INTERNATIONAL JOURNAL OF RESEARCH, EXPLORATION & KNOWLEDGE (IJREK)
Official Scholarly Journal Portal: **https://www.ijrek.org** | DOI: **https://doi.org/10.5281/zenodo.23482025**

AI-POWERED PHISHING DETECTION

Mrs.Dhage T.S.[1], Aayush shimpi[2]

[1] Faculty Guide & Supervisor | [2] Dattakala Group Of Institutions Faculty Of Engineering Swami - Chincholi tal. Daund dist. Pune, India

Corresponding author: **shimpiaayush79@gmail.com**

ARTICLE ID	ARTICLE TYPE	RESEARCH AREA	PUBLICATION
IJREK-2026-00095	Research Article	Computer Science	Volume 1, Issue Issue 1 (2026)
RECEIVED	**REVISED**	**ACCEPTED**	**PUBLISHED**
19 October 2026	02 November 2026	19 October 2026	02 November 2026

Abstract

Phishing is a social-engineering threat in which attackers impersonate trusted organisations or individuals to persuade users to disclose information, open harmful attachments, or visit deceptive websites. Conventional filters can be challenged by newly created domains, changing message patterns, and carefully crafted text. This paper proposes an AI-Powered Phishing Detection system that analyses selected email, URL, and message features to estimate whether an item may be suspicious. The workflow includes input validation, feature extraction, machine-learning classification, risk scoring, explanation of influential indicators, and a user-facing warning or review action. Candidate features include URL structure, domain signals, message wording, sender metadata, and selected email-header indicators where available. The system is a defensive screening aid, not a guarantee that a message is safe or malicious. The paper describes architecture, workflow, implementation, evaluation, applications, and limitations. Diagrams and chart values are illustrative, not experimental results.

Keywords: *Phishing detection; artificial intelligence; machine learning; cybersecurity; malicious URL; email security; classification; threat detection*

2. PROBLEM STATEMENT AND OBJECTIVES

The problem is to identify potentially deceptive emails, URLs, or messages early enough to warn users while limiting incorrect classifications. Static rules can miss new variants, while an AI model can fail when input differs from training data. The proposed system combines feature-based analysis with a clear risk indicator and an opportunity for human review. The objectives are: • Analyse selected email, URL, and message attributes associated with phishing. • Prepare and validate input features for machine-learning classification. • Compare baseline and candidate models using labelled data. • Display a risk score with understandable warning indicators. • Evaluate false positives, false negatives, precision, recall, and response time. • Protect user data and support safe, human-reviewed handling of suspicious items.

3. LITERATURE REVIEW

Phishing detection research has used blacklists, heuristic rules, supervised machine learning, and deep-learning methods. Blacklists can block known malicious domains quickly but may not recognise newly registered or compromised sites. Heuristic systems flag suspicious patterns such as misleading hostnames or unusual URL structure, but attackers can adapt to fixed rules. Models such as logistic regression, decision trees, random forests, support vector machines, and gradient boosting can classify messages from labelled features. Text-based methods use token patterns and linguistic signals, while URL models examine hostname, path, length, special characters, and other structural properties. Hybrid systems combine these signals, though quality depends on data and evaluation. A recurring challenge is dataset drift: legitimate and malicious message patterns change over time. Datasets may contain duplicates, outdated samples, or unrealistic class balance. Evaluation should separate training and test data carefully and, where possible, include a time-based test split. This paper focuses on a defensive pipeline and evaluation plan rather than claiming unmeasured accuracy.

4. SCOPE AND ASSUMPTIONS

The prototype flags potentially suspicious messages and URLs for review. It should not automatically delete messages or make decisions based solely on one score. Header or domain-reputation features depend on available data and permissions. The system should not visit suspicious links automatically or expose users to harmful content during analysis.

