International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: https://ijwos.com OPEN ACCESS E-ISSN : 3049-2424
Volume 3 Issue 3 , March 2026
DOI: https://doi.org/10.71366/ijwos03032691564

Detection of Phishing Websites Using Machine Learning

R. Pavithra[1], P. Anitha [2]
[1] PG Student, Department of Computer Applications, Vellalar College for women, Erode, India
[2] Assistant Professor, Department of Computer Applications, Vellalar College for women, Erode, India.

Article Info

Article History:

Published:05 March 2026

Publication Issue:
Volume 3, Issue 3
March-2026

Page Number:
35-38

Corresponding Author:
R. Pavithra

Abstract:

Phishing attacks continue to be one of the most pervasive and damaging forms of cybercrime, exploiting human trust to steal sensitive information through fraudulent websites. This paper proposes a machine learning-based framework for the automatic detection of phishing websites by analyzing URL structures, domain-level attributes, and webpage content features. We evaluate four classification algorithms — Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost — on the UCI phishing websites dataset containing 11,055 labeled instances. Our experimental results demonstrate that the Random Forest classifier achieves the highest detection accuracy of 97.4%, with a precision of 0.975 and an F1-score of 0.973. Feature importance analysis reveals that URL length, presence of an IP address in the URL, and abnormal URL patterns are the most discriminative indicators of phishing. The proposed system offers a scalable, real-time solution that outperforms conventional blacklist-based approaches and generalizes well to previously unseen phishing websites.

Keywords: Phishing Detection, Machine Learning, URL Feature Analysis, Random Forest, XGBoost, Cybersecurity, Website Classification

1. Introduction

The internet has become an indispensable part of modern life, enabling banking, commerce, healthcare, and social communication at an unprecedented scale. However, this widespread digitization has simultaneously expanded the attack surface for cybercriminals. Among the many threats in the digital landscape, phishing remains one of the most dangerous and persistent. A phishing website is a fraudulent page designed to mimic a legitimate service — such as a bank, social network, or e-commerce platform — with the intention of tricking users into surrendering sensitive information like passwords, credit card numbers, or personal identification details.

Traditional defenses against phishing rely primarily on blacklists: databases of known phishing URLs maintained by security vendors and browsers. While these are effective for well-documented threats, they are inherently reactive. Cybercriminals frequently register new domains, make minor URL modifications, and take down pages within hours of launching an attack — a strategy that renders blacklist-based approaches insufficient on their own. This motivates the exploration of proactive, data-driven detection mechanisms.

Machine learning offers a compelling alternative. By learning discriminative patterns from large collections of labeled phishing and legitimate website data, ML models can generalize to detect novel phishing attempts without relying on pre-existing blacklists. This paper presents a systematic, feature-engineered ML pipeline for phishing website detection, evaluating multiple classifiers on a benchmark dataset. Our contributions are as follows:

- A feature extraction pipeline covering URL-based, domain-based, and HTML/JavaScript-based attributes.

- A comparative evaluation of four ML classifiers: Logistic Regression, SVM, Random Forest, and XGBoost.
- A feature importance analysis identifying the most discriminative predictors of phishing behavior.
- A discussion of practical deployment considerations for real-time phishing detection systems.

2. Literature Review

The problem of phishing website detection has been extensively studied in the literature. Mohammad et al. [1] introduced a widely used benchmark dataset containing 30 handcrafted features derived from URL patterns, WHOIS data, and HTML content. This dataset became a standard testbed for subsequent research and is also employed in our study.

Sahingoz et al. [2] explored seven different ML algorithms on a real-time phishing dataset and reported that Random Forest consistently achieved the best detection accuracy. Their work highlighted the importance of real-time feature extraction in operational settings, particularly the challenge of obtaining WHOIS and DNS records within acceptable latency constraints.

Deep learning approaches have also gained traction. Yerima and Alzaylaee [3] applied convolutional neural networks to character-level URL sequences, achieving competitive accuracy without manual feature engineering. However, such models typically demand more computational resources and offer less interpretability than traditional ML models, making them less suitable for lightweight browser-side deployment.

Jain and Gupta [4] proposed a client-side detection system using 17 features extractable without server interaction, achieving 96.7% accuracy. Their work underscored the trade-off between detection accuracy and feature availability under real-world constraints, a consideration that motivates our careful feature selection in this work.

Despite substantial progress, gaps remain in comparative evaluations across multiple algorithm families, transparent feature importance analysis, and practical deployment discussions. Our study addresses these gaps.

3. Methodology

3.1 Dataset

We used the UCI Machine Learning Repository Phishing Websites Dataset [5], which contains 11,055 instances: 4,898 legitimate URLs and 6,157 phishing URLs. The phishing examples were sourced from PhishTank, a community-curated repository of verified phishing sites, while legitimate URLs were drawn from the Alexa Top Sites list. Each instance is labeled as phishing (−1) or legitimate (+1). The dataset was partitioned into 80% training and 20% test sets using stratified random sampling to preserve class proportions.

