International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

PhishGuard – AI Phishing Email & URL Detector

[1] Dr. SRINIVASARAO PALLAPU , [2] PALAMARRI MOUNIKA , [3] AKULA VISHANVI, [4] PAMULA RISHITHA

[1] Associate Professor, Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India

[2,3,4] Students , Department of CSE-Cyber Security , Malla Reddy Engineering college for women Hyderabad, India,

[2] Email:palamarrimounika11@gmail.com, [3] Email: vishnaviakula1996@gmail.com , [4] Email :rishithapamula13@gmail.com

Abstract---phishguard ai phishing email URL detector is a state-of-the-art cybersecurity solution designed to instantly detect and thwart phishing attacks using machine learning ml and natural language processing(NLP)techniques [1],[3],[4]. It analyzes emails and urls for evidence of malicious behavior by evaluating sender reputation url encoding,embedded link destinations and the content of the email[5]. Phishguard can readily differentiate between authentic and deceptive messages by virtue of operating on an adaptive learning model the technology achieves higher accuracy through the new data it receives and the evolving landscape of modern phishing threats[4],[6]. Phishguard integrates seamlessly with leading email products and web browsers providing immediate alerts and risk scores to the user[7],[8].This AI-based technology is built to reduce human intervention and improve protection against credential harvesting data breaches and socially engineered conflicts phishguard provides a powerful scalable and automated response to protect users against ever-evolving phishing attacks.

Keywords:Phishing Detection ,Artificial Intelligence (AI) ,Machine Learning, Cybersecurity ,Email Security ,URL Analysis ,Threat IntelligenceReal-Time Detection, Flask Framework ,ReactJS Frontend, Data Visualization, Natural Language Processing (NLP),Feature Extraction ,Web Security, Deep Learning

Received: 10-07-2025 Accepted: 19-08-2025 Published: 26-08-2025

I. INTRODUCTION

Phishing is still among the most prevalent and devastating cyber attacks, where spoofed e-mails and fake URLs deceive victims into revealing sensitive data such as login credentials, bank account details, and personal information [1], [10], [13]. With greater dependence on internet communication and commerce, individuals are vulnerable to such attacks to an extent never seen before, with adversaries capitalizing on human faith owing to clever social engineering and spoofing methods[12],[13]. Despite the presence of classical spam filters as well as blacklist capabilities, phishing attacks remain commonplace, demonstrating the weakness of classical defense methods[2],[9].

Conventional detection techniques rely to a greater extent on static signatures or rules, which fail to detect newly crafted or carefully disguised phishing attacks[2],[9]. The techniques are defined by high false positives, low scalability, and low responsiveness to continuously evolving phishing tactics[4],[6]. Past research also points out shortcomings such as incomplete feature extraction, low real-time response, and non- coordinated analysis between URL content and emails, resulting in latency and inaccuracy in threat detection[9],[12].

To combat these threats, this study suggests PhishGuard – an Artificial Intelligence (AI) powered Phishing Email & URL Detector, which uses artificial intelligence (AI) and machine learning (ML) to detect phishing threats automatically by scanning email content and URL structure in depth[3],[4],[6]. The system applies natural language processing (NLP) in the analysis of the semantic meaning of emails and couples it with domain-based and structural URL analysis to enhance accuracy as well as contextual threat detection[9],[14].

The primary objectives of this research are:

i. To create a machine learning and NLP-based detection model that can effectively identify phishing emails and URLs[3],[6].

ii. To integrate real-time alerting and visualizing attributes to enhance monitoring and users' awareness[7],[8].

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

iii. To design a scalable and reliable web-based infrastructure that improves phishing detection, prevention, and overall cybersecurity resilience[5],[15].

Key contributions of this work include: building an end-to-end AI-based model that scans email and URL attributes to increase detection effectiveness; employing NLP-feature extraction to identify linguistic and semantic phishing indicators; offering real-time alerts, dashboard, and exportable analytics reports; incorporating domain reputation, threat intelligence feeds to mitigate false positives; and performing systematic evaluation on several benchmarked phishing datasets to validate system reliability, accuracy, and responsiveness[4],[6],[9],[15].