5. PROPOSED SYSTEM ARCHITECTURE

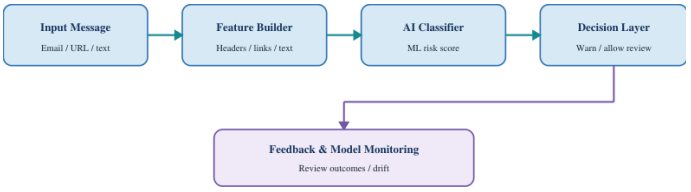


Figure 1. Proposed architecture of the AI-powered phishing detection system.

5.1 INPUT LAYER

The input layer accepts a URL, email text, or safely obtained message record. File uploads and email integrations should be validated, size-limited, and handled with least privilege. The system should avoid opening links or attachments as part of routine classification.

5.2 FEATURE EXTRACTION

Potential URL features include hostname structure, URL length, unusual symbols, IP-address use, suspicious subdomain patterns, and differences between visible link text and destination when both are available. Message features may include urgent wording, credential requests, impersonation cues, and suspicious link counts. Header and domain-age features should be added only when trustworthy data are available. No single feature proves phishing.

5.3 CLASSIFICATION ENGINE

The model processes prepared features and returns a score or class such as suspicious or likely legitimate. A baseline such as logistic regression or a decision tree can be compared with random forest or gradient boosting. The final model should be

selected using held-out evaluation, computational cost, interpretability, and the consequences of errors.

5.4 DECISION AND MONITORING LAYER

The interface can display a warning, list key indicators, and recommend verifying the sender through an independent channel. Minimal decision metadata can be logged for auditing. Feedback and reviewed outcomes can inform future evaluation and drift monitoring.

6. DETECTION WORKFLOW AND METHODOLOGY

MESSAGE ANALYSIS WORKFLOW 1. Receive Email / URL 2. Extract Text / metadata 3. Predict Risk estimate 4. Explain Key indicators 5. Respond Warn / review Human review and feedback support ongoing monitoring

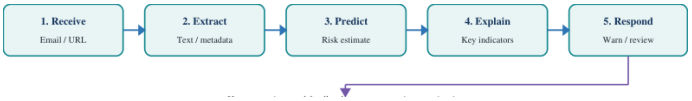


Figure 2. Defensive workflow from input analysis to user review.

6.1 DATASET PREPARATION

Use a documented, authorised dataset containing phishing and legitimate examples. Remove duplicates and inspect label quality. Split data so near-identical messages or URLs do not appear across partitions. A time-based test set can better represent deployment conditions. Avoid features that would only become available after the detection decision.

6.2 MODEL TRAINING

Train a simple baseline and compare candidate models using the validation set. Tune parameters without using the final test set. If classes are imbalanced, report per-class metrics and consider appropriate weighting. Select warning thresholds based on validation results and the operational cost of missed phishing versus false alarms.

6.3 RISK SCORE AND EXPLANATION

A risk score can combine model output with selected policy rules, but its meaning must be defined. The interface may explain that a URL has an unusual hostname or a message requests credentials urgently, provided those signals were observed. A model score is not a calibrated probability unless calibration has been tested.

6.4 SAFE USER RESPONSE

For a suspicious result, recommend not entering credentials, not opening attachments, and verifying the request using an official website or known contact method. Provide a false-positive review option. Do not ask users to interact with a suspected link to test it.

6.5 PRIVACY AND SECURITY

Email bodies, addresses, and URLs may contain confidential information or tokens. Minimise collection, redact sensitive fields where possible, restrict access, and define retention periods. Protect model endpoints against abuse and validate inputs to prevent injection or resource-exhaustion problems.

7. RESULT VISUALISATION AND INTERFACE DESIGN

The interface should distinguish a model estimate from a confirmed security verdict. A clear label, risk indicator, and short explanation can help users decide when to seek further review. The system should communicate uncertainty and provide safe next steps rather than relying only on colour. ILLUSTRATIVE CLASSIFIER SCORES Email A URL B Email C URL D Email E Model score

Figure 3. Illustrative classifier scores for interface design only; these are not measured results.

