Suyash Kangude, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

PhishShield: A Detection and Prevention System for Phishing Domains

Suyash Kangude, Prashant Aher, Aniket Mehare, Sanket Bhavari, Harshada Ahire, Abhishek Badadhe
Dept. of Information Technology, Sanjivani College of Engineering, Savitribai Phule Pune University Pune, India

Abstract- Phishing remains one of the most widespread and dangerous cyberattacks, where adversaries create deceptive domains and fraudulent websites to obtain confidential information from unsuspecting users. As phishing techniques evolve and become increasingly advanced, traditional security measures are often unable to recognize or block these threats, since attackers continually modify their strategies. This project introduces PhishEye a machine learning–based phishing domain detection system designed to identify, evaluate, and respond to phishing attempts. The framework follows a structured pipeline consisting of data collection, feature extraction, model training, real-time threat detection, takedown request automation, and dashboard monitoring. Datasets were gathered from trusted sources such as PhishTank and OpenPhish, complemented by additional feeds and domain permutations generated with Dnstwist. To create robust feature vectors, multiple attributes were included: lexical, domain, network, SSL/TLS, and webpage content features. Machine learning models—including Random Forest and Logistic Regression—were trained and validated using evaluation measures such as accuracy, precision, recall, and F1-score. Once deployed, the system monitors domains in real time and assigns a risk score (low, medium, or high) to estimate their legitimacy. For domains classified as high risk, PhishEye initiates an automated takedown request to the hosting provider, registrar, or relevant CERT authority. Finally, a web-based dashboard provides a comprehensive interface for administrators, showing real-time statistics and system performance for effective phishing defense.

Keywords— Phishing, Machine Learning, Cybersecurity, PhishEye, Domain Detection, PhishTank, OpenPhish.

I. INTRODUCTION

In today's digital world, phishing is now known as one of the most common and dangerous cyberthreats. Through the use of phishing websites and domains, attackers trick users into revealing personal information, passwords, and banking credentials. The complexity of phishing attempts is increasing, making it increasingly challenging to detect them using traditional security measures. This emphasizes the pressing need for automated and intelligent solutions. This work aims to create a phishing domain detection system based on machine learning. By examining a variety of characteristics, including URL structure, server information, SSL/TLS certificates, and even webpage content, the system is intended to detect malicious domains. Compared to conventional rule-based techniques, the system can identify suspicious websites more accurately by integrating these various features. The approach of the proposed system includes data collection, feature extraction, model training, real-time threat detection, takedown request processing, and a monitoring dashboard. To replicate actual phishing scenarios, data was gathered from reliable sources like PhishTank and OpenPhish, processed, and enhanced using tools like Dnstwist. Machine learning models such as Random Forest and logistic regression were trained and evaluated using recognised performance criteria to ensure reliability. Overall, by creating a workable and efficient phishing detection solution, this study strengthens cybersecurity. The system is helpful for both individuals and organizations in thwarting phishing threats because it not only detects malicious domains but also offers monitoring and takedown assistance.

Methodology: In this study, we designed and constructed a phishing domain detection system using an organized methodology. Data collection, feature extraction, model training, threat detection and monitoring, takedown request flow, and dashboard and monitoring were the six main steps that made up the process[1]. Below is a detailed explanation of each step.

Data Collection: Collecting information from trustworthy sources was the first step. We gathered data about phishing from reliable websites like PhishTank and OpenPhish, which keep lists of phishing websites that are current. Besides them, we used security feeds, user-reported dubious websites, and public phishing reports. Automated web crawling techniques were used to find more suspicious domains on the internet in order to diversify the dataset. The dataset needed preprocessing because errors are common in raw data. This step involved flagging incomplete records for review, eliminating duplicate entries, and eliminating malformed URLs. Following cleaning, we had a domain dataset that was organized and properly labeled, ready for the following phase. This dataset served as the project's foundation since the quality of the data directly affects the detection system's performance. In addition to the initial data collection pipeline, the system also integrates long-term historical phishing datasets to study attacker behavior patterns over time. Historical datasets help in identifying recurring phishing campaigns, commonly abused registrars, and frequently targeted brands[2]. This temporal data allows the system to detect seasonal attack trends and emerging threat waves.

The platform also incorporates API-based integrations with threat intelligence providers to continuously enrich the dataset. These APIs supply fresh domain indicators, blacklist feeds, and zero-day phishing URLs, ensuring that the dataset remains current and adaptive. Crowd-sourced intelligence is another major component of data expansion, where users can submit suspicious URLs directly through the dashboard interface[3].

Feature Extraction : Identifying useful attributes that could help in identifying phishing domains from authentic ones was the next step after the dataset was prepared. Features are quantifiable attributes that give machine learning models crucial cues. The features were divided into five groups by us:

Lexical Features: These are determined by the URL's actual structure. Examples include the domain name's randomness (entropy), length, subdomain amount, and unique character existence, such as "@" or "-." Phishing websites frequently confuse users with odd or excessively lengthy URLs.

