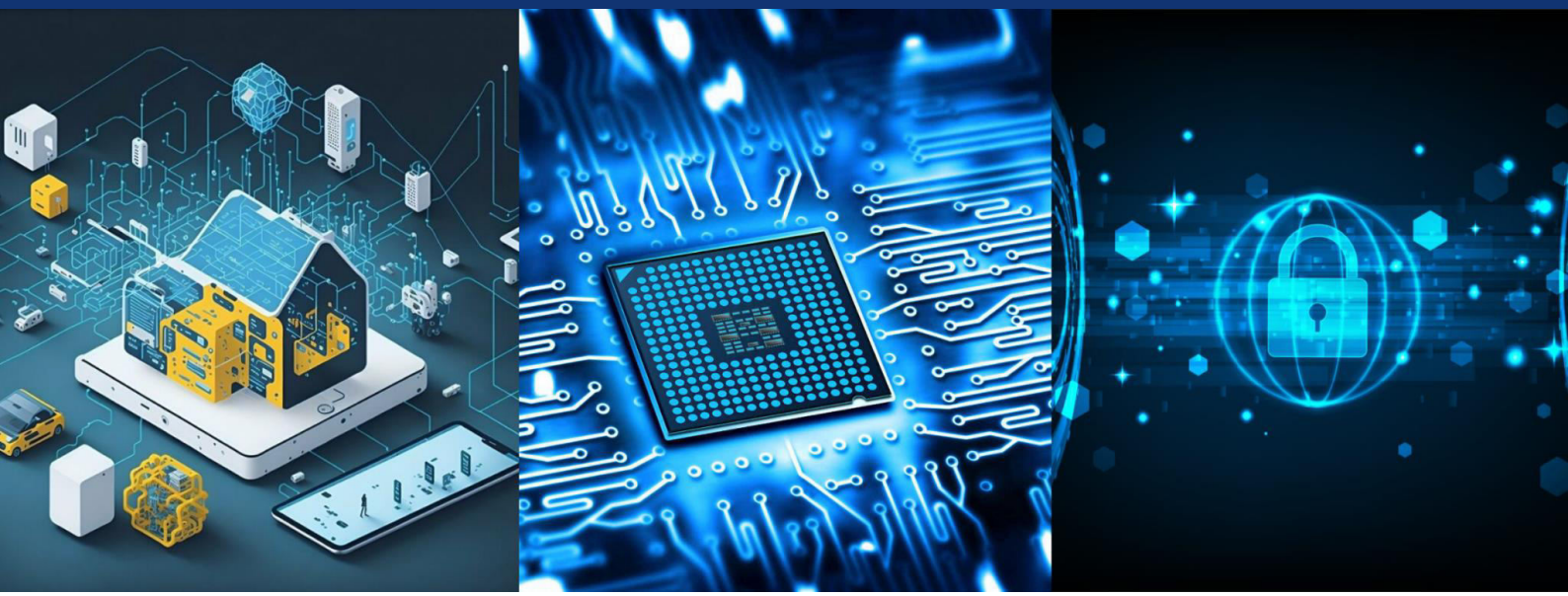
IJIRCCE

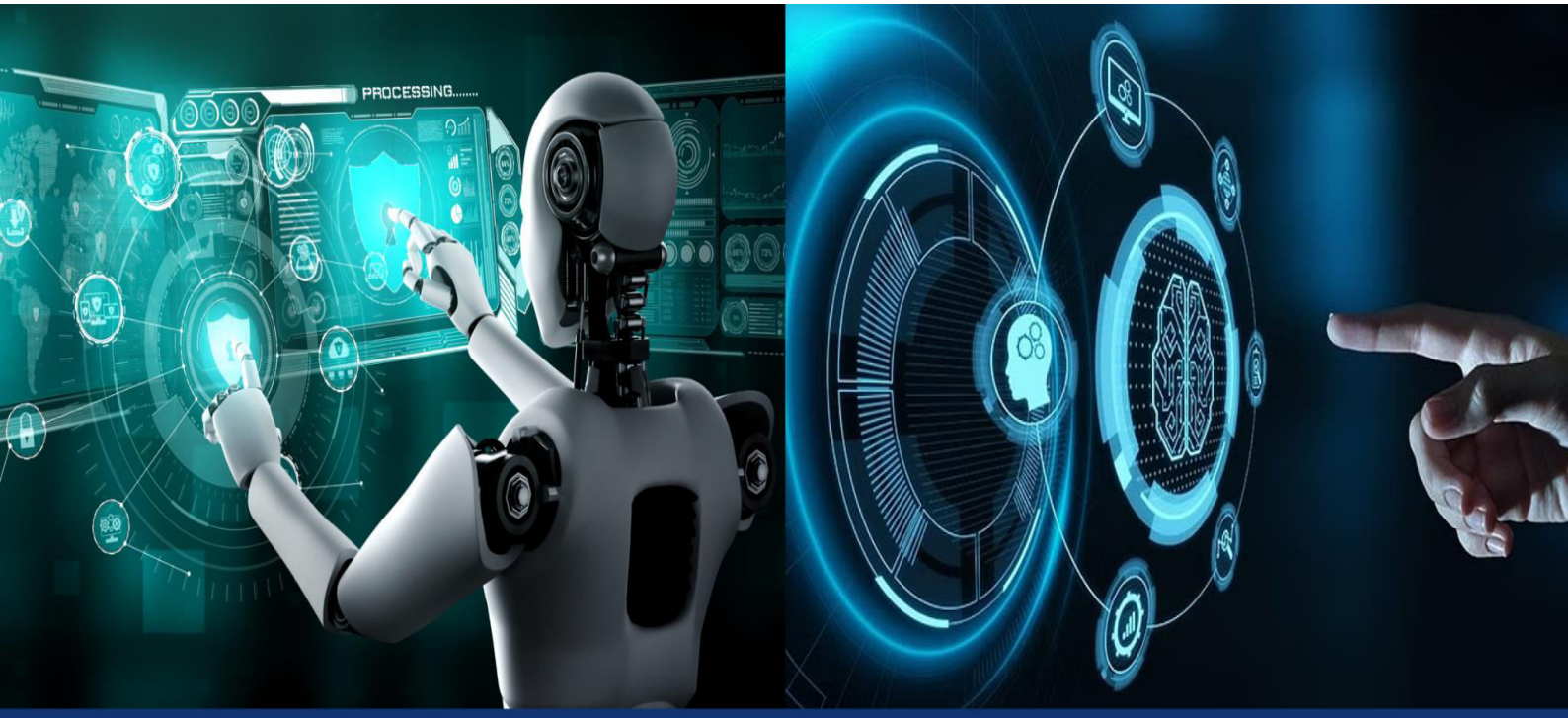
ISSN(O): 2320-9801

ISSN(P): 2320-9798



# International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)




**Impact Factor: 8.771**

**Volume 13, Issue 5, May 2025**

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305077

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

# Real-Time Phishing Email Detection using AWS and Machine Learning: A Serverless Pipeline

**Parth Parab, Yuvan Dayakar, Akash N, Jitesh Kumar M, Dr. K A Varun Kumar**

IV B.Tech., SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

IV B.Tech., SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

IV B.Tech., SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

IV B.Tech., SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

**ABSTRACT:** Phishing emails remain a critical cybersecurity threat, necessitating scalable and real-time detection systems. This paper proposes a serverless pipeline that integrates the Gmail API with Amazon Web Services (AWS) and a tuned XGBoost model for phishing email detection. Trained on a dataset of approximately 120,000 emails, the system leverages AWS Lambda for processing, S3 for storage, and CloudWatch for monitoring, achieving high accuracy in live email classification. Experimental results demonstrate robust performance in real-time scenarios, with email fetching and classification completed efficiently. The serverless architecture offers scalability and cost-efficiency, making it suitable for enterprise adoption. Compared to static or unscalable detection methods, this pipeline provides a practical, cloud-native solution. Future enhancements include deep learning integration and broader email platform support.

