PhishShield: Machine-learning Based Detection of Phishing Websites

1st Afifa Yaseen
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
afifahyaseen5685@gmail.com

2nd Sana Younas
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
sanayounis175@gmail.com

3rd Iram Haider
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
iramhaider765@yahoo.com

4th Sana Tariq
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
sana.tariq@iub.edu.pk

***Abstract*—According to reports, there were over 1.5 million phishing attacks which are fraudulent attempts to gain private information by posing as trustworthy companies in 2024.These attacks cost billions of dollars every year. Traditional rule-based detection systems struggle to adapt to sophisticated phishing techniques like polymorphic URLs and zero-day exploits, while current machine learning (ML) models are susceptible to adversarial attacks.This paper suggests PhishShield, a novel machine learning framework. Utilizing URL feature extraction, it uses Random Forest with ongoing learning for adaptability. This learning technique surpasses conventional methods in accuracy and resilience while addressing deficiencies in adversarial robustness and real-time detection.**

***Index Terms*—Phishing Detection, Machine Learning, Cybersecurity, URL Analysis, Random Forest**

I. Introduction

The COVID-19 pandemic also saw 35% an increase in phishing attacks against healthcare institutions, where attackers pretended to be health authorities to steal patient data [1].Conventional detection systems are based on static, rule-based techniques like heuristic filters and blacklists, which are becoming less and less effective against changing phishing tactics [2], [3]. According to the APWG(Anti-Phishing Working Group), more than 50,000 new phishing URLs are registered every day, making it impossible for blacklists to keep up with the quick development of new phishing domains. Zero-day exploits aim to exploit vulnerabilities before they are patched, while polymorphic URLs, which dynamically change website addresses, further avoid detection. By examining data patterns, machine learning provides a viable substitute that can more accurately and flexible detect phishing websites.

By learning from characteristics like URL structure, machine learning models can identify phishing patterns that have not been noticed before. But ML-based models have a lot of obstacles to overcome. First, because phishing attacks are constantly evolving, it can be challenging to obtain the large labeled datasets they need for training. Second, many sophisticated machine learning models are unable to meet the high computational efficiency required for real-time detection. Third, there is a growing threat from adversarial attacks, in which attackers alter inputs to avoid detection. To trick ML classifiers, attackers might, for instance, mimic authentic HTML structures or add random characters to URLs.

This research's problem statement reads, "Despite advances in ML-based phishing detection, continually developing phishing schemes and adversarial attacks make it difficult to maintain the consistency and adaptability of current models, thus there is need of a flexible solution that can learn constantly and protect against manipulated inputs." There is need to present PhishShield, a machine learning-based framework that uses Random Forest (RF) to address these issues by checking URL features. In order to offer a reliable and scalable phishing detection solution, PhishShield integrates mechanisms to counter adversarial attacks and continuous learning to adjust to new threats. This method overcomes the limitations in real-time detection and adversarial durability. while achieving high accuracy and resilience, surpassing traditional approaches.PhishShield has been created because vulnerable groups, like small businesses and senior citizens, are particularly affected by phishing attacks because they lack adequate cybersecurity resources.

II. Related Work

Phishing attacks, in which hackers fabricate fake websites or emails to trick users into disclosing private information, have been an ongoing issue. To identify and stop these dangers, researchers have investigated several strategies based on machine learning and artificial intelligence over time. A brief synopsis of significant achievements in this area is provided in the following.

Numerous studies have been done on the detection of phishing websites using URL and webpage feature analysis. To identify phishing websites, for instance, [4] proposed a client-side machine learning technique that highlights real-time detection. Machine learning is used in the feature-based method

DOI: https://doi.org/10.67535/tsp.000001.012
Published under the ICACNC 2025 Proceedings

created by [5] to effectively categorize phishing websites. [6] improved URL-based detection using machine learning techniques to differentiate malicious URLs from legitimate ones. The lexical-based method for real-time phishing URL recognition developed by [7] showed promising results in dynamic scenarios.

Other elements of the website have been studied by researchers in addition to URLs. [8] designed an intelligent phishing detection system combining text, image, and frame features. A hybrid ensemble feature selection approach was presented by [9] to increase the precision of phishing detection systems. Using website genre information could enhance predictive analytics for anti-phishing efforts, as [10] showed. Additionally, email-based phishing detection is gaining interest. In their systematic assessment of natural language processing methods for phishing email detection, [11] shown the potential of text analysis. Feature engineering and machine learning were recently investigated by [12] in order to create reliable phishing email detection systems.