Domain Features: Details collected from WHOIS records, including the domain's age, registrar reputation, and registration information. A large number of phishing domains are brand-new and come from registrars that are not as reliable.

Network Features: Technical information like the hosting server's location, Autonomous System Number (ASN) lookup, and IP address reputation. Attackers frequently use hosting companies with weak or lacking security measures.

Features of SSL/TLS: Information about security certificates, such as their validity, issuer, and expiration date. A lot of phishing websites make use of temporary or invalid certificates.

Content Features: Features of the webpage itself, including the HTML code structure, the existence of login forms, and any hidden or obfuscated scripts that might be attempting to steal data[4].

We developed a multi-dimensional profile for every domain by integrating all these characteristics, It makes it easier for machine learning models to identify phishing attempts.

Beyond conventional lexical and network features, the system incorporates behavioral indicators such as abnormal redirection patterns, suspicious JavaScript execution, and cookie manipulation behavior. These behavioral signals help uncover sophisticated phishing sites that attempt to evade static detection techniques. Language-based

features are also extracted using natural language processing to identify grammatical errors, abnormal sentence structure, and deceptive wording commonly found in phishing content.

Visual similarity analysis further enhances detection accuracy by comparing phishing webpages with legitimate brand websites. Logo detection, color scheme similarity, and layout matching help identify visually deceptive pages that impersonate trusted services. All features are normalized and correlation-tested to remove redundancy and improve model efficiency[5]. This multi-layered feature representation provides a comprehensive behavioral and structural profile of each domain.

Model Training : We proceeded to train the machine learning models after extracting the features. The Dnstwist tool was utilized to enhance the dataset. As a common tactic used by attackers to deceive users, this tool creates permutations of domain names (for instance, "goggle.com" instead of "google.com"). To choose the most crucial features prior to training, we used feature engineering techniques such as information gain and chi-square tests[6].

This reduced data noise and improved model performance. The data was then divided into three parts: 70% for training, 20% for validation, and 10% for testing. This allowed the models to be trained without overfitting and evaluated properlyClick or tap here to enter text..

We experimented with several machine learning techniques, including Random Forest and Logistic Regression, because of their applicability to categorisation problems. The models were tested on previously unseen data after learning patterns from the retrieved characteristics[7]. To evaluate performance, we used common measures such as F1-score, recall, accuracy, and precision.We could clearly see from these metrics how well the models identified phishing domains while reducing false alarms[8].

To improve prediction reliability, ensemble learning techniques are applied by combining multiple classifiers into a unified decision engine. This approach increases resilience against evasion techniques and improves classification accuracy across diverse phishing strategies. Stratified k-fold cross-validation ensures balanced learning and prevents bias in model training.

Hyperparameter optimization is performed using grid search and Bayesian optimization to fine-tune model performance. Deep learning models such as LSTM networks are also evaluated for sequential URL pattern analysis. Model explainability frameworks like SHAP and LIME are integrated to provide transparent decision reasoning, which is essential for trust, auditability, and regulatory compliance[9].

Threat Detection & Monitoring : After training, the model was incorporated into a system that could detect threats in real t ime. The system extracts the features of a new website or domain and compares them to the model's prior knowledge. The system gives the domain a risk score based on this analysis: High Score → Most likely a phishing website.

II. MEDIUM SCORE → SUSPICIOUS; MANUAL REVIEW IS NECESSARY

Low Score → Reliable and secure.
This method guarantees that malicious phishing websites are promptly reported. Additionally, the system continuously scans websites to identify cloned websites, phony login pages, and other phishing tactics as they emerge. Because of this, the solution remains flexible and efficient even when attackers modify their tactics[10].

The detection engine is deployed as a real-time microservice capable of scanning large volumes of URLs with minimal latency. Stream-processing pipelines enable instant feature extraction and classification, ensuring immediate threat identification. Newly registered domains are continuously monitored through DNS feeds and automatically scanned before attackers can weaponize them.

Adaptive learning mechanisms retrain the model periodically using newly discovered phishing samples, allowing the system to evolve with

changing attack techniques. Automated alerting notifies security teams via multiple channels when critical threats are detected[11]. The monitoring platform also provides attack analytics, campaign correlation, and infrastructure mapping for deeper threat investigation.