7.1 SUGGESTED INTERFACE COMPONENTS

• Input field for a URL or message, with safe validation. • Result label such as “Potentially suspicious” or “No strong indicators found.” • Explanation of observed warning indicators. • Timestamp and model version for auditability. • Safe next-step guidance and a false-positive feedback option. • Privacy information and controls for deleting submitted content.

7.2 EVALUATION METRICS

Precision measures how many flagged items are phishing within the labelled test set; recall measures how many phishing examples are detected. F1-score balances precision and recall, while false-positive rate indicates how often legitimate items are incorrectly flagged. ROC-AUC or precision-recall AUC may be useful depending on class balance. Latency and resource usage should also be measured.

8. IMPLEMENTATION PLAN

Table 1. Proposed modules and responsibilities

Module	Function	Key consideration
Input handling	Receive URL or message	Validation and safe parsing
Feature extraction	Build text, URL, and metadata features	Consistency and data quality
Classifier	Estimate phishing risk	Held-out testing
Decision layer	Set warning or review status	Threshold trade-offs
User interface	Show result and indicators	Clear safe next steps
Monitoring	Track feedback and drift	Privacy and model updates

8.1 POSSIBLE TECHNOLOGY STACK

Python can support preprocessing and model development with standard machine-learning libraries. A lightweight API may expose classification results to a web interface, while structured logs store minimal audit information. The selected stack should follow institutional resources and security requirements. 9. Applications Potential applications include email security assistants, URL screening tools, awareness-training demonstrations, and internal message triage. The system can help users prioritise suspicious items, but should complement established controls such as multi-factor authentication, email authentication, endpoint protection, and user education rather than replacing them.

10. LIMITATIONS AND ETHICAL CONSIDERATIONS

Phishing campaigns evolve rapidly, and a model trained on older examples may fail on new techniques. Attackers can obfuscate URLs, compromise legitimate websites, or write messages that resemble normal communication. Legitimate messages may also be falsely flagged. Dataset bias, class imbalance, and changing deployment conditions can reduce performance. A “no strong indicators found” result must not be presented as proof that a message is safe. Sensitive message content should be protected, and the system should retain only what is necessary. Model updates require validation and rollback plans. Human review should remain available for uncertain cases, and the detection output should be one signal within a broader security process.

11. EXPECTED OUTCOMES

The intended outcome is a prototype that analyses supported message or URL inputs, returns a risk estimate, explains selected indicators, and provides safe next steps. Its effectiveness must be established through a documented test dataset and reproducible evaluation; no experimental performance figure is claimed in this paper.

12. FUTURE SCOPE

Future work may explore multilingual phishing detection, improved domain-reputation signals, attachment metadata analysis in a sandboxed environment, browser integration, and continual evaluation against emerging threats. External data sources should be authorised and their freshness documented. Privacy-preserving processing and robust explanations may also improve trust and maintainability. 13. Conclusion This paper proposes an AI-Powered Phishing Detection system combining message and URL feature extraction, machine-learning classification, risk scoring, and user-facing explanations. The design aims to identify suspicious indicators and guide users toward safer decisions while acknowledging false positives and false negatives. A reliable implementation requires representative data,

leakage-resistant evaluation, threshold selection based on operational needs, privacy protection, and ongoing monitoring. The system should complement, not replace, established cybersecurity controls and human judgement.

REFERENCES

[1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[2] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in Proceedings of the Anti-Phishing Working Groups eCrime Researchers Summit, 2007.

[3] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, pp. 443–458, 2014.

[4] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: A feature-rich machine learning framework for detecting phishing web sites," ACM Transactions on Information and System Security, vol. 14, no. 2, 2011.

[5] Anti-Phishing Working Group, Phishing Activity Trends Reports. APWG, periodic reports and resources.

[6] National Institute of Standards and Technology, Cybersecurity Framework 2.0. NIST, 2024.

[7] M. 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.

[8] Internet Engineering Task Force, Domain-based Message Authentication, Reporting, and Conformance (DMARC), RFC 7489, 2015.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on **https://www.ijrek.org** | Article ID: IJREK-2026-00095