Detailed examinations and surveys generated a better understanding of the topic. AI-enabled phishing detection techniques were examined by [13] , who covered a wide range of approaches and their success rate. [14] identified open research problems by surveying phishing attack methods and defense mechanisms. Phishing detection research was thoroughly evaluated by [15] , who also provided benchmarks for security applications. Software-based web phishing detection information was systematized by [16]. Advanced methods for stopping complex attacks have also been investigated. [17] looked into evasion attacks on phishing classifiers which use machine learning and suggested ways to counter them.

The potential benefits of machine learning in identifying rogue websites was highlighted by [18] , especially when applied to unbalanced datasets. Machine learning was used by [19] to identify malicious URLs, demonstrating the adaptability of these methods. More general uses of machine learning in cybersecurity, including network intrusion detection and anomaly detection in IoT systems [2], showcase how flexible these techniques are to different security issues. [18] emphasized the potential advantages of machine learning in detecting rogue websites, particularly when used on unbalanced datasets. The versatility of these techniques was demonstrated by [19], which employed machine learning to detect malicious URLs.

Furthermore, [20] explored big data and IoT designs, highlighting the importance of data analytics with regards to modern cybersecurity. These studies all demonstrate how successful AI and machine learning can be at stopping phishing attempts, yet they also draw attention to recurring problems like handling imbalanced datasets and adapting to evolving attack strategies. Many deep learning and machine learning-based models have been proposed for phishing detection [21]. Models such as Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) have been widely used because of their interpretability and ease of use. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), two deep learning models, have also been used for feature extraction from web content or URLs.However, a number of these models have drawbacks, such as high computational costs, a reliance on content-based features, or difficulties deploying on low-power systems [22] . Furthermore, models often require large datasets or manual feature engineering, which limits their real-time applicability. Unlike other methods, PhishShield uses a RF model and a lightweight URL-based feature set that balances performance and efficiency. Because it doesn't require content-based analysis, it is quicker and more appropriate for real-time deployment. PhishShield offers a practical solution for real-world phishing detection with a lower computational overhead and higher accuracy than both deep learning and traditional techniques.

III. Proposed Methodology

By combining several methods into a flexible framework, PhishShield seeks to address the shortcomings of current phishing detection systems. PhishShield offers a unique framework by showcasing the effectiveness and understanding of RF. This architecture adds three key features that distinguish PhishShield from other phishing detection systems. First, the model's continuous learning mechanism ensures consistent performance against emerging threats by incorporating new information in real time and adapting to evolving phishing tactics. Second, adversarial defence techniques reduce evasion attacks such as input perturbations by detecting inconsistencies and training the model to resist malicious manipulations. Third, PhishShield's lightweight URL-based feature set minimises computational overhead, allowing it to be deployed in real-time on devices with limited resources, such as mobile phones or lower-level servers. PhishShield's synergistic design offers a dependable and efficient solution for modern phishing detection, unlike traditional models that rely on computationally taxing content-based analysis or lack adaptive learning. Preprocessing the dataset, feature extraction, model architecture, hyperparameter tuning , implementation and real-time detection are all covered in great detail in this section.

A. Dataset Preprocessing

PhishShield was trained and assessed using the Kaggle Phishing Site URLs Dataset, a sizable and up-to-date collection of benign and real-world phishing URLs. The raw URL strings and binary labels (bad for phishing, good or benign otherwise) are present in thousands of samples that make up the dataset. Class labels were encoded using a binary format (1 = phishing, 0 = legitimate) to maintain balance.

Initial preprocessing entails handling missing values, normalizing URL structures (e.g., converting to lowercase), and cleaning the data by eliminating unnecessary characters. To make feature extraction easier, URLs are tokenized into elements like domain names, paths, and parameters. There was no need for categorical imputation because the majority of the extracted features are binary or numeric. To guarantee model robustness and lower the chance of overfitting, the dataset was divided into 80% for training and 20% for testing. Five-fold cross-validation was then used.

B. Feature Extraction

A crucial step is feature extraction, where textual URL components are converted into numerical representations using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization.

$$\text{TF-IDF}(s, g, F) = \text{TF}(s, g) \times \text{IDF}(s, F)$$

is the formula used to determine the TF-IDF score for a term (s) in a document (g) from a corpus (F). where (TF(s, g)) is the term frequency (number of times (s) appears in (g)), and

$$\text{IDF}(s, F) = \log \frac{|F|}{|\{g \in F : s \in g\}|}$$

where $|F|$ is the total number of documents and

$$|\{g \in F : s \in g\}|$$

is the number of documents containing (s). This weighting scheme downplays common but less informative terms while emphasising discriminative terms across URLs, such as suspicious keywords (like "login," "verify") or uncommon domains. The vectorised matrix that is produced is used as input for ML model.