**KEYWORDS:** Phishing Detection, Machine Learning, AWS, Email Security, ServerlessComputing, Real-time Systems, XGBoost, Gmail API, Cyber Security, Cloud Computing.

## I. INTRODUCTION

Phishing emails have become the predominant threats to cybersecurity attacking human vulnerabilities for stealing information, planting malware, or conducting financial fraud. As per the 2024 Verizon Data Breach Investigations Report, phishing made up more than 30% of the data breaches while incurring losses in excess of $4.5 billion annually across the globe. Phishing has now come of age while its techniques are evolving, consequently making it difficult to detect-the generic mass emails of the past have become more nuanced targeting a particular individual or company over the past decade with great success in deception. Nowadays phishing emails follow legitimate communications and utilize social engineering tricks against users into clicking on malicious links or uploading credentials. A study in 2023 noted a surge in spear-phishing attacks by 20%, especially in the finance and healthcare sectors, with attackers using stolen data to send out highly personalized messages [2].

The rise of polymorphic phishing emails changing their structure to evade detection among the newer advanced techniques ever harder to characterize. They often get past traditional rule-based filters relying on static signatures which cannot accommodate zero-day attacks [3]. Besides, the increasing volume of email traffic in the enterprise environment that stalled is at over 300 billion emails sent across the globe each day is posing threats to scalability for real-time detection systems [4]. This search for adaptive, scalable, and efficient solutions has sparked interest in machine learning-ML-based approaches promising to identify phishing patterns with high accuracy [1], [2].

But the challenge many ML models face is training on static datasets and hence lack the ability to process live email streams in real timem which hinder the deployment practically [5].

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305077

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

## II. RELATED WORK

The technique of machine learning (ML) has made its advancements towards phishing detection at a higher rate in the past few years. Gupta et al. suggested methods combining natural language processing (NLP) with feature engineering for phishing email detection. The methodology attacked the problem using natural language processing features from email content but could not develop systems that are real-time deployable. Similarly, [2]Sahingoz et al. undertook a comparative study of ML and deep learning (DL) models, finding that XGBoost achieved about 98% benchmark accuracy for researching datasets.

However, they did not consider real-time scenarios and did not address live email processing. Approaches based on deep learning have been used for phishing detection. Sharma and Singh assessed methods like TF-IDF and [5]AdaBoost, getting quite strong accuracy scores from them.

## III. PROPOSED METHODOLOGY

**The two principal aspects in this proposed AI-based phishing detection system are:**

- Model Training Module: It provides the essential requirements for dataset preparation, feature extraction, and training of the machine learning model for phishing detection.
- System Architecture Module: It is providing serverless pipeline of real-time email processing, classification, and monitoring.

Combining these two, this architecture would provide a complete end-to-end framework for real-time phishing detection in a cloud-native environment.

**Model Training Module**

This module will comprise steps to prepare the email dataset, extract features, and finally train a machine learning model relating to phishing detection.

**Dataset and Preprocessing:**Five public datasets from Kaggle and Hugging Face were combined into a dataset of approximately 120,000 emails. The final dataset totals 60,600 safe emails and 59,400 phishing emails after preprocessing. Figure 1 shows the distribution of phishing and safe emails across the datasets.

