IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

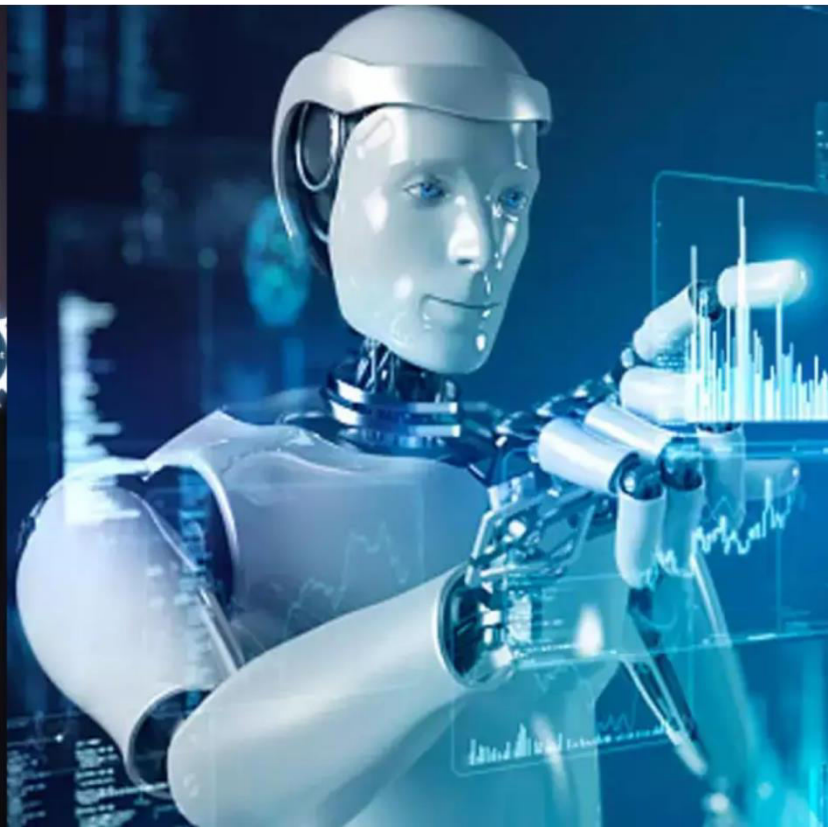
ISSN(O): 2319-8753
ISSN(P): 2347-6710



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 9.935

Volume 15, Issue 10, October 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|

XGboost for Phishing Detection

Vedant Khawale, Samiksha Puri, Kasturi Teware, Sarthak Yende

Department of Information Technology, Prof Rammeghe Institute of Technology & Research, Badnera, Maharashtra, India

ABSTRACT: PhishGuard AI is a machine learning-based system designed to detect phishing websites and protect users from online threats. It uses URL feature analysis and the XGBoost algorithm to identify suspicious websites. The system also includes brand similarity detection, SSL verification, and browser-level protection. It aims to prevent data theft and provide safer online browsing.

KEYWORDS: Phishing Detection, Machine Learning, XGBoost, URL Analysis, Cybersecurity, SSL Verification.

I. INTRODUCTION

1.1 Problem Statement

Phishing attacks are a major cybersecurity threat in which attackers create fake websites resembling legitimate banking, e-commerce, and business platforms to steal users' credentials and personal information. Traditional security systems mainly depend on blacklists of known malicious URLs. However, newly created phishing websites can remain undetected until they are identified and added to these lists. This delay exposes users to zero-day phishing attacks and increases the risk of data theft and financial loss. Therefore, an intelligent, real-time phishing detection system is required to identify suspicious URLs before users become victims.

1.2 Motivation

PhishGuard AI aims to overcome the limitations of traditional blacklist-based security systems by providing proactive, real-time phishing detection. It combines machine learning, URL feature extraction, brand similarity analysis, SSL certificate verification, and browser-based form monitoring. The system is designed to identify suspicious websites and prevent users from submitting sensitive information to potentially malicious domains.

1.3 Objectives

The main objectives of PhishGuard AI are:

- URL Feature Extraction: Extract 55 lexical and structural features from URLs to identify suspicious patterns.
- Machine Learning Detection: Use an XGBoost classification model to distinguish legitimate URLs from phishing URLs.
- Brand Similarity Detection: Apply Levenshtein distance to identify fake websites that imitate trusted brands.
- SSL Certificate Verification: Check certificate validity and identify potential security issues.
- Browser Protection: Develop a Chrome extension that monitors suspicious websites and intercepts risky form submissions.
- Threat Intelligence: Use MongoDB to store scan results, user feedback, and confirmed threats for continuous improvement.