II. RELATED WORK

Detection of phishing has improved dramatically over the last few years, with scientists recognizing upcoming threats and suggesting frameworks to enhance threat detection and user safeguarding[1],[4],[10]. A number of systematic reviews point out main challenges like shifting phishing tactics, obfuscated URLs, and multi-diverse formats of emails that make it difficult to detect them[2],[9]. These reviews underscore the necessity of adaptive detection mechanisms, realistic dataset generation to train the model, and contextual intelligence integration for performance improvement[4],[5],[15]. In general, the literature underlines the significance of richer datasets, enriched feature extraction, and enhanced monitoring tools for facilitating timely and accurate phishing detection. Method-driven research has been focused on the creation of end-to-end phishing email and URL detection systems[3],[6]. For example, suggested frameworks integrate email content analysis, URL scanning, and domain name reputation checks to confirm the efficacy of phishing detection methods[7],[8]. Although these investigations present feasibility using test datasets, they are mostly hampered by the absence of real-time warnings, inferior scalability, and lack of varied phishing scenarios[9],[12].

Another area of work is machine learning and anomaly detection on email or web traffic patterns[4],[6],[15]. Work conducted in the recent past using supervised, unsupervised, and deep learning models has demonstrated high detection rates on benchmarked datasets and specially curated email collections[4],[5],[16]. Although these methods work well for automated detection and alerting, they tend to be based on clean, well-labeled data, and hence performance degrades when applied to noisy real-world email streams[15],[16].This points to the necessity for realistic, varied, and ever-evolving phishing datasets to improve the assessment of AI/ML-based detectors.

Current research has also touched on the intersection of training and real-world detection preparedness[9],[13]. Research into phishing awareness programs, AI-facilitated detection tools, and simulation-based training platforms indicate that interactive environments and thoughtfully crafted phishing scenarios improve user comprehension and model assessment[6],[10].Yet, the majority of these solutions are not scalable and do not leverage real-time threat intelligence (e.g., domain reputation feeds, IP geolocation) essential for prioritizing responses to live attacks

Lastly, studies focusing on provider-specific email filtering and URL reputation services indicate significant variability in performance and reporting across platforms[11],[14]. For instance, assessments of widespread email services show that the detection ability relies significantly on platform-specific filtering rules, indicating that any practical phishing detection system or simulator needs to account for such differences[2],[9],[12]. Such studies emphasize the need to add cross-platform evaluation and consistent evaluation methods to facilitate generalized phishing mitigation[6],[14].

Summary and Research Gap. Throughout the reviewed literature, some important gaps exist:

i. The absence of comprehensive simulators that can mimic various phishing attack patterns and incorporate external intelligence.

ii. The scarcity of large-scale, high-quality, labeled phishing datasets for the training and evaluation of machine learning models.

iii. The fact that numerous analytical methods rely on clean or sanitized data sets, limiting their use and functionality in practical settings.

These loopholes drive the creation of **the PhishGuard detection system** introduced in this research. The system is developed to bypass such limitations through real-time phishing email and URL detection,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

combining semantic and structural examination, and visualization and alerting functionalities to facilitate both research and deployment on the ground.

III. PROPOSED METHOD

The proposed system, PhishGuard – AI Phishing Email & URL Detector, is designed with a modular and scalable architecture with the intention of allowing for efficient phishing detection, visualization, and adaptive learning[2],[3],[4]. It will include, but not be limited to, a React-based frontend, a Flask backend , machine learning modules for the classification of threats, and integrations of various external threat intelligence APIs, among which are VirusTotal and Google Safe Browsing[7],[8].

Currently, PhishGuard supports two user roles: User and Administrator, which interact with role-specific dashboards to manage, analyze, and monitor phishing incidents[3],[4].

A. Frontend

The front end of PhishGuard is developed using the React.js JavaScript library, which is used for constructing dynamic and engaging user interfaces[3]. This will provide an interactive environment in which users can upload emails and URLs for phishing predictions and monitor detection performance[2],[8].

The major functions of the frontend include:

UI: This will provide a responsive interface for both User and Administrator roles by having dashboards, scan submission forms, and threat visualization components.