3.2 Feature Extraction

We extracted a total of 30 features grouped into three categories:

- URL-Based Features (12 features): URL length, presence of an IP address as the domain, use of URL shortening services, count of dots and special characters ('@', '-', '//'), depth of directory path, HTTPS token in the domain name, length of subdomain, and presence of port number in the URL.
- Domain-Based Features (9 features): Domain age, DNS record availability, WHOIS registration data completeness, web traffic rank (Alexa), Google PageRank score, statistical report membership, favicon loaded from external domain, and SSL certificate age.
- HTML and JavaScript Features (9 features): Presence of iframe tags with hidden attributes, right-click disabling via JavaScript, proportion of suspicious links in anchor tags (pointing to external domains), meta

tag redirection, use of popup windows, onmouseover event handlers, disabling of right-click context menus, forwarding behavior count, and percentage of null hyperlinks.

All features were encoded as categorical values (−1, 0, or 1) following the encoding scheme of the original dataset, where −1 indicates a phishing signal, 0 is neutral, and 1 indicates a legitimate signal.

3.3 Machine Learning Models

We trained and evaluated four classifiers: (1) Logistic Regression as a linear baseline; (2) Support Vector Machine with an RBF kernel, chosen for its effectiveness in high-dimensional feature spaces; (3) Random Forest with 100 decision tree estimators, selected for its robustness and interpretability via feature importance; and (4) XGBoost, a gradient boosting framework known for its strong performance on tabular data.

All models were implemented in Python 3.10 using the scikit-learn (v1.3) and xgboost (v1.7) libraries. Hyperparameter optimization was performed using 5-fold stratified cross-validation combined with grid search. Evaluation metrics included accuracy, precision, recall, F1-score, and AUC-ROC.

4. Results and Discussion

4.1 Classifier Performance Comparison

Table 1 summarizes the performance of each model on the held-out test set. The Random Forest classifier achieved the highest accuracy of 97.4%, followed by XGBoost at 96.9%. Logistic Regression, while the simplest model, still delivered 92.1% accuracy — a testament to the quality of the engineered features. SVM achieved 95.3% but required significantly more training time due to the quadratic complexity of kernel computation.

Table 1: Performance Comparison of Machine Learning Classifiers

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	92.1%	0.921	0.918	0.919	0.971
SVM (RBF Kernel)	95.3%	0.953	0.951	0.952	0.985
Random Forest	**97.4%**	**0.975**	**0.973**	**0.973**	**0.994**
XGBoost	96.9%	0.970	0.968	0.969	0.993

(Highlighted row indicates best-performing model)

4.2 Feature Importance Analysis

Using the Random Forest feature importance scores (mean decrease in impurity), we identified the top five most discriminative features: (1) URL length — longer URLs are a strong indicator of phishing due to obfuscation tactics; (2) presence of IP address as the domain — legitimate services rarely use raw IP addresses in their URLs; (3) abnormal URL structure — phishing URLs often include the target brand name in subdirectories or subdomains; (4) domain age — phishing domains are typically newly registered; and (5) HTTPS status — while increasingly common among phishing sites, the SSL configuration context remains informative.

These findings are consistent with prior literature and provide actionable insights for rule-based pre-filtering in lightweight detection systems.

4.3 Discussion

The strong performance of ensemble methods, particularly Random Forest, can be attributed to their ability to capture nonlinear feature interactions and their natural robustness to class imbalance — a concern in our dataset where phishing instances slightly outnumber legitimate ones. XGBoost performed comparably with the added benefit of faster inference at test time, making it attractive for latency-sensitive deployment scenarios.

One limitation of our approach is the reliance on features that require external API calls, such as domain age (WHOIS) and PageRank. In real-time browser-side deployment, these lookups introduce latency. Future work should explore lightweight subsets of features that are locally computable without network requests, as explored by Jain and Gupta [4].

5. Conclusion

This paper presented a machine learning-based approach to phishing website detection that combines URL-based, domain-based, and HTML/JavaScript features with ensemble classifiers. Our experiments on the UCI Phishing Websites Dataset demonstrated that the Random Forest classifier achieves state-of-the-art performance with 97.4% accuracy and an F1-score of 0.973. Feature importance analysis highlighted URL structure and domain age as the most discriminative predictors, offering interpretable insights for security practitioners.

The proposed framework provides a scalable, real-time complement to conventional blacklist defenses, with the potential for integration into browser extensions, web proxies, or email security gateways. Future directions include deep learning models capable of learning directly from raw HTML, adversarial robustness evaluation against feature-manipulation attacks, and live deployment evaluations with continuously updated phishing datasets.

References

[1] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, no. 2, pp. 443–458, 2014.

[2] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[3] S. Y. Yerima and M. K. Alzaylaee, "High accuracy phishing detection based on convolutional neural networks," in Proc. 3rd Int. Conf. Computer Applications and Information Security, Riyadh, Saudi Arabia, 2020.

[4] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," Telecommunication Systems, vol. 68, no. 4, pp. 687–702, 2018.

[5] UCI Machine Learning Repository, "Phishing Websites Dataset," 2012. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites

[6] A. Ramesh and S. Vijayalakshmi, "A comprehensive survey on phishing website detection using machine learning techniques," Journal of Cybersecurity and Privacy, vol. 3, no. 1, pp. 22–39, 2023.

[7] T. Nguyen, M. Tran, and L. Nguyen, "XGBoost-based phishing URL detection with enhanced feature engineering," IEEE Access, vol. 10, pp. 31542–31555, 2022.