1.4 Organization of Seminar Report

This seminar report is organized into nine chapters. Chapter 1 introduces the problem, motivation, and objectives. Chapter 2 presents the literature survey and comparative analysis of existing research. Chapter 3 explains the methodology and technology stack. Chapter 4 describes the system design and architecture. Chapter 5 discusses implementation details. Chapter 6 presents results, demonstrations, and performance evaluation. Chapter 7 summarizes the conclusions. Chapter 8 discusses the advantages and limitations. Chapter 9 describes practical applications and future enhancements.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|

II. LITERATURE SURVEY

Research on malicious URL and phishing detection has advanced significantly with the development of machine learning and deep learning techniques. Researchers have explored URL feature extraction, neural networks, ensemble classifiers, and threat intelligence to improve detection accuracy and identify emerging phishing attacks. This chapter summarizes important research contributions and highlights the improvements introduced by PhishGuard AI.

2.1 Review of Related Literature

Sahoo, Liu, and Hoi (2017) presented a comprehensive survey of machine learning techniques for malicious URL detection, discussing feature-based classification and deep learning approaches [2]. Sahingoz et al. (2019) developed a URL-based phishing detection system using lexical features and machine learning classifiers [3]. Dutta (2021) explored recurrent neural networks and LSTM models for detecting phishing websites without depending entirely on static blacklists [1].

Ahammad et al. (2022) evaluated different machine learning methods for phishing URL detection, while Alani and Tawfik (2022) proposed a cloud-based detection approach [4], [5]. Alnemari and Alshammari (2023) compared classifiers such as Decision Tree, SVM, ANN, and Random Forest [7].

Recent studies have investigated deep learning, feature optimization, domain adaptation, and explainable AI. Do et al. (2024) explored transformer-based phishing detection [12], while Saranya et al. (2024) investigated explainable AI techniques for improving detection transparency [14]. Prabakaran and Kumar (2024) applied gradient boosting to multi-class malicious URL classification [15]. Research published in 2025 and 2026 further explored ensemble learning, advanced URL feature engineering, and multistage verification methods [17], [21], [23].

2.2 Literature Survey Overview

Existing studies demonstrate the effectiveness of machine learning and deep learning in identifying phishing URLs. However, challenges remain in detecting newly created threats, reducing inference latency, identifying brand impersonation, validating SSL certificates, and preventing sensitive information submission.

PhishGuard AI addresses these limitations through a hybrid architecture incorporating 55-dimensional URL feature extraction, XGBoost classification, fuzzy brand matching, SSL certificate verification, and a Chrome extension with DOM-based form interception. Fast local analysis, asynchronous backend operations, and MongoDB-based threat feedback support responsive detection and continuous improvement.

III. METHODOLOGY

3.1 System Concept

PhishGuard AI is a hybrid, multi-layered phishing detection system that combines browser-side monitoring with a FastAPI backend. The browser extension tracks navigation, maintains local caches, and monitors webpage forms, while the backend performs URL feature extraction, machine learning classification, brand similarity analysis, and threat verification. This architecture enables real-time detection of suspicious URLs and helps prevent credential theft.

3.2 Proposed System

The proposed system consists of five major modules:

1. URL Feature Extraction: Extracts 55 numerical features, including URL length, character patterns, entropy, subdomain depth, and suspicious keywords.
2. Machine Learning Engine: Uses an XGBoost classifier to predict whether a URL is legitimate or malicious.
3. Brand Similarity Detection: Uses RapidFuzz and Levenshtein distance to identify domains that imitate trusted brands.
4. SSL Verification: Checks certificate validity, issuer information, and expiration status.
5. Form Interception: Monitors password fields and external form actions to identify potentially unsafe credential submissions.

3.3 Threat Scoring Logic

PhishGuard AI calculates a risk score between 0 and 100 using multiple security indicators. The score considers machine learning confidence, suspicious keywords, brand impersonation, HTTP usage, SSL certificate issues, webpage

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|

form vulnerabilities, and external threat intelligence. Additional penalties are applied to suspicious indicators, while confirmed threats receive the highest risk score. Based on the final score, websites are classified as Safe, Suspicious, or Dangerous.

