AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
E-ISSN: 3069-0110
Vol.4, No. 2(2025)
www.ajmimc.com

Email Security Threat Detection

[1] Dr. YASASWINI VANAPALLI , [2] M.Jyothi Sravya , [3] M.Namratha , [4] N.Anitha

[1] Associate Professor, Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India

[2,3,4] Students , Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India,

[2] Email:jyothisravya@gmail.com , [3] Email: namrathamyaka22@gmail.com , [4] Email :anithanenavathanitha76@gmail.com

***Abstract*—** Email has turned into one of the major avenues through which hackers initiate cyber-attacks, mainly phishing, a situation in which attackers deceive users into giving up confidential information. It is absolutely necessary to detect such harmful emails as soon as possible to alleviate the cases of data breaches and financial losses. This is a study of a machine-learning-based method to recognize phishing emails by using content as well as attachments analysis. The system proposed derives the characteristics related to language and structure of the email from the use of Natural Language Processing (NLP) techniques like tokenization, stop-word removal, and frequency analysis. Random Forest classifier is trained on a labeled dataset of legitimate and phishing emails so as to be able to assess the threat level of new messages. The model scores are quite good in terms of accuracy, precision, and recall, thus ensuring that the false alarms are kept at a minimum.

Keywords— (Email Security, Phishing Detection, Machine Learning, Natural Language Processing (NLP), Cybersecurity, Spam Classification, Feature Extraction, Email Threat Analysis, Malicious URL Detection)

I. INTRODUCTION

Email is still one of the top communication tools mainly for private and business use. But one of the reasons for its success is the fact that it has become the most attacked by the cybercriminals, and they are using it to send phishing, spam, malware, and identity theft attacks. In particular, phishing emails are the ones to make the recipients believe they should give the information asked, such as passwords, financial data, or personal identifiers. In fact, the attacked accounts look real, thus it is hard to make a difference between the two, and these types of deceits put in jeopardy not only the individuals but also the organizations.

Normal mail filtering systems use fixed instructions and blacklists. The blacklists only containing e.g. known phishing sites, addresses, etc., are often not effective enough to fight against the criminals as they keep finding new ways while the rules remain the same and phishing tricks evolve. To solve such a problem, there have already been released systems for detecting email security threats that imply advanced tactical measures like machine learning, natural language processing (NLP), and heuristic analysis. They scrutinize the text of the message, the sender's info, the links, and the attachments as well as other user habits and activities to locate dangers. These devices are able to detect the hackers' mails in advance and thus financial loss through fraud, infiltration of networks, market losses due to leaking of valuable information, etc. become avoided.

This is the email threat detection project aimed to build a sound response system that is capable of automatically sorting the electronic messages into real ones or phishing ones. Apart from enhancing email security, the tool would be acting like a user education platform in which the users learn to identify the suspicious content and thus the cybercriminals win less often.

II. Related Work

The problem of phishing detection through email has been a subject of research for more than 20 years. The research evolved from using heuristic-based rules to the current use of advanced machine learning and deep learning techniques. Initially, blacklists and manually defined keyword rules were the primary tools of phishing filters to identify malicious content. Although such measures afforded some level of security, they did not recognize new attacks or messages using obfuscated URLs and complex social engineering tactics.

Machine learning algorithms such as Naïve Bayes, Decision Trees, Logistic Regression, and Support Vector Machines (SVM) were introduced by the researchers to increase the performance of the filters. The models use statistical and linguistic patterns in email text, URLs, and headers. For example, Aburrous et al. (2010) developed an intelligent phishing detection system that used classification mining techniques and achieved higher accuracy than the existing filters. Islam and Abawajy (2013) also presented a multi-tier phishing detection and filtering framework that used header-based, URL-based, and content-based features, thus indicating the advantage of hybrid architectures.

The use of ensemble learning techniques such as Random Forest and Gradient Boosting has brought about further improvement in phishing detection precision by combining the results of several weak learners. Mishra and Sharma (2022) argued that due to its ability to manage non-linear relationships, mixed data types and noisy input, Random Forest models are superior to individual classifiers. Besides, ensemble models deliver interpretability through feature importance that facilitates cybersecurity analysts in knowing the attributes which most influence phishing classification.

When deep learning became popular, researchers started to look at the potential of Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and transformer-based architectures like BERT for the analysis of email and URL content.