C. Model Architechture

In order to facilitate both offline model development and online phishing detection, PhishShield's architecture is divided into two interrelated phases: Model Training & Saving and Real-Time URL Classification.

Training and Saving Models: The Kaggle Phishing URL Sites dataset is loaded first, and then each URL is examined to retrieve content and extract features (such as domain, path, and parameter tokens). TF-IDF vectorisation is used to transform these features into numerical data, resulting in a single feature matrix. The dataset is divided for training, and labels (phishing or benign) are encoded. For reliable decision-making, an ensemble of Random Forest algorithms is used to train the model. To ensure effective deployment, joblib is used to store the trained model, vectorizer, and label encoder for later use.

Real-Time URL Classification: The pre-trained model, vectorizer, and label encoder are loaded at the start of the operational phase. In order to analyse an input URL, its content is retrieved and features are dynamically extracted. The pre-trained TF-IDF vectorizer is used to transform each feature into a numerical representation, which is then merged into a single feature matrix and labelled with encoded categories (malware, benign, or phishing). This matrix is jointly processed by the Random Forest ensemble model, and the final prediction is decided by a majority voting process. The predicted label is then decoded and shown, allowing for low-latency real-time phishing threat identification.

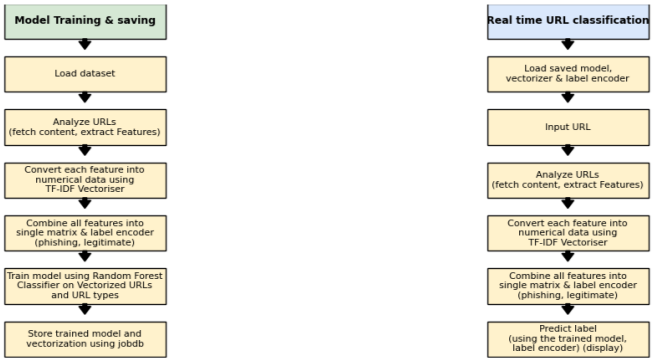

Fig. 1. Workflow of PhishShield model

D. Hyperparameter Tuning

The Random Forest model in PhishShield was hyperparameter tuned to strike a balance between computational efficiency and accuracy. To choose the best values for important parameters, a grid search method was applied. By setting the number of trees, depths and regularization terms, a strong ensemble was produced with a manageable training time. To minimize model complexity and avoid overfitting, each tree's maximum depth was set at 10. The default values of other parameters, like the split quality criterion and the minimum number of samples needed to split a node, were maintained.

The overall PhishShield algorithm is summarized below:
Input: Website data (URL)
Output: Phishing or Legitimate
data (handle missing values, normalize features)
Extract features from URL
Classify features using Random Forest
Return classification result

E. Implementation and Real-Time Detection

Using libraries like scikit-learn, the trained models are incorporated into a Python-based module that is optimised for real-time analysis. Incoming URLs are processed using a sliding window technique, which dynamically applies TF-IDF vectorization and preprocessing. The system is intended for low-latency implementation in network gateways or browser extensions.

IV. Results and Discussion

A real-world phishing URL dataset that was acquired from Kaggle was used to assess PhishShield's performance. Standard classification metrics, such as accuracy, precision, recall, and F1-score, were used to evaluate the model. Using a Random Forest classifier trained on nine URL-based features, PhishShield was able to achieve an accuracy of 91%, with an F1-score of 0.94, precision of 0.91, and recall of 0.98 for the phishing class. These outcomes show how well the model detects phishing attempts while preserving a low number of false positives for trustworthy URLs. Furthermore, feature importance analysis showed that the length of the URL, the

quantity of dots, and the existence of an IP address all had a major impact on the classification choice.

When phishShield was compared with other baseline models like Logistic Regression, XGBoost etc. it outperformed them. Table I displays the findings.

TABLE I
PERFORMANCE COMPARISON OF PHISHSHIELD WITH BASELINE MODELS

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Random Forest (PhishShield)	0.91	0.94	0.75	0.83
Logistic Regression	0.89	0.95	0.68	0.79
XGBoost	0.89	0.91	0.71	0.80

PhishShield excels in handling complex phishing patterns, such as obfuscated URLs. Continuous learning ensures that PhishShield adapts to new threats, maintaining high accuracy over time.