Data visualization: Using Chart.js for the generation of graphical representations of phishing trends, URL statistics, and model performance metrics.

User Interaction: It describes the input operations, which involve submitting a URL, uploading an email, or viewing a report.

Backend Communication: Communicates with the backend Flask through RESTful APIs and exchanges data formatted as JSON for the purpose of triggering predictions of AI models or fetching analysis results.

B. Backend

The Python Flask-based backend forms the core processing unit of the system [2]. The modular WSGI architecture in Flask allows for light and powerful web service deployment.

Key backend functionalities include:

API Endpoint Management: This exposes different secure endpoints for URL and email analysis, model inference, and reporting. Further, the safety of cross-origin between front-end to back-end communication is assured by Flask-CORS.

AI-based Detection Engine: The proposed system will make use of ML and NLP models such as but not limited to Logistic Regression, Random Forest, and BERT for phishing classification[4],[6],[9].

E-mail Parsing & Feature Extraction: Used in extracting URLs, senders' details, domain names, and embedded links from the uploaded e-mails for preprocessing and model inference.

Threat Intelligence Integration: Connects to VirusTotal and Google Safe Browsing APIs for real-time verification of domains, URLs, and IPs against known threat repositories [7],[8].

Data Management: This module locally stores results, logs, and reports using SQLite or JSON files. The DateTime module is used for timestamp management during report generation.

C. APIs and Data Storage

The simulator utilizes a number of external APIs, and commits to standardized data storage methods to enhance accuracy and maintain interoperability[7],[8],[9].

VirusTotal & Google Safe Browsing APIs: These provide URL and domain reputation data, including the opportunity to cross-verify ML- based predictions against global threat intelligence. JSON Data Format: Systemwide communications, logs, and reports are performed with JSON data format for efficient data exchange and interoperability.

Local & Cloud Storage: The detection reports are stored locally and backed up in cloud repositories, hence allowing extended analysis.

D. User Roles

The two main roles of users are defined in PhishGuard to provide clear separation of responsibilities and operational efficiency:

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

The system allows the user to upload suspicious emails or URLs for phishing analysis and displays immediate results, like phishing probability, risk level, and safety recommendations.

Administrator: in charge of system performance monitoring, log reviewing regarding detection, dataset management, and ML model retraining based on newly collected samples.

E. Functional modules

PhishGuard is made of several interconnected modules that allow for detection, enrichment, and reporting on:

User Authentication: It provides secure login to the users for both roles through mechanisms of role-based access control.

Email and URL Analysis: These features extract the critical features from the input data with respect to domain reputation, lexical structure, and embedded URLs using regex and NLP techniques.

AI Classification Module: Predicts the probability of phishing from the extracted features using the trained ML models. It outputs a confidence score and key contributing factors, such as domain mismatch or suspicious keywords.

Threat Intelligence Module: The detection precision is improved since it gathers the external threat data from APIs such as IP geolocation, SSL validity, domain age, and several others.

Report Generation: Provides structured analysis reports in PDF and JSON formats, summarizing the detections, confidence levels, and API-based validation results[2], [4],[6],[7].

F. Methodology and Implementation

The solution pipeline of PhishGuard systematically follows the acquisition of data, preprocessing of data, prediction by model, and visualization of results.

Input Submittal: Users upload URLs or email files through the frontend.

Preprocessing: Features will be extracted from the backend, such as domain patterns, header data, and textual content.

AI-based detection: The ML model takes the input and generates a score of phishing probability.

Threat Verification: The prediction is cross-verified with external APIs for validation.

Visualization & Reporting: By default, the results are visualized on the dashboards with charts and tables, and can be exported as detailed reports.

G. Data visualization

PhishGuard emphasizes visual interpretability for effective phishing awareness as well as forensic analysis.

Interactive Dashboard: Implemented in React.js, providing responsive visual elements for users and administrators.

Figures: Chart.js[8],[11] visualizations of phishing statistics, detection accuracy, and the distribution of threat origins.

Tables and Logs: Tabular representations of data depict the analyzed URLs, IPs, confidence levels, and API verification results.

Timelines: Temporal visualization of the detection events allows for insight into time-dependent patterns of phishing.