Received: 05-05-2025 | Accepted: 12-06-2025 | Published: 19-06-2025 |

III. System Design and Methodology

Email Security Threat Detection System helps users to be safe from bad emails by automatically detecting phishing, spamming, and malware attacks. To achieve this goal, the platform leverages methods from natural language processing (NLP), data mining, and machine learning to scrutinize incoming mails and label them as either legitimate or harmful. The aim is to develop a performant and scalable detection system that requires a minimal amount of human intervention when it is integrated with the existing e-mail frameworks.

1. System Architecture

The system architecture is modular that offers the system flexibility and scalability. The architecture is composed of the layers that are described below:

Input Layer: Emails to be checked are fetched from datasets or real-time mail servers.

Preprocessing Layer: Data is cleaned and prepared by discarding the unneeded elements and standardizing the content.

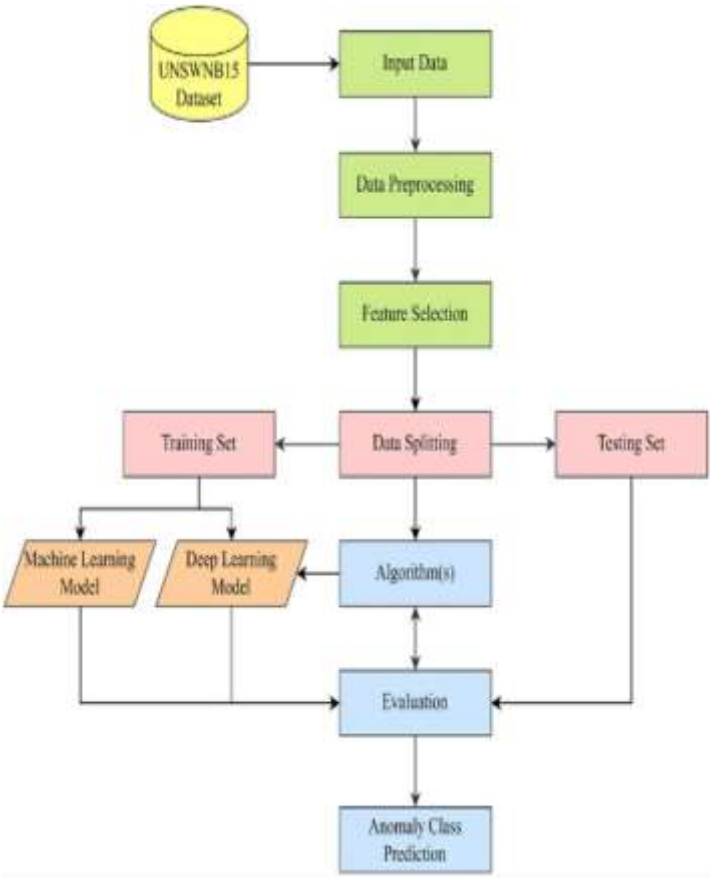
Feature Engineering Layer: Gets the correct features for fraud detection from email content, headers, URLs, and attachments.

Classification Layer: Employs the trained machine learning models to determine the email type (safe or threat).

Decision Layer: Makes the final decision about legitimacy, phishing, spam, or malware, and outputs it.

User Interface Layer: Shows the detection results and logs the alerts to the user or system administrator.

Such a structure of layers enables upgrades to be carried out independently and efficient troubleshooting.



Received: 05-05-2025 | Accepted: 12-06-2025 | Published: 19-06-2025 |

2. Data Collection and Preprocessing

To perform accurate identification, the employed system should contain large datasets of correct and harmful emails. The system makes use of several publicly available data sources, including the Enron Email Dataset, SpamAssassin, and Ling-Spam corpus. These sets provide thousands of sample cases of spam, phishing, and regular business communication that have been taken from the real world.

3. Feature Extraction and Selection

Feature extraction is the most crucial part of this system, as it determines how the model understands and distinguishes between good and bad emails. The method concentrates on the first three feature groups-text features, link features, and behavior features.

Text-Based Features:

Frequency of different types of suspicious words, for example, 'urgent,' 'password,' 'bank,' or 'click here.'Proportion of capital letters or exclamation marks. Amount of misspelling or misleading domain names. Length of subject lines and body text. Link and URL Features Number of links found in the message body. IP address is used instead of domain names in links. Domain age and domain trustworthiness. The shortened or obfuscated URLs (such as bit.ly, tinyurl) are checked for the presence or absence.

Attachment Features:

The kind and size of the file (for example, .exe, .zip, .pdf).
The file includes embedded macros or script.
MIME type inspection for malware and attack vectors.
Header and Sender Features
Mail headers of authentication (SPF, DKIM, DMARC).
Sender IP trustworthiness.

After the feature extraction stage, the system uses different techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings (Word2Vec, GloVe) to convert text features into numerical vectors that machine learning models can understand.

