig.4, whereas the Naive Bayes model achieves a higher AUC of 0.99, as shown in Fig.5. This further confirms the excellent performance of Naive Bayes model regarding both sensitivity and specificity. The ROC curve analysis underscores the effectiveness of Naive Bayes in spam detection, providing a robust measure of its discriminative power.

4. Summary

This study introduces a method for classifying spam emails using machine learning methods, specifically logistic regression and Naive Bayes, within the Apache Spark framework. The primary objective is to develop an effective spam detection system that accurately distinguishes between spam and ham emails.

To achieve this, a comprehensive data preprocessing pipeline is proposed, involving text cleaning, tokenization, stop words removal, and TF-IDF feature extraction. Both logistic regression and Naive Bayes models are trained and evaluated using this processed data. The workflow includes loading and preprocessing the email data, splitting it into training and testing sets, training the models, and using criteria like F1 score, ROC curves, accuracy, recall, and precision to assess their effectiveness. To assess these methods, a great deal of experiments are conducted. According to the experimental findings, the Naive Bayes model is superior to logistic regression in every index, making it a more effective choice for spam email classification.

In the future, research on spam detection can focus on several areas: applying adversarial training to enhance model robustness against attacks, developing more interpretable models to understand the decision-making process better, processing multimodal data to handle diverse data types, and creating real-time detection systems to meet the demands of instant communication. These advancements will ensure that spam detection systems remain effective and reliable in an evolving digital landscape.

References

[1] Paswan, M. Kumar, P. Shanthi Bala, and G. Aghila. Spam filtering: Comparative analysis of filtering techniques. IEEE-International Conference on Advances in Engineering, Science and Management, (2012).

[2] J. Doshi, K. Parmar, R. Sanghavi, et al. A comprehensive dual-layer architecture for phishing and spam email detection [J]. Computers & Security, 133 (2023), 103378.

[3] Rao, Sanjeev, A. Kumar Verma, and T. Bhatia. A review on social spam detection: Challenges, open issues, and future directions. Expert Systems with Applications, 186 (2021), 115742.

[4] Labonne, Maxime, and S. Moran. Spam-t5: Benchmarking large language models for few-shot email spam detection. arXiv preprint:2304.01238 (2023).

[5] Dada, E. Gbenga, et al. Machine learning for email spam filtering: review, approaches and open research problems. Heliyon, 5(6) (2019).

[6] A.N.M. JayaLakshmi, and K.V. Krishna Kishore. Performance evaluation of DNN with other machine learning techniques in a cluster using Apache Spark and MLlib. Journal of King Saud University-Computer and Information Sciences, 34(1) (2022), 1311-1319.

[7] Park, WooHyun, N. Muhammad Faseeh Qureshi, and D. Ryeol Shin. Pseudo NLP Joint Spam Classification Technique for Big Data Cluster. Computers, Materials & Continua, 71(1) (2022).

[8] L. Yekai, et al. SpamDam: Towards Privacy-Preserving and Adversary-Resistant SMS Spam Detection. arXiv preprint:2404.09481 (2024).

[9] E. Mohamed, A. Kotha, and A. Matrawy. Introducing Adaptive Continuous Adversarial Training (ACAT) to Enhance ML Robustness. arXiv preprint:2403.10461 (2024).

[10] Information on: http://www.kaggle.com/wcukierski/enron-email-dataset.