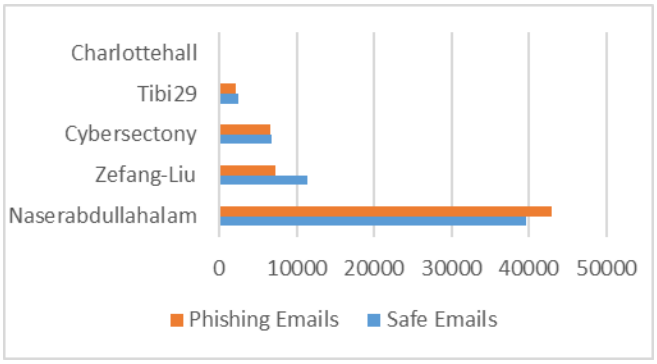



Figure 1: Distribution of Phishing and Safe Emails per Dataset

Preprocessing includes dropping the receiver and date columns. Rows with the null sender or subject value are filled with "no sender" and "no subject" respectively, while rows with null body value were completely dropped. The collection primarily consists of emails in English and the very limited number of spear phishing samples were also segregated into training and testing sets (in the ratio of 80:20).

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305077

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| XGBoost | 0.975 | 0.961 | 0.990 | 0.975 |
| Logistic Reg. | 0.958 | 0.959 | 0.958 | 0.958 |
| SVC | 0.955 | 0.956 | 0.955 | 0.955 |
| KNN | 0.952 | 0.952 | 0.952 | 0.952 |
| Random Forest | 0.951 | 0.952 | 0.951 | 0.951 |
| Naïve Bayes | 0.932 | 0.932 | 0.932 | 0.932 |
| Adaptive Boosting | 0.899 | 0.903 | 0.899 | 0.899 |

## IV. SYSTEM ARCHITECTURE MODULE

The proposed system architecture, depicted in Figure 2, leverages the Gmail API and AWS serverless services to enable real-time phishing email detection. The pipeline processes emails in a sequential flow, starting with email monitoring, followed by fetching, storage, classification, and monitoring. Each component is detailed below, following the data flow from email arrival to detection and analysis.

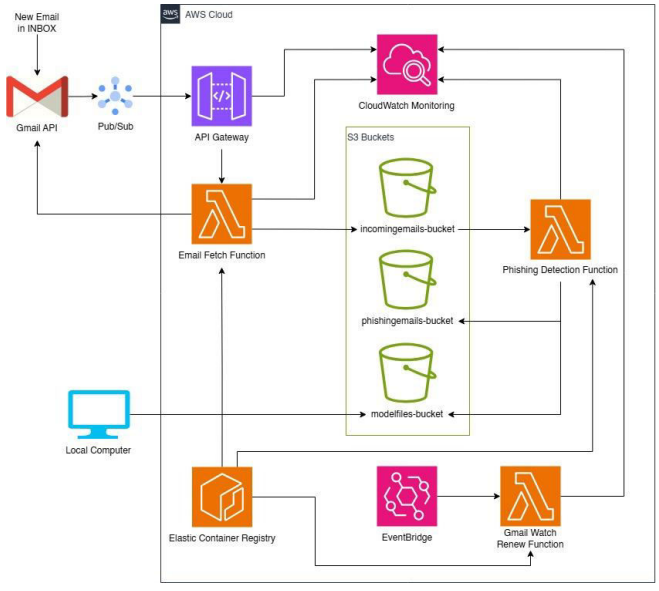



Figure 2: System Architecture Diagram

**Gmail Api:**The pipeline begins with the Gmail API, which monitors the inbox of the user for new emails. The users.watch method is employed to watch the INBOX label, generating a historyId (e.g., 9341) for each change in the mailbox state.

**Google Cloud Pub/Sub:** Upon detecting a new email, the Gmail API sends the historyId to a Google Cloud Pub/Sub topic. This topic acts as a messaging queue, decoupling the Gmail API from the downstream AWS infrastructure.

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305077

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

**Email Fetch Lambda Function**: The email-fetcher Lambda function, invoked via the API Gateway, processes the Pub/Sub message containing the historyId. It uses the Gmail API's users.history.list method to retrieve new email metadata and content, downloading the email as an .eml file (e.g., email_195efefc501fb355.eml).