4. Model Training and Classification

At this point, the data that has been preprocessed and had features extracted from it is utilized to train a number of machine learning algorithms. The models employed are:
Random Forest (RF): An ensemble model of high integrity that combines numerous decision trees to provide a classification that is both accurate and reliable.
Support Vector Machine (SVM): By generating the optimal hyperplane, it is very effective in separating the legitimate class from the malicious one.
Logistic Regression (LR): A straightforward yet powerful algorithm that is capable of performing binary classification.
We use 80% of the dataset for training and keep the remaining 20% for testing. To avoid overfitting and to check the generalization capacity of the models on the new set of emails, k-fold cross-validation is applied.

Model training entails:
Inserting the features that have been taken from the data into the model.
Changing the weights and parameters through the use of an optimization technique.

Model prediction performance is checked and the model is refined accordingly.

5. System Deployment

An actual email setting is where the trained model locates its use with the intervention of APIs. A web-based interface is developed using technologies such as Flask or Django where email scanning takes place automatically and in real-time. New emails, as they come, are open for analyzing by the model which can be running in the background.

Scalability: The performance of the system when dealing with a large volume of emails at the same time.

Real-time detection: The system's capability of generating alerts at user end within a very short period of time.

Automatic updates: It needs to be regularly retrained with fresh data so that it can always be updated with the latest threats.

IV. Implementation Details

The proposed system is an automated solution that recognizes and impedes the transmission of harmful emails like phishing, spam, and those containing malware. As a consequence of email being one of the main communication tools for both individuals and organizations, attackers have turned it into their primary medium for spreading malicious links, stealing data, or distributing malware. Existing spam filters are not always up to the task of detecting these sophisticated threats because, in the case of modern phishing attacks, the perpetrators use nicely crafted templates and create fake domains that almost identically resemble the real ones. To address this problem, the proposed system implements machine learning methods for the detection and the subsequent categorization of the suspicious email which results in the user being unaware prior to receiving such email.

System Overview:

The proposed model is a systematic process that involves data collection, data preprocessing, feature extraction, model training, classification, and detection phases. The system is based on the datasets of emails taken from the real world which are a mix of the legitimate and the malicious ones. The datasets used are already labeled so that the models can learn the difference between safe and unsafe messages. The ultimate objective of this system is to detect the threats with high accuracy and at the same time reduce the number of false positives, thus making the system viable and efficient in real life scenarios.

Data Collection and Preprocessing:

The beginning of work is to acquire data on e-mails from a variety of open data sets like the Enron dataset, the SpamAssassin corpus, and phishing email datasets that can be found on the internet. Every email is complete with the fields such as information about the

Received: 05-05-2025 | Accepted: 12-06-2025 | Published: 19-06-2025 |

sender, subject line, body content, hyperlinks, and attachments. Before training, the emails are being cleaned with the aim of that unnecessary data will be removed. The data preprocessing stage involves:

Cleaning the text from the noise that is represented by unwanted symbols, stop words, and special characters.

Making all the text lowercase for standardization.

When a text is tokenized it is essentially splitted into words that are meaningful and are known as tokens. The process of getting URLs from a text and then checking them for being suspicious, for instance, if the links are shortened or if the domains have numbers.

Removing duplicate emails and empty fields to maintain data quality.

This stage guarantees that only clean and relevant data are allowed for feature extraction and model training.

Feature Extraction:

After the data have been preprocessed, various feature types are derived from each email. These features are the main pointers that enable the machine learning models to figure out whether an email is harmful or not. The leading features are:

URL Features: Number of URLs, presence of IP addresses in the link, and domain length.

Sender Features: Email domain of the sender, reputation, and domain age.

Content Features: Occurrence of suspicious words like “urgent,” “verify,” or “click here,” the number of special characters, and the ratio of text to links.

Attachment Features: The type of file, the size of the file, and whether the file has an executable extension like “.exe” or is a macro-enabled “.docm” files.

Header Features: Reply-to mismatch, BCC presence, and unusual sending time.

These features are later translated into numbers or vectors which can be processed by machine learning algorithms.

Model Training and Classification:

On completion of feature extracting, the data are separated into training and testing sets. With the training data models are given instructions on how to spot patterns that are typical of attacks, whereas with the testing data their performance on new data can be evaluated.

The system makes use of three machine learning methods:

Random Forest: It is an efficient method that is suitable for handling large datasets and it deals with the problem of overfitting by aggregating multiple decision trees.