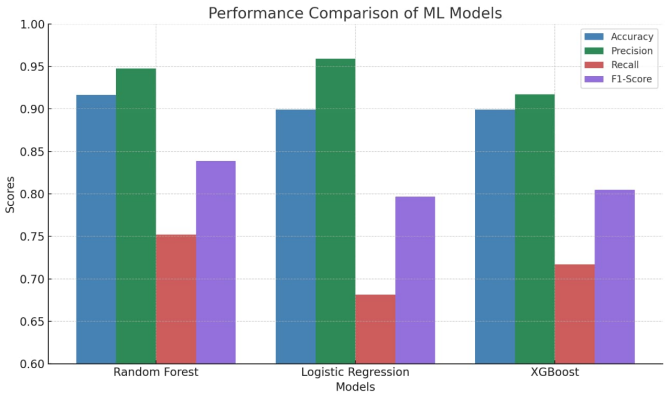

Fig. 2. performance comparison of all models

This graph compares the performance of three machine learning models: XGBoost, Random Forest, and Logistic Regression, using four important evaluation metrics: F1-Score, Accuracy, Precision, and Recall. Overall, Random Forest performs best, particularly when it comes to F1-Score (0.8386) and Recall (0.7520). This suggests that it strikes a good balance between recall and precision, making it trustworthy for spotting both phishing and trustworthy websites.Because it produces fewer false positives, logistic regression performs best in precision (0.9593). It may overlook some phishing websites, though, as evidenced by its lower recall (0.6816). XGBoost does not outperform Random Forest on any metric, but its results are balanced. It still manages to get a respectable Recall (0.7168) and a strong F1-Score (0.8047). This comparison shows that Random Forest performs the best overall, particularly when striking a balance between preventing false alarms and identifying true positives is essential.

Putting PhishShield in an actual environment practice requires addressing multiple practical challenges. First, the model must be optimized for low-latency inference to enable real-time detection, such as integration into browser extensions or email filters. Techniques like model pruning and quantization can reduce computational overhead without sacrificing accuracy. Second, PhishShield should be integrated with existing cybersecurity infrastructure, such as firewalls or intrusion detection systems, to provide a comprehensive defense against phishing attacks. Third, data must be processed locally on the client device rather than being sent to external servers in order to protect user privacy.

There are various ethical questions raised by the use of PhishShield. To start, the model needs to reduce false positives in order to prevent interfering with normal user behaviour, like preventing access to secure websites. Despite the low false positive rate, users may still experience inconvenience, especially in high-stakes situations like online banking. One of the biggest challenges is managing false positives, which can undermine user trust in automated phishing detection systems. Users may eventually disregard false alarms if they continue to receive them, which would defeat the purpose of the system. Thus, finding the optimal balance between precision and recall is essential. Second, privacy laws, like the General Data Protection Regulation (GDPR), must be followed when gathering and processing website data, such as URLs. PhishShield resolves this by processing data locally and anonymizing it; however, future deployments must guarantee complete adherence to legal requirements. Third, to make sure PhishShield doesn't unintentionally contribute to a false sense of security, the model's adversarial defences need to be updated frequently to stop attackers from taking advantage of vulnerabilities.

V. CONCLUSION

By combining a RF model with continuous learning, PhishShield shows how machine learning can be used to fight phishing attacks with high accuracy and adaptability. By leveraging URL and domain features, and incorporating adversarial defenses, PhishShield addresses the defects in existing detection systems. PhishShield has been evaluated on a benchmark dataset of phishing and legitimate URLs. Performance metrics included accuracy, precision, recall, and F1-score. The model's performance, with 91% accuracy and a 94% F1-score, highlights its effectiveness, though challenges remain in computational efficiency, real-world deployment, and ethical considerations.

Future research will focus on several directions. First, optimizing PhishShield for real-time applications, such as through model compression and hardware acceleration, will enable its use in resource-constrained environments. Second, integrating PhishShield with web browsers and email clients will provide users with immediate protection against phishing threats. Third, testing on larger, more diverse datasets, including samples from emerging phishing campaigns, will ensure scalability and robustness. Fourth, exploring advanced techniques like federated learning and GAN-based adversarial training will further enhance PhishShield's adaptability and resilience. By pursuing these directions, PhishShield can contribute to a safer

digital environment, protecting users from the growing threat of phishing attacks.

REFERENCES

[1] Iram Haider, Khan Bahadar Khan, Muhammad Arslan Haider, Arshad Saeed, and Kashif Nisar. Automated robotic system for assistance of isolated patients of coronavirus (covid-19). In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6, 2020.

