



PHISHING WEBSITE DETECTION USING MACHINE LEARNING

Mrs. Dhage T.S.¹, Dhiraj Dattatray Meher²

¹ Faculty Guide & Supervisor | ² Dattakala Group of Institutions, India

Corresponding author: dhirajmeher5555@gmail.com

ARTICLE ID IJREK-2026-00173	ARTICLE TYPE Research Article	RESEARCH AREA Machine Learning & Deep Learning	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 28 October 2026	REVISED 29 October 2026	ACCEPTED 28 October 2026	PUBLISHED 30 October 2026

Abstract

Phishing websites imitate legitimate online services to trick users into revealing passwords, payment details, and other sensitive information. Static blacklists are useful for known threats but may not recognize newly created or rapidly modified sites. This paper presents a proposed machine-learning framework that classifies a website URL as likely legitimate or potentially phishing by examining lexical, domain, and—where safely available—HTML-based features. The workflow includes data validation, leakage-aware preprocessing, model training, threshold selection, and user-facing risk explanations. Random Forest, Support Vector Machine, and gradient-boosting models are considered as candidate classifiers. The evaluation plan uses precision, recall, F1-score, ROC-AUC, PR-AUC, false-positive rate, and inference time, with special attention to class imbalance and temporal testing. This paper is a design proposal; it does not claim newly conducted experiments or measured performance. The intended outcome is an understandable prototype that supports safer browsing while preserving human judgment and privacy.

Keywords: Phishing detection; machine learning; malicious URL; cybersecurity; feature extraction; Random Forest; explainable A

I. INTRODUCTION

Online banking, shopping, education, and communication increasingly depend on web services. Attackers exploit this trust by creating pages and URLs that resemble genuine organizations. A user may be directed to a deceptive login page through email, messaging applications, advertisements, or social media. Successful phishing can lead to account takeover, financial loss, identity theft, and further compromise. Machine-learning detection can generalize from patterns in labeled examples instead of relying only on previously reported URLs. URL lexical features can be computed without opening a potentially dangerous site; domain and page-content features can provide additional context when collected through controlled, safe processes. No single feature or model is sufficient in every environment, so robust evaluation and cautious user messaging are essential.

II. PROBLEM STATEMENT AND OBJECTIVES

The problem is to design a practical detector that identifies suspicious websites while limiting false alarms, processing cost, and unnecessary collection of user data. Objectives are to: (1) extract useful URL and domain indicators; (2) compare candidate classifiers using consistent data splits; (3) produce interpretable risk outputs; (4) evaluate minority-class detection and runtime; and (5) document limitations and responsible deployment requirements.

III. RELATED WORK

Research on phishing detection commonly groups defenses into list-based, similarity-based, heuristic, and machine-learning approaches. Blacklists can be efficient for known malicious resources, but they depend on timely updates. Similarity methods compare a suspicious page with a known brand or legitimate site, while machine-learning methods infer patterns from URL, domain, page-content, and behavioral features. A broad IEEE Access survey reviews these categories and discusses datasets and research gaps [1]. A recent IEEE Access systematic review focuses specifically on feature engineering for phishing website detection. It groups features into URL/lexical, HTML/content, domain, behavioral, and hybrid categories, and notes ongoing challenges involving evolving attacks, dataset

limitations, high dimensionality, and efficiency [2]. Other IEEE work studies lexical analysis of URLs and evaluates conventional classifiers [3], while additional surveys review broader webpage-detection techniques, including content and visual signals [4], [5]. These sources motivate a modular design rather than dependence on one feature family.

IV. PROPOSED SYSTEM

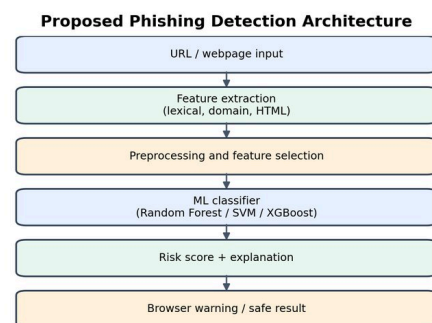


Fig. 1. Proposed system architecture for phishing website detection.

