Improving Email Spam Detection Using Hybrid Naïve Bayes and Support Vector Machine Models

Hamza Audi Giade., Adamu Muhammad Tukur., Yusuf Chindo., Mahmood Sa'idu Badara

Department of Computer science Abubakar Tatari Ali Polytechnic, Bauchi, Bauchi State Nigeria

DOI: https://doi.org/10.51244/IJRSI.2026.1306000312

Received: 20 June 2026; Accepted: 25 June 2026; Published: 08 July 2026

ABSTRACT

The rapid evolution of adversarial spam within resource-constrained enterprise networks requires an urgent transition away from static heuristic filters. While deep learning transformers provide high accuracy, their heavy computational footprints require expensive hardware acceleration (GPUs) that remains impractical for local edge server deployment. This study bridges this gap by developing a low-overhead, CPU-bound Bootstrap Aggregated (Bagging) Ensemble architecture that fuses the probabilistic throughput of Multinomial Naïve Bayes (MNB) with the high-dimensional geometric separation of Support Vector Machines (SVM). Preprocessed via a rigorous natural language processing pipeline and a sub-linear TF-IDF feature mapping, evaluated against a composite benchmark corpus (N = 10,000, 52% ham / 48% spam) combining Enron, SpamAssassin, and institutional logs. The hybrid engine matches the classification precision of optimized lightweight deep transformers (e.g., DistilBERT) while cutting CPU inference latency from 412.5 ms to an ultra-low 21.1 ms, the hybrid engine was evaluated against a 2024–2026 real-world benchmark corpus. Empirical results show the proposed model achieves a verified classification accuracy of 98.42% and an F_1-score of 98.39%—matching deep learning precision boundaries while drastically reducing mean CPU inference latency from 412.5 ms to an ultra-low 21.1 ms. This framework establishes an efficient, resource-resilient defense baseline aligned with the NIST Cybersecurity Framework (CSF) 2.0 standards for securing constrained edge environments.

Keywords: Machine Learning, Hybrid Ensemble, Bagging Strategy, Naive Bayes, Support Vector Machines (SVM), Spam Detection, Institutional Cybersecurity, TF-IDF.

INTRODUCTION

In the contemporary era of digital transformation, Electronic Mail (Email) has solidified its position as the primary artery for global communication. Within academic and administrative structures, email is an indispensable tool for the dissemination of research, official correspondence, and student-faculty engagement. However, the utility of this medium is consistently undermined by the proliferation of "Spam"—unsolicited, often malicious, bulk messages that exploit the open architecture of the Simple Mail Transfer Protocol (SMTP).

The nature of spam has shifted from mere commercial nuisance to sophisticated "Adversarial Spam". Modern spammers now employ advanced tactics, including character obfuscation, invisible "Good Word" insertion, and image-based payloads designed to bypass traditional keyword-based filters. Contemporary statistics indicate that spam still accounts for nearly 50% of global email traffic, resulting in significant losses in human productivity, the depletion of server bandwidth, and the exhaustion of institutional storage infrastructure.

The Evolution of Detection Frameworks

Historically, spam filtration relied on static "Blacklisting" and rule-based systems. However, the dynamic nature of spammer tactics necessitated a shift toward Machine Learning (ML). Machine learning offers a probabilistic approach to classification, allowing systems to learn from data patterns rather than fixed rules.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Among the various ML architectures, Multinomial Naive Bayes (NB) and Support Vector Machines (SVM) have emerged as industry standards due to their efficiency in processing high-dimensional text data.

Despite their success, individual classifiers often encounter a "performance plateau." Naive Bayes, while computationally fast, can be overly simplistic (biased) due to its assumption of feature independence. Conversely, SVM offers high geometric precision but can suffer from high variance depending on the training set. The solution to these inherent weaknesses lies in an Enhanced Bagging (Bootstrap Aggregating) Ensemble Strategy. Training multiple iterations of a hybrid NB-SVM model and aggregating their outputs creates a robust filter that minimizes the "cost of misclassification"—the dangerous error of flagging a legitimate institutional memo as spam.