**S3 buckets:** Two S3 buckets manage email storage, s3://incomingemails-bucket/ for incoming emails and s3://phishingemails-bucket/ for emails classified as phishing. The email-fetcher function uploads new emails to incomingemails-bucket, triggering the phishing-detector Lambda function via an S3 event notification.

**Phishing Detection Lambda Function:** The phishing-detector Lambda function, triggered by S3 events from incomingemails-bucket, performs phishing detection. It downloads the email and model files, caching them in /tmp to optimize performance.

**Experimental Results and Analysis**

This section presents a detailed analysis of the system's performance across offline and live testing scenarios.

Offline Evaluation

XGBoost achieved 97.5% accuracy (post-tuning from 99.3% due to overfitting), with precision, recall, and F1-score all at 0.967. The confusion matrix ([11385, 626], [173, 11688]) indicates balanced performance. Figure 3 and 4 shows the training and validation accuracy/loss curves for XGBoost, demonstrating the effectiveness of hyperparameter tuning in reducing overfitting.

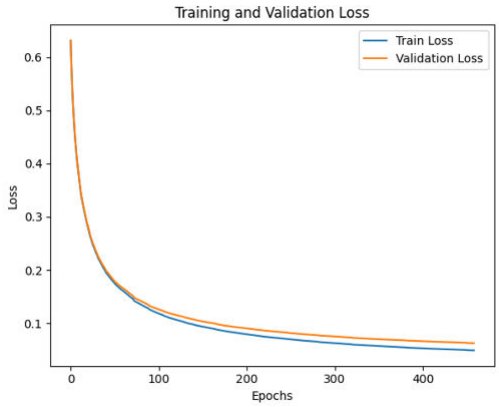



Figure 3: XGBoost Training and Validation Loss

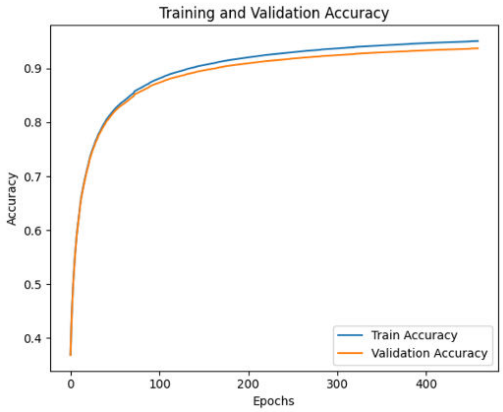



Figure 4: XGBoost Training and Validation Accuracy

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305077

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

**Feature Importance Analysis**

To understand the model's decision-making process, feature importance was analyzed using XGBoost's built-in feature importance scores. Figure 5 lists the top fifteen features contributing to phishing detection.

**Live Pipeline Evaluation**

Live testing processed 100 emails, achieving 96.0% accuracy (96 correctly classified) with an average fetch time of 6349ms and an average detection time of 557ms.

| | Minimum (ms) | Average (ms) | Maximum (ms) |
|---|---|---|---|
| **Email Fetch** | 6281 | 6349 | 26172 |
| **Phishing Detection** | 20 | 557 | 1019 |

**Error Analysis**

An error analysis was conducted to understand the model's misclassifications. Of the 100 emails tested in the live pipeline, four were misclassified: two false positives (safe emails flagged as phishing) and two false negatives (phishing emails classified as safe). The false positives were emails with promotional content containing terms like "urgent" and multiple URLs, which the model mistook for phishing patterns. The false negatives were spear-phishing emails with minimal URLs and highly personalized content, highlighting the dataset's limitation in capturing such samples.

**Observations**

XGBoost's tuned performance aligns with literature,while real-time processing (557ms detection) enables practical deployment. The slight drop from offline (97.5%) to live (96.0%) accuracy suggests minor distribution shifts in live emails, consistent with findings in real-time phishing detection studies.