TABLE I

Module	Function	Example output
URL intake	Normalize and validate URL syntax	Normalized URL string
Feature extractor	Compute lexical and optional domain/page features	Numeric feature vector
Classifier	Estimate phishing likelihood	Probability or class score
Explanation layer	Show major contributing indicators	Suspicious tokens, unusual length
User interface	Present cautionary result	Likely safe / suspicious / uncertain

V. FEATURE ENGINEERING

TABLE II

Feature family	Illustrative attributes	Strength / caution
Lexical URL	Length, special characters, token count, digit ratio	Fast; attackers can obfuscate patterns
Host/domain	Subdomain depth, domain age when available, DNS context	Adds context; external lookups may fail or be stale
HTML/content	Form destinations, iframe use, page text indicators	Potentially useful; requires safe page retrieval
Reputation	Known blacklist or trusted-list match	Strong for known URLs; misses new domains
Behavioral/visual	Redirect sequence, brand similarity	Can improve context; higher cost and privacy risk

VI. DETECTION WORKFLOW

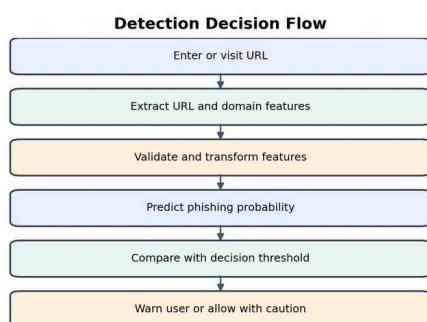


Fig. 2. Detection and user-notification flowchart.

VII. IMPLEMENTATION METHODOLOGY

A prototype can be implemented in Python using pandas for tabular data, scikit-learn for preprocessing and classification, and a lightweight web interface such as Streamlit. A reproducible pipeline should combine preprocessing and model fitting so the same operations are applied consistently during training and inference. Model versions, feature definitions, random seeds, and dataset versions should be recorded.

A. Model Development Chain

Model Development Chain

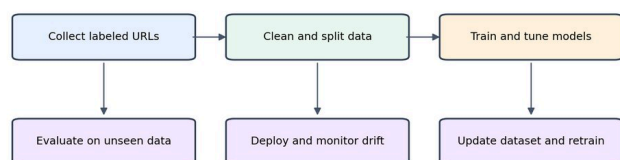


Fig. 3. End-to-end development and maintenance chain.

TABLE III

Metric	Purpose	Interpretation
Precision	Share of flagged URLs that are phishing	Higher precision means fewer false alarms among alerts
Recall	Share of phishing URLs detected	Higher recall means fewer missed phishing URLs
F1-score	Harmonic mean of precision and recall	Useful summary when both matter
PR-AUC	Precision-recall trade-off	Especially informative for imbalanced data
False-positive rate	Legitimate URLs incorrectly flagged	Important for user trust
Inference time	Time needed for a prediction	Relevant for browser integration

VIII. SECURITY AND PRIVACY

The system should avoid storing complete browsing histories unless essential and explicitly disclosed. Logs should minimize personal information, protect access, and use retention limits. Suspicious pages must not be opened in an ordinary authenticated browser during data collection. The model and feature pipeline should be protected against tampering, and users should be able to report false alarms. A phishing label should be treated as a risk estimate rather than a legal or factual determination.