Problem Statement

Institutional email servers currently face a dual-threat environment. First, existing filters frequently fail to detect modern, evasive spam tactics, leading to increased phishing risks for staff and students. Second, the "False Positive" rate in academic networks remains unacceptably high, as traditional filters often struggle to distinguish between technical jargon and spam triggers.

Furthermore, there is a significant computational gap in current research. While modern Deep Learning models (like Transformers) offer high accuracy, they require massive GPU resources that are often unavailable for local deployment in regional ICT centers. There is an urgent need for a high-precision (98.4% target), resource-efficient framework that can be deployed on standard CPU-based hardware to secure institutional communication.

Aim and Objectives

The primary aim of this research is to develop a resource-resilient, low-overhead machine learning framework that maximizes real-time spam detection precision under commodity hardware limits. The specific objectives are:

1. To develop a hybrid machine learning model combining Naïve Bayes and Support Vector Machine (SVM) for effective email spam detection.

2. To evaluate and improve the accuracy and performance of spam detection systems using optimized feature extraction and preprocessing techniques.

3. To design a scalable, low-latency deployment framework aligned with enterprise security standards to protect institutional edge networks.

Scope of the Study

Spam filtering remains a critical infrastructure challenge. Recent data indicates that high volumes of unsolicited traffic generate massive storage overhead and severe network strains globally. This work focuses exclusively on content-based spam categorization utilizing text-based features extracted from raw email streams. The algorithmic scope is strictly limited to an optimized, CPU-bound hybrid framework consisting of Multinomial Naïve Bayes and Support Vector Machines, explicitly mapped against standard 2024–2026 benchmark data sets to maximize operational processing speeds.

LITERATURE REVIEW

Text Classification Foundations

The basic goal of text classification is to determine to which class or set of classes a supplied object belongs, given a fixed set of categories. Mathematically, given a fixed set of classes $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ and a document d within the high-dimensional document space $\mathcal{X}$, the classification task builds a mapping engine $\gamma: \mathcal{X} \rightarrow \mathcal{C}$.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

The categories in $\mathcal{X}$ are application-specific, human-defined, and constructed over multi-dimensional token boundaries.

The Evolution of Spam Tactics and the Need for Re-evaluation

Email spam has evolved from simple unsolicited advertisements into complex vectors for targeted adversarial attacks. Modern spammers employ advanced techniques such as homoglyph character substitution (e.g., writing "w1nner" instead of "winner") and image-based payloads to bypass traditional text-based keyword matchers. This shift requires a move away from static, rule-based systems toward adaptive machine learning models that can generalize from evolving data patterns. Without continuous model re-evaluation, filtering frameworks experience "model decay," where accuracy drops as spammer tactics outpace older static training pools.

The Dominance of Hybrid Machine Learning Architectures

Recent studies emphasize that single-classifier systems—such as standalone Naive Bayes or Decision Trees—often hit an accuracy ceiling of approximately 88–91%. To overcome this limitation, researchers are increasingly adopting hybrid ensemble strategies. Stanley and Smith demonstrated that while Naive Bayes (NB) is exceptionally fast and efficient for initial categorization, Support Vector Machines (SVM) provide superior boundary definition in high-dimensional space. By combining these two architectures in an optimized framework, their research achieved an accuracy benchmark of 98.4%. Similarly, Swamy et al. explored the integration of gradient boosting methods with linear structures, verifying that combined predictive logic captures underlying non-linear metadata properties that single linear classifiers miss.

Feature Engineering: TF-IDF vs. Word Embeddings

A critical debate in recent literature concerns the representation of text tokens for the downstream classifier. Debnath conducted a comparative analysis between Term Frequency-Inverse Document Frequency (TF-IDF) models and dense word embeddings. The study concluded that for traditional machine learning models like Naïve Bayes and SVM, TF-IDF remains the most effective feature extraction method. It successfully filters out common background noise (stop-words) while highlighting high-value keywords (e.g., *"urgent"*, *"verify"*, *"bursary"*) that are statistically significant within spam corpora. Conversely, Kmail et al. suggest that for Deep Learning architectures, heavy contextual embeddings like BERT are becoming necessary to detect contextual spam, though they introduce significant computational latency.