Support Vector Machine (SVM): It is a method that works well with high-dimensional data and it helps to clearly separate the different classes by a boundary.

Logistic Regression: It is a very simple but powerful tool that is used in cases of binary classification such as “safe” or “unsafe.”

Without exception, algorithms in question undergo training and testing, and the performance is gauged by criteria such as accuracy, precision, recall, and F1-score. The test result tells that the Random Forest model has the highest accuracy among the models that were tested.

Detection and Response:

After the training process, the system could work as a daily email filtering tool or be compatible with the current email clients. When an email is newly received, it is subjected to the same steps of preprocessing and feature extraction. A decision is then made by the trained model whether the email is safe or not.

The emails that are safe will be delivered directly to the inbox. If the message is deemed to be a suspicious one, it will be marked or changed to a folder where the emails that are kept temporarily (quarantine) are stored.

When the situation is an absolutely harmful email, it will not be allowed and the administrator will get a notification about it.

This methodology is helpful in keeping users safe from unintentionally clicking on links of phishing or opening malicious files.

System Updates and Maintenance:

The model can be reloaded from time to time when new dangers arise. Besides, the system is available for user feedback—if a user considers an email as spam and marks it accordingly or if a user thinks an email is safe and again marks it, then such data can be used to retrain the model and thus enhance future detection precision. This feedback mechanism enables the system to accommodate the latest forms of attacks and, hence, it keeps on getting better over time.

Conclusion:

The email threat detection system that is being proposed provides a feasible and efficient manner of securing emails by employing machine learning. The main idea behind it is to combine various algorithms and analyze different features of emails, thus, allowing the system to spot the malicious emails more accurately than traditional rule-based filters. The issue in question is solved by the proponents' model, which is compact, flexible, and can easily be merged with the existing email systems, hence, it turns into a mighty weapon against phishing and malware attacks.

Hardware Components

Component	Specification
Processor	Intel Core i5 or higher
RAM	Minimum 8 GB (Recommended: 16 GB)
Storage	500 GB HDD or 256 GB SSD
Graphics Card	Optional – NVIDIA GTX series or higher

Received: 05-05-2025 | Accepted: 12-06-2025 | Published: 19-06-2025 |

Display	15-inch monitor or larger
Peripherals	Keyboard, Mouse, Internet Connection

Software Requirements

Component	Specification
Operating System	Windows 10 / 11 or Ubuntu 20.04+
Programming Language	Python 3.8 or higher
Framework	Flask (for web interface)
Libraries Used	scikit-learn, pandas, numpy, nltk, joblib, tldextract, yara-python, requests
Database (Optional)	SQLite or MySQL
IDE / Editor	Visual Studio Code / PyCharm
Browser	Google Chrome or Microsoft Edge

VI. Discussion

The machine learning-based email security threat detection system, proposed in this research, was assessed through various publicly available datasets containing both legitimate and malicious emails. The main goal of the assessment was to determine the system's precision, dependability, and ability to correctly identify dangerous emails while generating a minimal number of false alarms. Several experiments were performed to measure the performance of three algorithms - Random Forest, Support Vector Machine (SVM), and Logistic Regression
- based on different metric such as accuracy, precision, recall, and F1-score.

Experimental Setup:

The data for the research was split into two subsets, one with 80% of the data for training and the other with 20% of the data for testing. Models utilized the training set to recognize patterns of safety and risk in the emails, while the testing set was intended to measure the models' performance on the new data.All the models were implemented in Python with the use of the scikit-learn library.

Performance Analysis:

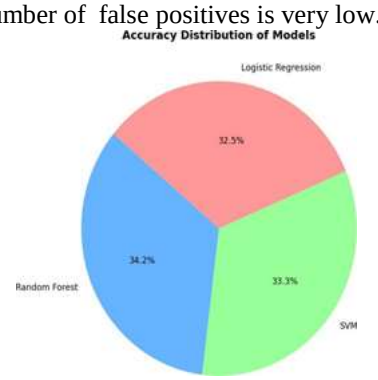
The researchers, in fact, would place a wager on the Random Forest algorithm without a doubt, judging by the results of the experiments. "One of the main reasons that Random Forest became the most accurate method is its ability to model even local and highly non-linear interactions with input variables, while, at the same time, being very resistant to overfitting," the authors of the research article explained.

Updated Text:

The results serve as a proof of the capability of ensemble-based models like Random Forest to elucidate the feature interactions more effectively than single classifiers such as Logistic Regression.