Feature / Parameter	Traditional / Baseline Phishing Detection	PhishEye (Proposed System)
Detection Approach	Rule-based / blacklist-based	AI + ML-based intelligent detection
Response Type	Reactive (after attack happens)	Proactive (early detection + prevention)
URL Checking	Only checks known phishing URLs	Detects even unknown / new phishing domains
HTTPS Handling	Often trusts HTTPS websites	Detects phishing even if HTTPS is present
Feature Analysis	Limited URL pattern matching	Domain + URL + Content feature extraction
Threat Scoring	Usually yes/no decision	Threat score (Low/Medium/High risk)
Real-time Detection	Slow / depends on database update	Real-time scanning and classification
Content Inspection	Not supported in many tools	Detects fake login pages using HTML/content analysis
Central Dashboard	Not available	Centralized dashboard for monitoring
Automation	Manual reporting required	Automated takedown request generation
Adaptability	Low (fixed rules)	High (model learns and improves over time)
Accuracy	Moderate, high false positives	Higher accuracy with ML + multi-feature validation

Takedown Request Flow : To protect users, phishing sites must be taken down from the internet in addition to being detected. We created a takedown request procedure for this reason. The system automatically creates a comprehensive report with all the evidence, including domain details, SSL certificate information, and detection reasons, when a high-risk domain is detected[12].

A takedown request is created using this report and submitted to either the hosting provider (the business that provides the server) or the domain registrar (the entity in charge of the website name). The request can be forwarded to higher authorities, such as law enforcement or Computer Emergency Response Teams (CERTs), if these providers do not reply.

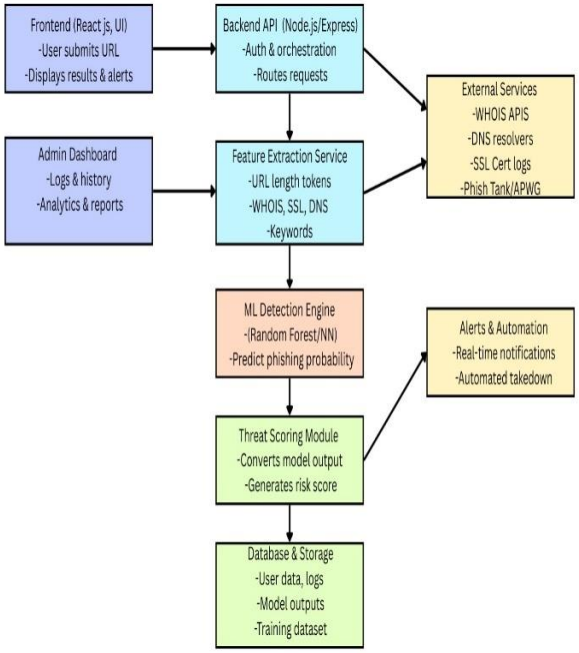
The system uses distinct states to track the takedown process in order to preserve transparency: submitted → in progress → removed. This guarantees that all malicious domains are appropriately pursued until they are dealt with.

The takedown workflow automates evidence generation while ensuring compliance with legal and forensic standards. Each takedown request includes digitally signed logs, screenshots, HTML source code, SSL certificates, and infrastructure metadata. This preserves evidentiary integrity and accelerates response from registrars and hosting providers[13].
A centralized abuse contact registry enables rapid routing of takedown requests to the appropriate authorities. Priority-based escalation ensures that high-impact phishing campaigns are handled with urgency. Jurisdiction-aware compliance tracking guarantees that requests meet international cybersecurity and regulatory requirements, enabling coordinated takedown operations with CERTs and law enforcement agencies.

Dashboard & Monitoring : A web-based dashboard was created to make the system transparent and

user-friendly. Real-time updates on phishing detection and removal efforts are available on this dashboard. Administrators and users can see:

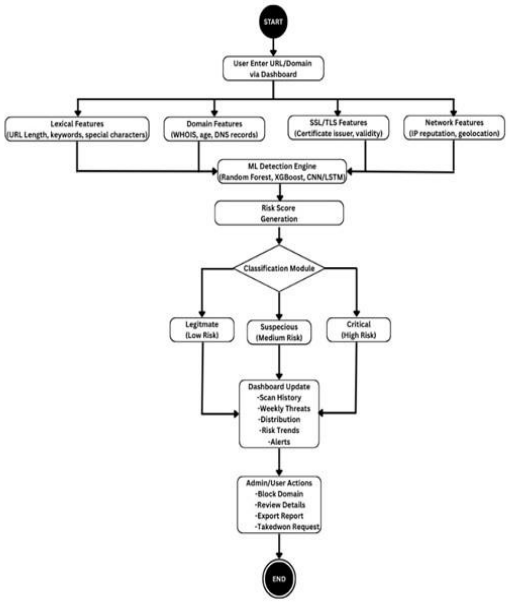
The number of phishing websites found.



How many takedowns are going on right now?
The system's overall performance and accuracy.
The dashboard employs color coding for improved visualization: green for secure domains, yellow for dubious sites, and red for high-risk phishing sites[14]. In order to facilitate data analysis and prompt action by security teams, charts and graphs are also included to display trends and detection statistics. The dashboard serves as the main hub for managing phishing threats by integrating detection, monitoring, and reporting into a single interface.