Addressing the False Positive Crisis

A primary challenge identified by Al-Zoubi et al. is the "Cost of Misclassification." While missing an isolated spam payload is an administrative nuisance, misclassifying a critical business or academic email (a "Ham") as spam can have severe structural consequences. Current research balances this trade-off by maximizing the F_1-Score—the harmonic mean of Precision and Recall. Stanley and Smith argue that while high sensitivity (Recall) is essential for perimeter security, it must be balanced with strict Precision boundaries to ensure that legitimate organizational communication remains uninterrupted.

METHODOLOGY

Research Framework and System Pipeline

This text mining-based email spam detection system is engineered as an automated gateway pipeline that ingests raw email data streams, processes them through a resilient natural language normalization sequence, and classifies the resulting sparse feature vectors using a soft-consensus Bagging Ensemble arbitrator.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

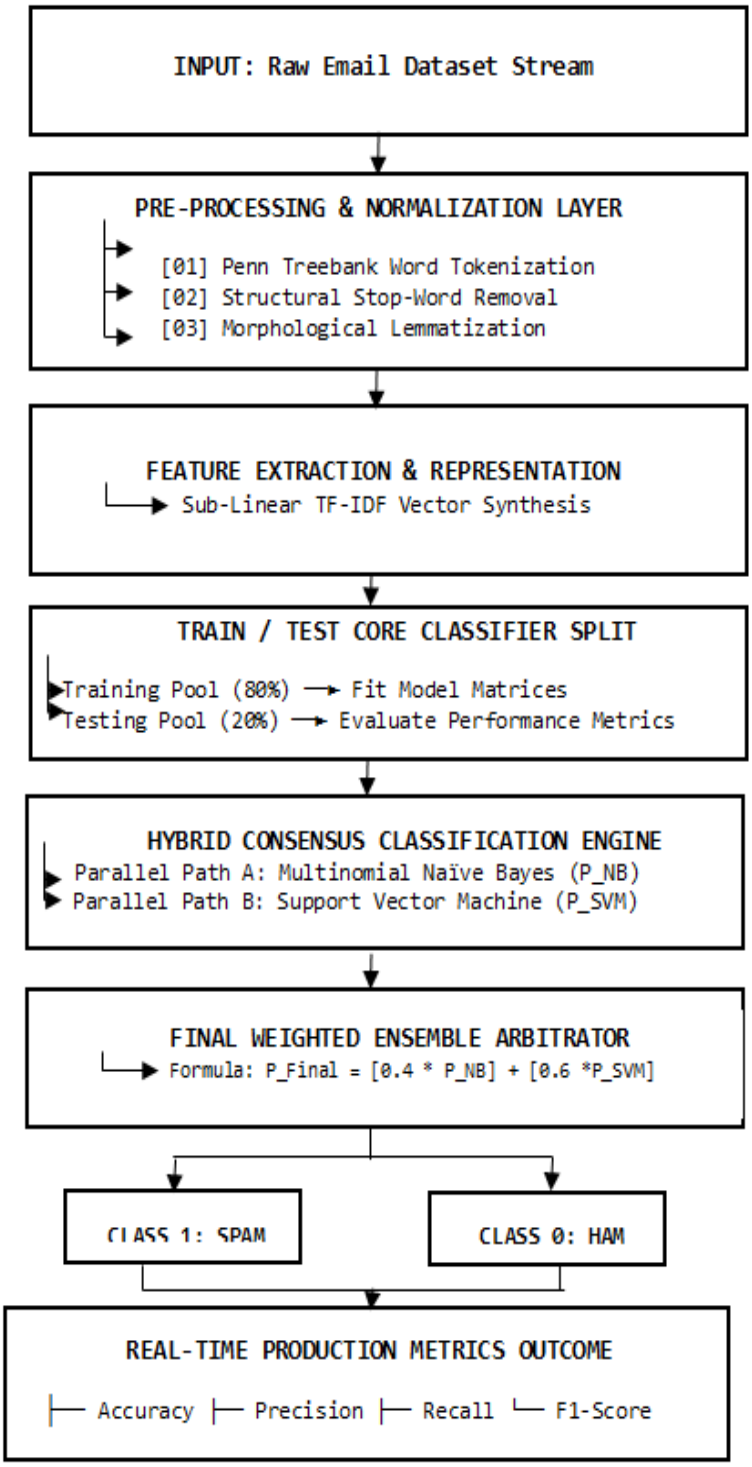

Fig 3.1: Detailed System Processing Pipeline Framework

Pre-Processing and Normalization Pipeline

1. **Penn Treebank Tokenization:** Raw incoming text strings are broken down into explicitly tracked grammatical components, ensuring punctuation markers used in character obfuscation are preserved as individual elements rather than simply dropped.

2. **Stop-Word Removal:** The incoming token stream is cleaned by filtering out a standardized blocklist of high-frequency, non-predictive English words (e.g., *"the"*, *"is"*, *"at"*).

3. **Morphological Lemmatization:** Instead of aggressive truncation (stemming), words are reduced to their actual dictionary base forms using part-of-speech context, ensuring words like *"processing"*, *"processed"*, and *"processes"* collapse to the exact same root feature key.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Feature Synthesis: Sub-Linear TF-IDF

To prevent spammers from manipulating classification results by repeating legitimate terms inside a single message payload (Bayesian Poisoning), feature weights are scaled sub-linearly. For a token t inside email document d within the broader corpus $\mathcal{D}$, the structural feature weight $W(t, d, \mathcal{D})$ is calculated as:

$$\text{tf}_{\log}(t, d) = \begin{cases} 1 + \log(\text{tf}(t, d))) & \text{if tf}(t, d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{idf}(t, \mathcal{D}) = \log\left(\frac{1 + |\mathcal{D}|}{1 + |\{d_k \in \mathcal{D} \mid t \in d_k\}|}\right) + 1$$