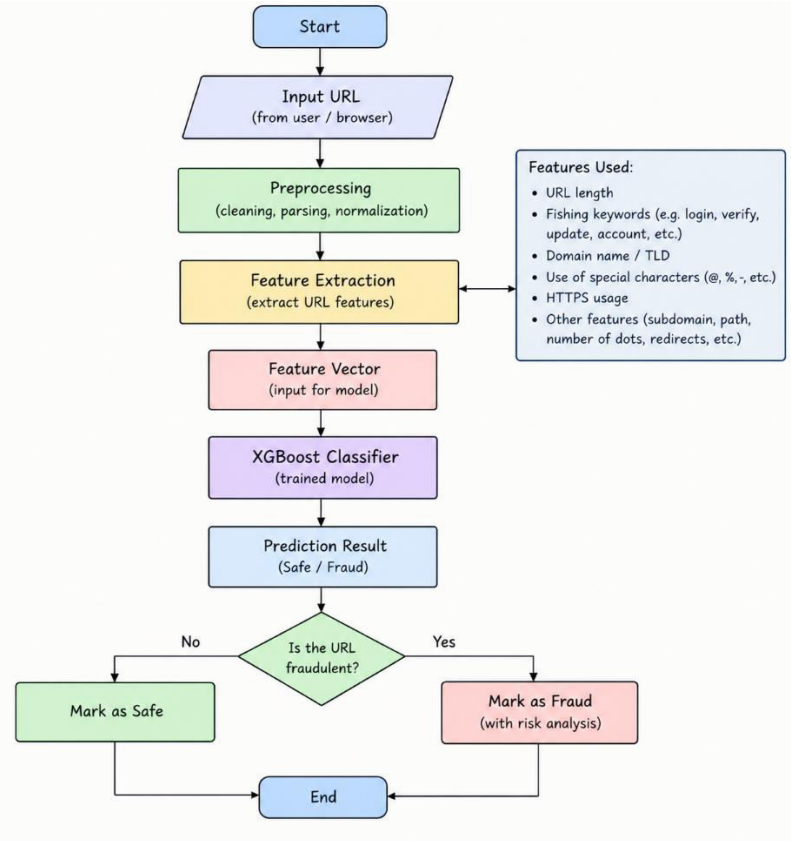
3.4 Technology Stack

The system uses the following technologies:

- Browser Extension: JavaScript, Manifest V3, HTML5, CSS3.
- Admin Dashboard: React.js, Vite, Tailwind CSS, Axios.
- Backend: Python, FastAPI, Uvicorn, Pydantic.
- Machine Learning: XGBoost, Scikit-learn, RapidFuzz, NumPy, Pandas.
- Database and Caching: MongoDB, Motor asynchronous driver, and TTL-based caching.

This combination of machine learning, browser monitoring, and threat intelligence provides an integrated approach to real-time phishing detection and user protection.

IV. SYSTEM DESIGN AND ARCHITECTURE



4.1 Four-Tier System Architecture Overview

PhishGuard AI follows a four-tier architecture to provide real-time phishing detection. The Client Tier uses a Chrome extension for URL monitoring, caching, warnings, and form interception. The Application Gateway Tier uses FastAPI for request validation, rate limiting, and secure communication. The Analysis Tier performs URL feature extraction, XGBoost classification, brand similarity matching, SSL verification, and reputation checks. The Data Tier uses MongoDB to store scan history, user feedback, trusted domains, and confirmed threats.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|

4.2 Component-Level Architectural Design
The browser extension sends URL information to the FastAPI backend through HTTPS REST APIs. The backend analyzes the URL using machine learning and security checks, calculates a risk score, and stores scan records in MongoDB. The result is returned to the extension, which displays the appropriate warning or blocking interface.

4.3 Active Protection Modes and Risk State Machine
PhishGuard AI provides three protection modes:

- Relaxed: Performs background scanning with limited alerts.
- Standard: Scans URLs and displays warnings for suspicious or dangerous websites.
- Paranoid: Applies strict checks, aggressive warnings, and form-submission interception for risky websites.

Websites are classified as Safe, Suspicious, or Dangerous according to their calculated risk scores.