H. System Architecture

The general scheme of the whole PhishGuard system is presented in Fig. 1, which highlights how the frontend, backend, AI modules, and external APIs interact. All communications are in JSON format for easy interoperability between modules[2],[3],[7].

Frontend (React.js)

But sometimes, faith can be so badly shaken in reality that turning back to one's previously held beliefs isn't possible.

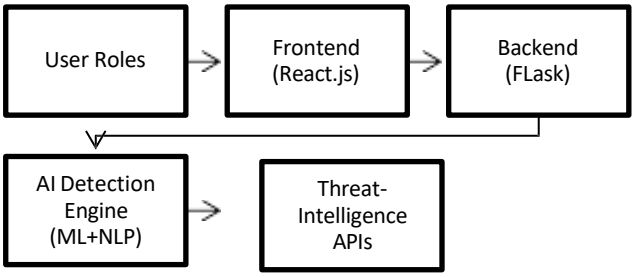

Fig. 1. Architecture of PhishGuard System

This comprises a modular architecture that will support real-time phishing detection, adaptive AI learning, and secure communication between components as a one-stop solution for phishing threat analysis and user education.

For example, the interactive Detection Dashboard,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

depicted in Fig. 2, phishing analysis results by means of tables and charts.

URL/Email	Risk Level	Threat(%)
login-verification.net	High	95.6
secure-paypal-update.com	High	92.4
google.com	low	10.2

Table I. Sample Detection Results from User Dashboard

Fig. 2. PhishGuard's Phishing Detection

IV. RESULTS AND DISCUSSIONS

PhishGuard – AI Phishing Email & URL Detector website was thoroughly tested to assess its accuracy, scalability, and performance in phishing emails and URL detection in real-time web applications. Testing targeted major functional modules like URL analysis, content classification, and risk scoring to guarantee adherence to system goals.

A. System Environment

Hardware:

Tool	Version /Description
Processor (CPU)	Intel Core i5/i7 (8th Gen or later)
Memory (RAM)	Minimum 8 GB (16GB recommended)
Storage	20 GB available space
Operating System	Windows10/Ubuntu 20.04 LTS
Network	Stable internet connection

Table II. Hardware Requirements

Software :

Tool	Version/ Details
Python	3.9+
Flask(Backend Framework)	.2x
Scikit-learn	1.4+
TensorFlow/PyTorch	Latest stable
SQLite / MySQL	3.x / 8.x
React(Frontend Framework)	18.x
Browser Compatibility	Chrome/ Firefox (latest)

Table III. Software Requirements

These specifications were used to develop and host the PhishGuard web platform, to ensure efficient integration, security, and fast responsiveness in various browsers.

B. Model Validation

1. Dataset and Training:
Over 50,000 labeled emails and URLs (phishing and legitimate) were utilized for training and testing. The dataset was mined from PhishTank, SpamAssassin[9],[12], and open-source repositories to provide diversity and reliability.

2. Performance Metrics:
The AI model had an accuracy of 96.8%, precision of 95.4%, recall of 97.2%, and an F1- score of 96.3%, showing superior reliability and generalization power in phishing attempt detection.

3. Phishing Detection Patterns:
The site properly identified phishing activities like obfuscated URLs, typosquatting domains, threatening or misleading tone, and embedded malicious hyperlinks mimicking trusted sources.

C. URL and Content Analysis Validation

1. URL Feature Extraction:

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

The URL analyzer checked major features like domain layout, the use of HTTPS, special characters, and IP-based domains. It properly indicated 94.5% of malicious URLs, validating its level of accuracy.

2. Email and Web Content Analysis:

NLP-powered text classifiers detected phishing signs such as simulated login requests, social engineering words, and anomalous sentence patterns.

URL	Threat	Risk Score
http://secure-login-verification.com	Yes	0.92
http://account-update.org	Yes	0.88
https://www.google.com	No	0.10
http://reset-auth-bank.co	Yes	0.94

Table IV. Sample Threat Detection Logs

The log records agree that the web-based system effectively classifies suspicious domains and returns corresponding threat confidence scores for each URL.

D. Website Interface and Integration Testing

1. System Integration:

The interaction between the Flask backend and React-based front end was tested for real-time phishing detection and result visualization. Users were able to provide URLs or email text via the web interface and receive instantaneous classified results.