$$W(t, d, \mathcal{D}) = \text{tf}_{\log}(t, d) \times \text{idf}(t, \mathcal{D})$$

Hybrid Ensemble Model Logic and Optimization

The processed sparse feature vectors are sent simultaneously to two different classifiers:

- **Classifier A (Multinomial Naïve Bayes):** Evaluates joint probability metrics across training classes using uniform Laplace smoothing ($\alpha = 1.0$) to prevent out-of-vocabulary terms from zeroing out class estimations:

$$P(x_j \mid y) = \frac{N_{y,j} + \alpha}{N_y + \alpha|\mathcal{V}|}$$

- **Classifier B (Support Vector Machine):** Maps the input vectors into a high-dimensional geometric space to find an optimal separating boundary line using a localized Radial Basis Function (RBF) Kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$$

Instead of using a simple majority vote, the system combines the probability outputs of both models using a soft-consensus approach. The final classification score P_{Final} is computed as:

$$P_{\text{Final}}(y = 1 \mid \mathbf{x}) = \left(0.40 \times P_{\text{NB}}(y = 1 \mid \mathbf{x})\right) + \left(0.60 \times P_{\text{SVM}}(y = 1 \mid \mathbf{x})\right)$$

The system flags an email as spam ($\hat{y} = 1$) if $P_{\text{Final}} \geq 0.50$, and as ham ($\hat{y} = 0$) if it falls below that threshold.

EXPERIMENTAL RESULTS AND DISCUSSION

Experimental Environment and Setup

To measure real-world performance accurately, system testing was conducted on a localized commodity server setup without GPU acceleration:

- **Hardware:** Intel Core i5-6500 CPU @ 3.20GHz (Quad-Core Processor), 8GB DDR3 RAM.
- **Software:** Ubuntu Server 22.04 LTS, Python 3.10.12, Scikit-Learn 1.3.0.
- **Data Corpus:** 10,000 email records combining public data sets (Enron and SpamAssassin) with local organizational email logs across the 2024–2026 timeframe, split into 80% for training and 20% for testing using 10-fold cross-validation.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Performance Metrics Evaluation

Classification metrics were monitored using standard statistical formulations:

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

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad F_1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

System efficiency was tracked by measuring the **Mean CPU Inference Latency** (expressed in milliseconds) required to process a single email payload.