4.4 Pre-Click Scanner and Form Interception
The pre-click scanner checks links when users hover over them, uses cached results when available, and displays risk information through a tooltip. The form interceptor monitors password fields and external form destinations. When a risky submission is detected, it pauses submission and displays a warning, allowing users to cancel or explicitly choose to proceed.

4.5 Detailed Sequence Architecture
The system operates through six steps: (1) detect browser navigation, (2) check trusted domains and cached results, (3) send unverified URLs to FastAPI, (4) perform machine learning and security checks, (5) calculate the risk score and store scan details, and (6) display the result and apply the selected protection mode.

4.6 Technology Stack Justification
The system uses JavaScript and Manifest V3 for browser protection, FastAPI and Uvicorn for asynchronous backend services, XGBoost and RapidFuzz for URL classification and brand matching, MongoDB and Motor for data persistence, and React.js with Vite for the administration dashboard. These technologies support modular development, efficient processing, and maintainable system integration.

V. IMPLEMENTATION

5.1 Overview
PhishGuard AI implements a hybrid phishing detection system using Python, machine learning, a Chrome extension, FastAPI, React.js, and MongoDB. Its modules work together to analyze URLs, identify threats, and protect users from phishing attacks.

5.2 URL Feature Extraction (feature_extractor.py)
The feature extraction module generates 55 lexical and structural features without requiring DNS or WHOIS queries. These include URL length, hostname structure, special characters, digit ratios, suspicious keywords, domain patterns, brand references, and URL redirections. Exception handling ensures malformed port values do not crash the application.

5.3 Machine Learning Engine (predictor.py)
The system uses an XGBoost classifier to predict whether a URL is legitimate or malicious. Joblib loads the trained model, while Pandas organizes features in the required order. A heuristic fallback provides basic detection when the model files are unavailable.

5.4 Brand Similarity Matcher (similarity.py)
The RapidFuzz library compares domain names with a predefined list of trusted brands using Levenshtein similarity. It identifies potential typosquatting and brand impersonation attempts and contributes to the overall risk score.

5.5 SSL Verification (ssl_verifier.py)
The SSL verification module checks HTTPS certificates for expiration, issuer information, and validity. Blocking socket operations are moved to worker threads to maintain asynchronous backend responsiveness.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|

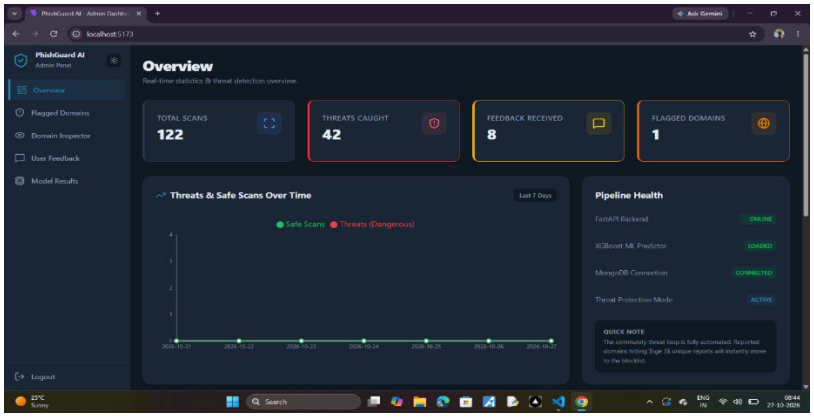
5.6 Google Safe Browsing Integration (safe_browsing.py)
The system queries Google Safe Browsing to identify known phishing and malware threats. Reusable asynchronous HTTP connections reduce connection overhead, and confirmed matches increase the calculated risk score.