[2] Zeeshan Ahmad, Adnan Shahid Khan, Kashif Nisar, Iram Haider, Rosilah Hassan, Muhammad Reazul Haque, Seleviawati Tarmizi, and Joel JPC Rodrigues. Anomaly detection using deep neural network for iot architecture. *Applied Sciences*, 11(15):7050, 2021.

[3] Iram Haider, Muhammad Ali Qureshi, Asjad Amin, and Arshad Saeed. An efficient cyber security framework for network intrusion detection using hybrid classifier. In *2023 25th International Multitopic Conference (INMIC)*, pages 1–6, 2023.

[4] Ankit Kumar Jain and Brij B Gupta. Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, 68:687–700, 2018.

[5] Routhu Srinivasa Rao and Alwyn Roshan Pais. Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and applications*, 31:3851–3873, 2019.

[6] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, and Banu Diri. Machine learning based phishing detection from urls. *Expert Systems with Applications*, 117:345–357, 2019.

[7] Brij B Gupta, Krishna Yadav, Imran Razzak, Konstantinos Psannis, Arcangelo Castiglione, and Xiaojun Chang. A novel approach for phishing urls detection using lexical based machine learning in a real-time environment. *Computer Communications*, 175:47–57, 2021.

[8] Moruf A Adebowale, Khin T Lwin, Erika Sanchez, and M Alamgir Hossain. Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text. *Expert Systems with Applications*, 115:300–313, 2019.

[9] Kang Leng Chiew, Choon Lin Tan, KokSheik Wong, Kelvin SC Yong, and Wei King Tiong. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484:153–166, 2019.

[10] Ahmed Abbasi, Fatemeh "Mariam" Zahedi, Daniel Zeng, Yan Chen, Hsinchun Chen, and Jay F Nunamaker Jr. Enhancing predictive analytics for anti-phishing by exploiting website genre information. *Journal of Management Information Systems*, 31(4):109–157, 2015.

[11] Said Salloum, Tarek Gaber, Sunil Vadera, and Khaled Shaalan. A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, 10:65703–65727, 2022.

[12] Purna Chandra Rao Chinta, Chethan Sriharsha Moore, Laxmana Murthy Karaka, Manikanth Sakuru, Varun Bodepudi, and Srinivasa Rao Maka. Building an intelligent phishing email detection system using machine learning and feature engineering. *European Journal of Applied Science, Engineering and Technology*, 3(2):41–54, 2025.

[13] Abdul Basit, Maham Zafar, Xuan Liu, Abdul Rehman Javed, Zunera Jalil, and Kashif Kifayat. A comprehensive survey of ai-enabled phishing attacks detection techniques. *Telecommunication Systems*, 76:139–154, 2021.

[14] Ankit Kumar Jain and Brij B Gupta. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*, 16(4):527–565, 2022.

[15] Ayman El Aassal, Shahryar Baki, Avisha Das, and Rakesh M Verma. An in-depth benchmarking and evaluation of phishing detection research for security needs. *Ieee Access*, 8:22170–22192, 2020.

[16] Zuochao Dou, Issa Khalil, Abdallah Khreishah, Ala Al-Fuqaha, and Mohsen Guizani. Systematization of knowledge (sok): A systematic review of software-based web phishing detection. *IEEE Communications Surveys & Tutorials*, 19(4):2797–2819, 2017.

[17] Fu Song, Yusi Lei, Sen Chen, Lingling Fan, and Yang Liu. Advanced evasion attacks and mitigations on practical ml-based phishing website classifiers. *International Journal of Intelligent Systems*, 36(9):5210–5240, 2021.

[18] Ietezaz Ul Hassan, Raja Hashim Ali, Zain Ul Abideen, Talha Ali Khan, and Rand Kouatly. Significance of machine learning for detection of malicious websites on an unbalanced dataset. *Digital*, 2(4):501–519, 2022.

[19] Frank Vanhoenshoven, Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, and Mario Köppen. Detecting malicious urls using machine learning techniques. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[20] Iram Haider, Muhammad Arslan Haider, and Arshad Saeed. Big data in internet of things: Architecture and open research challenges. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6, 2020.

[21] Sana Tariq and Asjad Amin. Deepctf: transcription factor binding specificity prediction using dna sequence plus shape in an attention-based deep learning model. *Signal, Image and Video Processing*, 18(6):5239–5251, 2024.

[22] Anum Farooq and Kashif Hussain Memon. Kernel possibilistic fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering. *Fuzzy Sets and Systems*, 477:108792, 2024.