IX. EXPECTED OUTCOMES AND DISCUSSION

TABLE IV

Approach	Advantages	Limitations to examine
Blacklist lookup	Fast and easy to explain	New or short-lived URLs may be absent
Heuristic rules	Transparent, low compute cost	Rules can become outdated
Logistic Regression	Simple baseline, relatively interpretable	May miss complex relationships
Random Forest	Handles non-linear feature interactions	Model size and calibration need testing
SVM / boosting	Can capture complex decision boundaries	Tuning and runtime costs vary

X. LIMITATIONS

The proposal depends on the quality and representativeness of available labeled datasets. Dataset labels can be outdated, and random splits can overestimate generalization when similar URLs occur in both training and testing. Domain-age and reputation features may require external services and can be unavailable or delayed. HTML-based analysis adds collection risk and processing overhead. Finally, an offline test does not prove effectiveness against all future phishing campaigns; continuous evaluation and independent security review are necessary.

XI. ETHICAL AND RESPONSIBLE USE

The goal is defensive detection and user protection. Dataset acquisition should respect provider terms, applicable law, and institutional guidance. Collection should avoid accessing private accounts or submitting information to suspicious pages. When students or users participate in usability testing, consent and data minimization should be applied. Model decisions should not be presented as certain when the evidence is weak. Accessibility and language clarity matter because a warning that users cannot understand may fail to protect them.

XII. FUTURE WORK

Future work can examine hybrid URL and page-content features, explainability tools, calibrated confidence estimates, and temporal evaluation across multiple sources. A browser extension could be evaluated in a controlled pilot. Additional research should measure latency, memory use, and false-positive impact on real browsing tasks. Adversarial testing can investigate how small URL changes affect predictions, while privacy-preserving aggregation may help improve models without centralizing raw browsing histories.

XIII. CONCLUSION

This paper proposed a machine-learning framework for phishing website detection using URL, domain, and optional page-content features. The architecture includes data preparation, candidate classifiers, risk scoring, explanations, and user-facing warnings. The methodology emphasizes leakage-aware splits, class-sensitive metrics, temporal testing, and privacy-conscious collection. Its practical value must be established through implementation and controlled evaluation; no experimental results are claimed in this design paper.

REFERENCES

- [1] G. L. S. et al., “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] Y. A. Kustiawan and K. I. Ghauth, “Feature Engineering for Phishing Website Detection Using Machine Learning: A Systematic Review,” IEEE Access, vol. 13, pp. 192080–192104, 2025, doi: 10.1109/ACCESS.2025.3630334.
- [3] “URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis,” in Proc. 12th Int. Conf. Information and Communication Systems (ICICS), 2021, doi: 10.1109/ICICS52457.2021.9464539.
- [4] “Phishing Webpage Detection: Unveiling the Threat Landscape and Investigating Detection Techniques,” IEEE Communications Surveys & Tutorials, vol. 27, no. 2, pp. 974–1007, 2025, doi: 10.1109/COMST.2024.3441752.
- [5] “Exploring Techniques for Web Phishing Detection: A Comprehensive Survey,” in Proc. IEEE ICICCS, 2023, doi: 10.1109/ICICCS56967.2023.10142655.
- [6] “Survey on Detection and Prevention of Phishing Websites using Machine Learning,” in Proc. IEEE ICACITE, 2021, doi: 10.1109/ICACITE51222.2021.9404714.
- [7] “Across the Spectrum In-Depth Review AI-Based Models for Phishing Detection,” IEEE Open Journal of the Communications Society, vol. 6, pp. 2065–2089, 2025, doi: 10.1109/OJCOMS.2024.3462503.
- [8] “Think Before You Click—Malicious URL Detection in Cybersecurity: A Systematic Review and Research Roadmap,” IEEE Access, vol. 13, pp. 154305–154325, 2025, doi: 10.1109/ACCESS.2025.3601387.
- [9] “Comparative Evaluation of Machine Learning Models for Phishing Website Detection Using URL-Based Features,” IEEE Communications Standards Magazine, vol. 10, no. 3, 2026.
- [10] Scikit-learn Developers, “scikit-learn: Machine Learning in Python,” software documentation. <https://scikit-learn.org/stable/>