## V. DISCUSSION

The pipeline's 97.5% offline and 96.0% live accuracy demonstrate robust phishing detection, competitive with prior work. The serverless architecture offers significant benefits:

- Scalability: Lambda auto-scales with email volume, handling spikes (e.g., 1000 emails/day) without manual intervention, as supported by studies on serverless computing.
- Cost-Efficiency: Free-tier usage minimized costs, with enterprise scaling estimated at $0.20 per million Lambda invocations and $0.023 per gb S3 storage, making it cost-effective for large organizations.
- Flexibility: Decoupled components (e.g., S3, Lambda) allow easy updates, enabling rapid adaptation to new phishing threats.
- Reliability: CloudWatch monitoring ensures system health, with fault tolerance mechanisms (e.g., retries, versioning) ensuring continuous operation.

The feature importance analysis provides insights into phishing patterns, confirming that URLs and email length are strong indicators, as noted in prior work [1]. However, the system's limitations include the dataset's English-only focus and limited spear-phishing samples, which led to false negatives in live testing. Live testing also showed fetch delays (6349ms), addressable with direct Gmail-to-S3 uploads, as suggested in cloud-based email processing studies [5]. Future improvements could focus on multilingual support and advanced NLP techniques to better handle spear-phishing emails.

## VI. CONCLUSION

This study presents a practical and scalable solution to address modern email security challenges by integrating machine learning techniques for real-time phishing detection. The proposed system bridges the gap between static ML models and dynamic, cloud-native deployment by leveraging AWS serverless infrastructure and Gmail API integration.The pipeline, encompassing email preprocessing, feature extraction, and real-time classification, achieved 97.5% accuracy in offline testing and 96.0% in live scenarios, processing emails with an average fetch time of 6349ms and classification time of 557ms. The system demonstrated efficient real-time performance, scalability, and reliable monitoring, making it a strong candidate for enterprise email security.

Beyond performance, this work emphasizes real-time applicability, ease of integration, and data-driven decision support, offering a blueprint for future cybersecurity applications in smart enterprise ecosystems.

## VII. FUTURE WORK

Future enhancements include:

- Direct Gmail-to-S3 Uploads: To reduce fetch latency.
- Amazon SageMaker Integration: For scalable model training.
- Deep Learning Adoption: Exploring BERT to improve accuracy and performance.
- Periodic Retraining: Automating model updates with new email data.
- AWS SES Integration: Replacing Gmail API to simplify the architecture

Looking ahead, several enhancements are envisioned to further improve the system's performance, scalability, and adaptability. One significant improvement involves replacing the current Gmail history polling mechanism with a direct Gmail-to-S3 upload path. This change would reduce latency by minimizing intermediate steps, allowing emails to be ingested and processed more quickly. It would also streamline the architecture by removing the reliance on Gmail's history ID tracking, which introduces delays and rate limitations under high-load conditions.

## REFERENCES

1. B. B. Gupta, Aakanksha Tewari, Ankit Kumar Jain, and Dharma P. Agrawal. 2017. Fighting against phishing attacks: state of the art and future challenges. Neural Comput. Appl. 28, 12 (December 2017), 3629–3654. https://doi.org/10.1007/s00521-016-2275-y
2. Sahingoz, Ozgur & Buber, Ebubekir & Demir, Onder & Diri, Banu. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications. 117. 345-357.
3. Gangavarapu, Tushaar & Jaidhar, C. & Chanduka, Bhabesh. (2020). Applicability of machine learning in spam and phishing email filtering: review and approaches. Artificial Intelligence Review. 53. 10.1007/s10462-020-09814-9.
4. Apruzzese, Giovanni & Colajanni, Michele & Ferretti, Luca & Marchetti, Mirco. (2019). Addressing Adversarial Attacks Against Security Systems Based on Machine Learning. 1-18. 10.23919/CYCON.2019.8756865.
5. Sharma, Bhawna & Singh, Parvinder. (2022). An improved anti-phishing model utilizing TF-IDF and AdaBoost. Concurrency and Computation: Practice and Experience. 34. 10.1002/cpe.7287.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details