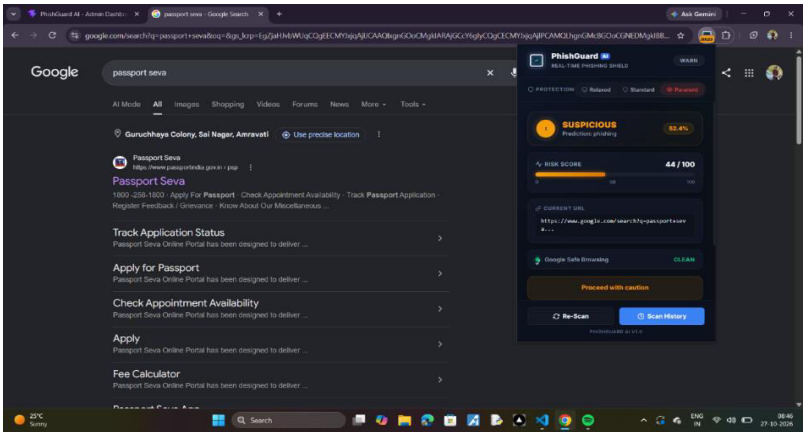
5.7 Chrome Extension (extension/)
The Manifest V3 extension monitors browser navigation, caches scan results, displays risk indicators, and scans links before users click them. Content scripts monitor suspicious form submissions and display warnings or blocking dialogs. HTML sanitization helps prevent cross-site scripting vulnerabilities in warning messages.

5.8 React.js Admin Console (admin/)
The administration dashboard uses React.js, Vite, and Tailwind CSS to display scan statistics, threat trends, and domain analysis. Administrators can inspect URL features, review predictions, manage trusted and flagged domains, and examine user feedback.

5.9 MongoDB Persistence (db.py)
MongoDB stores scan history, feedback, flagged domains, and trusted domains through the asynchronous Motor driver. Indexes improve query performance, while a 90-day TTL index automatically removes expired scan records. Atomic updates support concurrent threat reports, and parallel analytics queries improve dashboard responsiveness.
Overall, the implementation integrates machine learning, browser-level protection, certificate verification, and threat intelligence into a unified phishing detection platform.

VI. OUTPUT





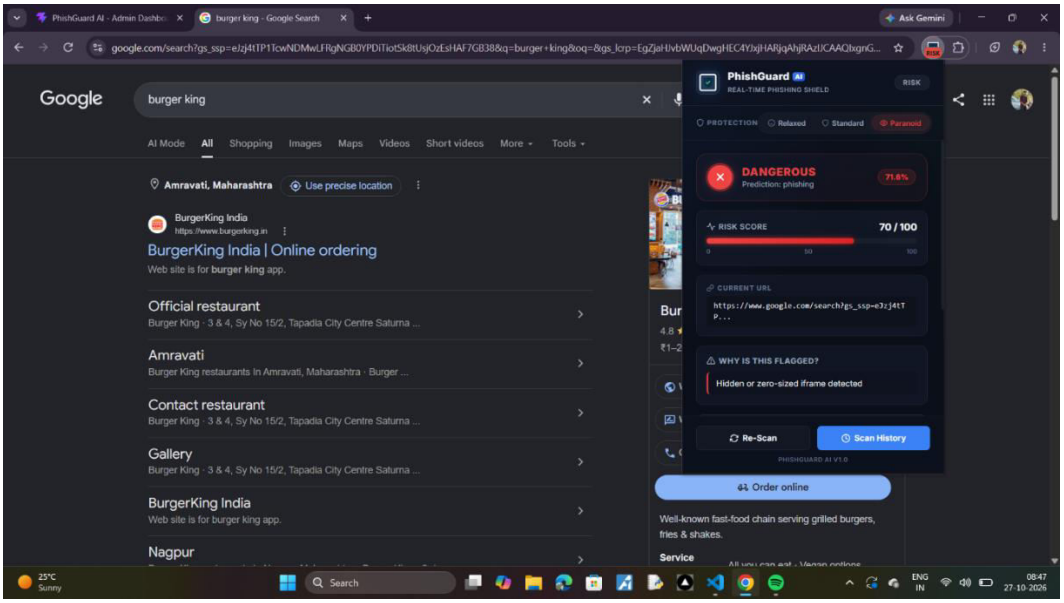
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