The dashboard functions as a full-scale Security Operations Center (SOC) interface, offering real-time visibility into phishing detection, takedown progress, and infrastructure analytics. Role-based access control ensures secure access for administrators, analysts, and auditors. Interactive visualizations provide deep insight into attack trends, threat distribution, and system effectiveness.

Predictive analytics modules forecast upcoming phishing surges based on historical attacker behavior. Exportable compliance reports support audits and executive decision-making[15]. By centralizing detection, monitoring, response, and reporting into a single interface, the dashboard serves as the operational core of the phishing defense ecosystem.



III. RESULT

The performance of the proposed system was evaluated using standard classification metrics, namely accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (ROC–AUC). Accuracy measures the overall correctness of the model and is defined as:

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

The proposed model achieved an accuracy of 97.33 %, indicating reliable overall performance.

Precision evaluates the correctness of positive predictions and is given by:

$$\text{Precision} = \frac{TP}{TP + FP}$$

A precision of 98.12 % confirms a very low false-positive rate. Recall, which measures the model's ability to correctly identify positive instances, is expressed as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

The recall value of 95.81 % demonstrates strong sensitivity.

The harmonic mean of precision and recall is represented by the F1-score:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The obtained F1-score is 96.95 %. The ROC–AUC metric, which evaluates the model's discriminative capability, achieved 99.62 %, indicating excellent classification performance.

IV. CONCLUSION

In order to detect phishing websites, we created a machine learning-based method in this study. After cleaning the data from reliable sources, we looked at key components like URLs, domain information, and certificates. Our models are able to evaluate websites in real time and provide a risk score that indicates whether the site is safe or dangerous. We also included a dashboard to make the results easy to understand and a takedown procedure to eliminate dangerous websites. All things considered, this system is an easy and efficient way to lessen phishing threats and keep users safer online.

REFERENCES

1. R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," IEEE Access, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.
2. S. Saeed et al., "Dataset of suspicious phishing URL detection," 2024, doi: 10.17632/6tm2d6sz7p.1.
3. U. Zara, K. Ayyub, H. Ullah Khan, A. Daud, T. Alsahfi, and S. Gulzar Ahmad, "Phishing Website Detection Using Deep Learning Models," IEEE Access, vol. 12, pp. 167072–167087, 2024, doi: 10.1109/ACCESS.2024.3486462.
4. S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," Applied Sciences (Switzerland), vol. 13, no. 8, Apr. 2023, doi: 10.3390/app13084649.
5. A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 590–611, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.004.
6. S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," Annals of Data Science, vol. 11, no. 1, pp. 217–242, Feb. 2024, doi: 10.1007/s40745-022-00379-8.
7. R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," IEEE Access, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.
8. S. H. Ahammad et al., "Phishing URL detection using machine learning methods," Advances in Engineering Software, vol. 173, Nov. 2022, doi: 10.1016/j.advengsoft.2022.103288.
9. J. Lee, P. Lim, B. Hooi, and D. M. Divakaran, "Multimodal Large Language Models for Phishing Webpage Detection and Identification," in eCrime Researchers Summit, eCrime, IEEE Computer Society, 2024, pp. 1–13. doi:10.1109/eCrime66200.2024.00007.
10. A. V Krishna, B. Jose, K. Anilkumar, and O. Thomas Lee, "Phishing Detection using Machine Learning based URL Analysis: A Survey." [Online]. Available: www.ijert.org
11. Kara, M. Ok, and A. Ozaday, "Characteristics of Understanding URLs and Domain Names Features: The Detection of Phishing Websites with Machine Learning Methods," IEEE Access, vol. 10, pp. 124420–124428, 2022, doi: 10.1109/ACCESS.2022.3223111.
12. A. Brasoveanu, M. Moodie, and R. Agrawal, "Textual evidence for the perfunctoriness of

independent medical reviews," in CEUR Workshop Proceedings, CEUR-WS, 2020, pp. 1–9. doi: 10.1145

13. S. H. Ahammad et al., "Phishing URL detection using machine learning methods," Advances in Engineering Software, vol. 173, Nov. 2022, doi: 10.1016/j.advengsoft.2022.103288.
14. M. Sanchez-Paniagua, E. F. Fernandez, E. Alegre, W. Al-Nabki, and V. Gonzalez-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," IEEE Access, vol. 10, pp. 42949–42960, 2022, doi: 10.1109/ACCESS.2022.3168681.
15. R. Mahajan and I. Siddavatam, "Phishing Website Detection using Machine Learning Algorithms," Int. J. Comput. Appl., vol. 181, no. 23, pp. 45–47, Oct. 2018, doi: 10.5120/ijca2018918026.