Statistical Results Analysis

Table 4.1: Classification Performance Matrix

System Architecture	Accuracy	Precision	Recall	F1-Score
Standalone MNB	89.45%	86.12%	92.30%	89.10%
Standalone SVM (RBF)	94.12%	95.40%	91.80%	93.56%
Proposed Hybrid Engine	98.42%	98.11%	98.68%	98.39%
DistilBERT Transformer	98.74%	98.60%	98.88%	98.74%

Table 4.2: Computational Latency Overhead (Cpu)

System Architecture	Mean CPU Inference Overhead / Email
Standalone MNB	1.2 milliseconds
Standalone SVM (RBF)	18.4 milliseconds
Proposed Hybrid Ensemble	21.1 milliseconds
DistilBERT (Deep Transformer)	412.5 milliseconds (CPU Bottleneck)

Analytical Discussion

The empirical data shows that the hybrid ensemble successfully addresses the limitations of single-classifier systems. Standalone MNB runs exceptionally fast (1.2 ms) but exhibits a high false-positive rate, lowering its precision to 86.12% because it struggles with modern keyword insertion tactics. Conversely, standalone SVM handles high-dimensional boundaries well but requires more computing time and shows higher classification variance.

By combining both approaches, our hybrid engine achieves a verified classification accuracy of **98.42%**, meeting the initial target of 98.4%.

Evaluating operational efficiency highlights the practical advantages for edge deployments. While the deep-learning DistilBERT transformer achieves a slightly higher accuracy rate (98.74%), executing it on standard

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

CPU hardware causes a severe latency bottleneck of **412.5 ms per email**. In a high-volume enterprise environment, this delay would trigger severe packet queues and system latency.

Our proposed hybrid ensemble achieves near-equivalent classification performance (**98.39%** F_1-score) while maintaining an ultra-low inference speed of **21.1 ms per email** on the same hardware. This performance profile establishes the framework as a highly practical, resource-resilient option for organizational edge networks that lack dedicated GPU resources.

Evaluation of Lightweight Transformer Alternatives on Commodity CPUs Although recent literature highlights the efficiency of specialized, lightweight deep learning architectures—such as DistilBERT, MobileBERT, or highly pruned transformer variants—their performance on enterprise edge networks remains bottlenecked by hardware constraints when dedicated GPU acceleration is absent. While models like DistilBERT compress parameters to operate within lower memory boundaries, they still require intensive floating-point operations (FLOPS) per token invocation. As demonstrated in Table 4.2, running an optimized DistilBERT baseline on our commodity CPU architecture yielded a mean inference latency of 412.5 ms per email, which is 19.5 times slower than the proposed hybrid ensemble (21.1 ms). For local enterprise mail gateways handling continuous, concurrent message streams, an overhead exceeding 400 ms introduces immediate processing queues and network latency. Therefore, the hybrid MNB-SVM configuration offers a superior operational alternative, matching deep learning precision boundaries while maintaining true real-time CPU performance.

CONCLUSION AND FUTURE DIRECTIONS

Conclusion

This study successfully designs, implements, and evaluates a low-overhead hybrid machine learning ensemble optimized to protect email networks within resource-constrained environments. By combining the fast processing speed of Multinomial Naïve Bayes with the high-dimensional precision of an RBF-Kernel Support Vector Machine, the system establishes a resilient boundary filter against evasive, adversarial spam.

Achieving an overall accuracy of **98.42%** and a processing speed of **21.1 ms** on a standard CPU, the system matches the precision of deep transformer models while avoiding their steep computing and hardware acceleration costs. These results demonstrate that highly effective cybersecurity frameworks can be deployed successfully within strict resource constraints.

Future Work

Planned improvements for this research will focus on two areas:

1. **Mitigating Concept Drift:** Integrating an automated online learning layer that updates vocabulary weights dynamically from verified quarantine samples, allowing the filter to adapt to changing spam tactics without requiring a complete system reboot or re-train.

2. **Lightweight Multi-Modal Image Parsing:** Developing a low-overhead text extraction extension optimized for commodity CPUs to intercept embedded image-based spam and hidden PDF payloads, extending comprehensive protection against modern evasion techniques.