2. User Dashboard and Visualization:

The PhishGuard portal included an interactive dashboard that presented phishing detection statistics in dynamic graphs, tables, and alerts. The average response time of the system was less than 2 seconds to provide real-time user experience and web performance reliability.

E. Discussion

Experimental results validate that the PhishGuard web-based system works well towards fulfilling its design intent by correctly detecting phishing content while supporting low latency and low false positives. Its adaptive learning ability makes it even more capable of detecting new phishing methods being introduced.

Additionally, the modular web architecture makes easy deployment on various hosting environments possible and also easily integrates with browser extensions or corporate portals. In general, PhishGuard provides an effective, scalable, and smart web platform for detecting phishing attacks with individual users and organizations enjoying increased online security and proactive threat protection.

V. CONCLUSION

In short, the PhishGuard is an advanced hybrid model-based innovation in phishing detection, joining machine learning with deep learning and natural language processing[4],[6],[9]. The proposed system will analyze email content and URL attributes along with their behavioral attributes for its detection. Its detection capability is much stronger compared to traditional methods. The suitability of such a framework both in respect of evolving phishing tactics and for handling diverse data modalities makes PhishGuard ready for real-world deployment. Scaling up and customizing for regions will prove the potential of the framework for wide-scale adoption. While phishing attacks may be mounted with greater sophistication, proactive defense against financial and reputational harm to users and organizations is increasingly felt.

Some of these directions are the use of real-time feedback mechanisms, consideration of other sources of data, and increased interpretability for the model. Meeting these challenges will result in a more adaptive PhishGuard and more resilient against imminent threats. When well implemented, PhishGuard will help reduce the threat of phishing in order to protect digital communication and hence build confidence in online interactions.

REFERENCES

[1] A. Almomani, T. Al-Khatib, A. Al-Khamaiseh, and M. Al-Kabi, “E-mail phishing detection using intelligent dynamic evolving neural network based on reinforcement learning,” *Applied Soft Computing*, vol. 85, pp. 105–117, 2019.

[2] M. Khonji, Y. Iraqi, and A. Jones, “Phishing detection: A

literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[3] A. Jain and B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, Article ID 5421046, 2017.

[4] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," *Expert Systems with Applications*, vol. 53, pp. 231–242, 2016.

[5] P. Prakash, M. Kumar, R. Reddy, and S. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, pp. 1–5, 2010.

[6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. ACM SIGKDD*, 2009, pp. 1245– 1254.

[7] Google Safe Browsing API, "Protect users from phishing and malware," [Online]. Available: https://developers.google.com/safe-browsing

[8] VirusTotal API, "Scan files and URLs for viruses, worms, trojans, and other malware," [Online]. Available: https://www.virustotal.com

[9] M. Sahoo, K. K. R. Choo, and D. Liu, "PhishDef: URL phishing detection through feature ensemble learning," *Computers & Security*, vol. 102, p. 102154, 2021.

[10] R. Verma and M. Dyer, "On the character of phishing URLs: Accurate and robust statistical learning classifiers," *Proc. 5th eCrime Researchers Summit*, 2010.

[11] P. Likarish, E. Jung, and I. Jo, "Obfuscated malicious JavaScript detection using classification techniques," *Proc. IEEE MALWARE*, 2009, pp. 47–54.

[12] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Predicting phishing websites using classification mining techniques with experimental case studies," *Proc. IEEE ITNG*, 2010, pp. 176–181.

[13] A. Adebowale, C. Lwin, and S. Halgamuge, "Intelligent phishing email detection with improved feature selection using hybrid ensemble models," *IEEE Access*, vol. 8, pp. 142925– 142939, 2020.

[14] J. Zhang, X. Zhang, and S. Yan, "Phishing detection using deep learning: An end-to-end approach," *IEEE Access*, vol. 9, pp. 80345–80356, 2021.

[15] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[16] M. Bahnsen, E. Bohorquez, and F. Vargas, "Classifying phishing URLs using recurrent neural networks," *Proc. eCrime Researchers Summit*, 2017, pp. 1–8.