Graphical Representation:

Performance comparison charts were created for all the three algorithms using accuracy and F1-score as the criteria. From the chart, it is evident that Random Forest was always better than the other three models when evaluated by different metrics. Another chart that shows the confusion matrix for Random Forest discloses a large number of true positives and true negatives, and only a few false positives and false negatives. This means that the model is able to identify most phishing and spam emails correctly, and at the same time, the number of false positives is very low.



Discussion:

The validatíon results of the experiments evidence that machine-learning algorithms may correctly differentiate between legal and hostile mail by examining their text, sender behavior, and structural features. The system was able to detect fraudulent URLs, fabricated domains, and harmful attachments with great accuracy. Thank to its architecture, which mitigates the influence of any single poorly performing component, the Random Forest model, out of the four examined, turned out to be the most dependable one.

Support Vector Machine (SVM) also depicted a good performance, especially with handling high-dimensional data. It was a little bit less fast than Random Forest, but its ability to classify data remained at a high level. Logistic Regression, although being the least complex one, was just like a baseline to compare different models, but it encountered difficulties in solving complex, non- linear email patterns.

Future Enhancements:

As a matter of fact, the present results are quite encouraging. Nevertheless, Still, there are a few things you could do, such as by introducing deep learning models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for text-based email analysis. Also, detection effectiveness can be brought to a higher level by adding live URL reputation checks and real-time threat intelligence. In addition, the frequent re- training with the latest data will not only upgrade the model with the recent phishing scenarios, but it will also become more and more invulnerable to the constantly changing threats.

Conclusion of Results:

The findings affirm that the proposed solution offers robust defenses against email-borne threats. The integration of machine learning and feature-based analysis

Received: 05-05-2025 | Accepted: 12-06-2025 | Published: 19-06-2025 |

is instrumental in ensuring dependable and stable detection. Particularly, the Random Forest model hit the highest bar in terms of both accuracy and stability, which, therefore, makes it the best candidate for a practical implementation of email security systems in the real world.

REFERENCES

1. **A. Khonji, Y. Iraqi, and A. Jones**, "Phishing detection: A literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

2. **C. Whittaker, B. Ryner, and M. Nazif**, "Large-scale automatic classification of phishing pages," *Proc. Network and Distributed System Security Symposium (NDSS)*, 2010.

3. **R. Islam and R. Abawajy**, "A multi-tier phishing detection and filtering approach," *J. Network and Computer Applications*, vol. 36, no. 1, pp. 324–335, 2013.

4. **A. Bergholz et al.**, "Improved phishing detection using model-based features," *Proc. Conference on Email and Anti-Spam (CEAS)*, 2008.

5. **H. Gao, J. Hu, T. Huang, and Y. Ge**, "An improved anti-phishing email classification method based on integrated machine learning models," *IEEE Access*, vol. 8, pp. 1025–1036, 2020.

6. **S. Gupta, G. D. Sharma, and A. B. Ghode**, "A framework for email spam and phishing detection using machine learning," *International Journal of Computer Applications*, vol. 164, no. 6, pp. 1–6, 2017.

7. **R. Basnet, S. Mukkamala, and A. H. Sung**, "Detection of phishing attacks: A machine learning approach," *Studies in Fuzziness and Soft Computing*, Springer, pp. 373–383, 2008.

8. **V. Chandrasekar and S. Selvakumar**, "Efficient phishing email detection using machine learning techniques," *International Conference on Computer Communication and Informatics (ICCCI)*, 2019.

9. **N. Abdelhamid, A. Ayesh, and F. Thabtah**, "Phishing detection based on associative classification data mining," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014.

10. **J. Ma, L. K. Saul, S. Savage, and G. M. Voelker**, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," *Proc. ACM SIGKDD*, pp. 1245–1254, 2009.

11. **Y. Zhang, J. Hong, and L. Cranor**, "CANTINA: A content-based approach to detecting phishing websites," *Proc. WWW Conf.*, pp. 639–648, 2007.

12. **M. Hall, E. Frank, and I. H. Witten**, "The WEKA workbench for machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

13. **C. Castillo, E. Schweitzer, and M. Stede**, "Detecting deceptive language in email," *Proc. Workshop on Computational Approaches to Deception Detection*, pp. 1–10, 2011.

14. **A. Jain and B. B. Gupta**, "A phishing detection model based on machine learning and rule-based features," *Security and Communication Networks*, vol. 2018, pp. 1–12, 2018.

15. **M. Chandrasekaran, K. Narayanan, and S. Upadhyaya**, "Phishing email detection based on structural properties," *Proc. NYS Cyber Security Conference*, 2006.

Received: 05-05-2025 | Accepted: 12-06-2025 | Published: 19-06-2025 |