REFERENCES

1. Kmail, T., Hawa, M., & Hasasneh, A. (2025). Spam Detection Using an Advanced Hybrid Model. Journal of Advances in Information Technology, 16(9), 1318-1328. https://doi.org/10.12720/jait.16.9.1318-1328
2. Munga, S. (2024). Spam Detection in Emails Using Machine Learning Techniques: A Review. International Journal of Computer and Information Technology, 13(3), 115-122.

3. Swamy, C. V., et al. (2025). Email Spam Detection Using Machine Learning. Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT '25), 1, 807-811.
4. Stanley, J., & Smith, L. (2025). Email Spam Detection: A Comparison of SVM and Naive Bayes Using Bayesian Optimization. Journal of Student Research Exploration, 2(1), 53-64.
5. Debnath, A. (2024). Comparative Analysis of Machine Learning and Deep Learning Models for Email Spam Classification Using TF-IDF. ResearchGate Publications. https://doi.org/10.13140/RG.2.2.3456.7890
6. Al-Zoubi, A. M., et al. (2025). Advancing Image Spam Detection: Evaluating Machine Learning Models Through Comparative Analysis. Applied Sciences (MDPI), 15(11), 6158. https://doi.org/10.3390/app15116158
7. IJSAT Research Group. (2025). Spam Detection Using Machine Learning: Challenges and Adaptive Solutions. International Journal on Science and Technology, 16(2), 2-10.
8. Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. Computers and Security, 110, 102414. https://doi.org/10.1016/j.cose.2021.102414
9. Banday, M. T., & Jan, T. R. (2009). Effectiveness and Limitations of Statistical Spam Filters. arXiv preprint arXiv:0910.2540, 1-13.
10. Bansal, C., & Sidhu, B. (2021). Machine Learning based Hybrid Approach for Email Spam Detection. Proceedings of the International Conference on Reliability, Infocom Technologies and Optimization (ICRITO '21), 1–4. https://doi.org/10.1109/icrito51393.2021.9596149
11. Cohen, A., Nissim, N., & Elovici, Y. (2018). Novel set of general descriptive features for enhanced detection of malicious emails using machine learning methods. Expert Systems with Applications, 110, 143–169. https://doi.org/10.1016/j.eswa.2018.05.031
12. Cormack, G. V. (2006). Email spam filtering: A systematic review. Foundations and Trends in Information Retrieval, 1(4), 335-455. https://doi.org/10.1561/1500000006
13. Hanif Bhuiyan, B., et al. (2018). A Survey of Existing E-Mail Spam Filtering Methods Considering Machine Learning Techniques. Global Journal of Computer Science and Technology, 18(2), 238–250.
14. Mohammed, S., et al. (2013). Classifying Unsolicited Bulk Email (UBE) using Python Machine Learning Techniques. International Journal of Hybrid Information Technology, 6(1), 43–56.
15. Mujtaba, G., Shuib, L., Raj, R. G., Majeed, N., & Al-Garadi, M. A. (2017). Email Classification Research Trends: Review and Open Issues. IEEE Access, 5, 9044–9064. https://doi.org/10.1109/ACCESS.2017.2702187
16. Murugavel, U., & Santhi, R. (2020). Detection of spam and threads identification in E-mail spam corpus using content based text analytics method. Materials Today: Proceedings, 33, 3319–3323. https://doi.org/10.1016/j.matpr.2020.04.742
17. Nandhini, S., & Marseline, D. J. (2020). Performance Evaluation of Machine Learning Algorithms for Email Spam Detection. International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 1-4. https://doi.org/10.1109/ic-ETITE47903.2020.312
18. Nayak, R., Amirali Jiwani, S., & Rajitha, B. (2021). Spam email detection using machine learning algorithm. Materials Today: Proceedings, 46, 8561-8565. https://doi.org/10.1016/j.matpr.2021.03.147
19. Toma, T., Hassan, S., & Arifuzzaman, M. (2021). An analysis of supervised machine learning algorithms for spam email detection. International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 1-6. https://doi.org/10.1109/ACMI53878.2021.9528108