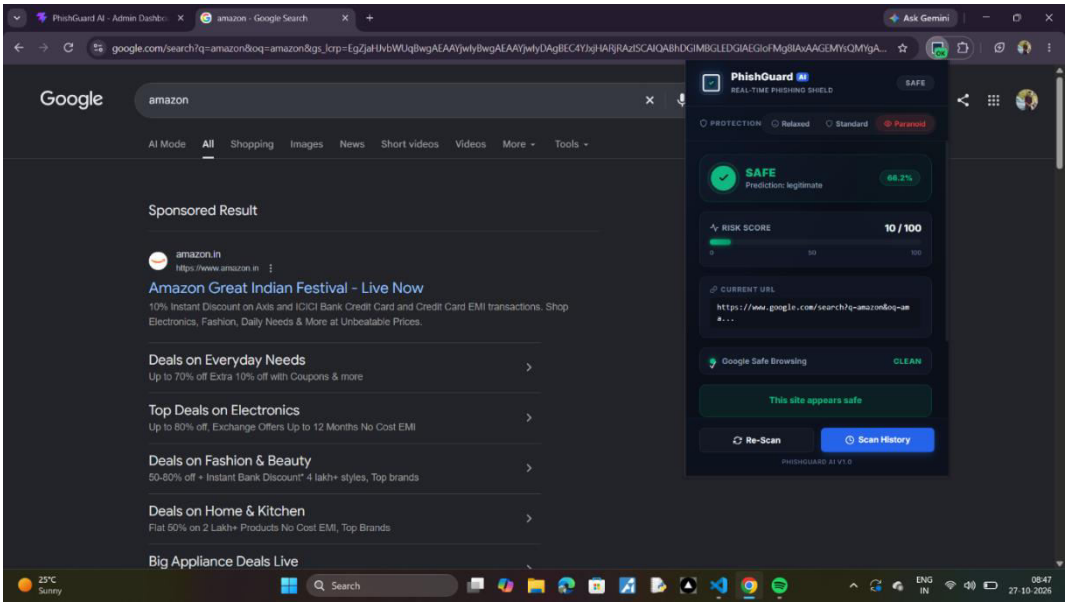
Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|





VIII. CONCLUSION

PhishGuard AI is a machine learning-based anti-phishing system that uses 55 URL features, XGBoost classification, brand similarity detection, SSL verification, and browser protection to identify suspicious websites. It protects users by warning them before they submit sensitive information, improves scanning speed through asynchronous processing, and stores user feedback and reported domains in MongoDB. Overall, PhishGuard AI combines machine learning, browser security, and threat intelligence to improve phishing detection and provide safer online browsing.The system can be further enhanced through continuous model training, updated threat databases, and broader real-world testing.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510100|

IX. ADVANTAGES AND DISADVANTAGES

9.1 Advantages

- Early Phishing Detection: Uses 55 URL features and machine learning to identify suspicious websites, including potential new phishing threats.
- Credential Protection: Intercepts suspicious form submissions and warns users before they share sensitive information.
- Brand Similarity Detection: Identifies look-alike domains and possible typosquatting attacks using fuzzy matching.
- Fast Scanning: Uses caching and efficient model inference to reduce scanning delays.
- Community Threat Intelligence: Collects user feedback to help maintain updated lists of suspicious and trusted domains.
- Administrative Dashboard: Provides scan statistics, domain analysis, threat monitoring, and domain management features.

9.2 Disadvantages and Limitations

- Internet Dependency: Scanning new URLs requires a network connection to the backend, although recent results may be cached.
- Obfuscated Web Pages: Complex JavaScript and dynamically generated form elements may make some phishing pages difficult to analyze.
- External API Limits: Google Safe Browsing checks depend on API availability and usage quotas, which may affect large volumes of requests.

REFERENCES

[1] A. K. Dutta, "Detecting phishing websites using machine learning technique," *PLOS ONE*, vol. 16, no. 10, 2021.
[2] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," *arXiv preprint arXiv:1701.07179*, 2017.
[3] O. K. Sahingoz et al., "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.
[4] S. K. H. Ahammad et al., "Phishing URL detection using machine learning methods," *Advances in Engineering Software*, vol. 173, 2022.
[5] M. M. Alani and H. Tawfik, "PhishNot: A cloud-based machine-learning approach to phishing URL detection," *Computer Networks*, vol. 215, 2022.
[6] S. Alnemari and M. Alshammari, "Detecting phishing domains using machine learning," *Applied Sciences*, vol. 13, no. 8, 2023.
[7] 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, 2023.
[8] R. Al-Tawil and K. M. Hosny, "Optimized metaheuristic deep learning architectures for zero-day phishing attack prevention," *Information Fusion*, vol. 97, 2023.
[9] N. Reyes-Dorta et al., "Detection of malicious URLs using machine learning," *Wireless Networks*, vol. 30, pp. 7543–7560, 2024



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details