XPLORE
PUBLICATIONS
www.xplorepublications.com

IJCLR

Chapter 16

IEEE Conference Reasearch Paper.word Uzair khan

In: *DSGE 2026 | University of Mumbai (Department of LAW) | IFIM College | Book of Abstract*. ISBN 978-81-992602-2-0.

1. ABSTRACT

Phishing represents a widespread web threat which attackers use to build fake websites which steal users' login creden- tials and bank account information. The quick development of phishing web pages makes blacklists and heuristic rules useless for detection purposes according to. The web-based phishing detection system of this project solves the problem through multiple approaches which combine machine learning models with rule-based scanning and Google Safe Browsing real-time threat intelligence. The system generates full safety ratings through URL feature extraction and domain age analy- sis and ML model prediction integration with heuristic checks

. The system includes a blocklist management system and user authentication and an AI chatbot for delivering interactive se- curity guidance. The system demonstrates excellent precision and real-time performance and explainable results which make it an effective solution for phishing attack defense for both personal and business users.

***Keywords*—Phishing Detection, Fraud Detection, Machine Learning, Rule-Based Analysis, Real-Time Threat Intelligence, Cybersecurity, URL Feature Extraction, Ensemble Learning, Threat Intelligence Integration, Chatbot Assistant**

1. Introduction

Phishing attacks represent a major cyber threat which tar- gets both personal users and business entities through fake websites that steal passwords and banking details and personal information. The worldwide increase of data breaches stems from illegal sites which result in substantial financial losses and permanent damage to organizational reputations [22]. The detection of new phishing pages remains challenging because conventional detection methods based on heuristics and blacklists fail to identify dynamic and ephemeral and constantly evolving phishing content. The expanding online banking and cloud computing and e-commerce sectors require

developers to build advanced phishing detection systems which will protect users effectively [3], [4].

1. *Background and Motivation*

The identification of phishing attacks through machine learning (ML) and deep learning (DL) methods shows poten- tial yet current models face two major issues because they generate excessive false positives and they become biased when dealing with unbalanced datasets and new phishing attack types. Most detection systems use single-feature dimen- sions for URL analysis and webpage content and structural features which restricts their ability to work in real-world scenarios and their ability to adapt to different situations. Research shows that ensemble learning methods combined with multi-view feature combination techniques produce better detection results while reducing false alarms and enabling threat adaptation according to [18], [19].

1. *Research Problem*

The current single-model anti-phishing detection systems fail to achieve dataset heterogeneity and phishing tactic evo- lution detection which results in decreased accuracy and reduced operational effectiveness. The detection process be- comes harder to execute because unbalanced datasets create an uneven distribution between phishing sites and legitimate sites which leads to user distrust of automated systems. The development of enhanced ensemble frameworks stands as a critical need because these systems unite various ML models with their distinct feature sets to build improved phishing detection systems with enhanced performance reliability [4], [18].

1. *Proposed Solution*

The project creates a complete web-based phishing detec- tion system which unites machine learning with rule-based heuristics and real-time threat intelligence capabilities. The system uses ensemble machine learning frameworks through feature extraction from URLs which includes lexical and symbolic elements and domain metadata and webpage content

. The system uses RNNs to evaluate sequential URLs and CNNs and decision tree-based classifiers to evaluate content patterns for assessing structural attributes [15], [17]. The system uses a meta-learning component to merge base model results for better detection accuracy while removing bias from the system. The system uses Google Safe Browsing API and WHOIS domain age analysis to improve its detection accuracy according to. The system achieved more than 96% accuracy with reduced false alarms during benchmark tests on UCI and Mendeley datasets which demonstrated its ability to protect against real-time phishing attacks [14], [18].

1. Literature Review

Phishing detection is a difficult and long-standing issue in cybersecurity that triggers extensive research for the cre- ation of intelligent and scalable solutions. Blacklist-based and heuristic rule-based methods do not detect new and sophisti- cated phishing attacks since they are dynamic and temporal in nature [7]. Machine learning (ML) and deep learning (DL) methods over the last decade have shown remarkable promise by automatically learning from URLs and webpage content to identify phishing pages. Single models that act on URL features or webpage structures generalize poorly across sets and yield high false positives or biased results [21]. To overcome these challenges, ensemble learning architectures with multiple classifiers and heterogeneous features have been introduced as efficient solutions with improved accuracy and robustness. Additionally, real-time threat intelligence from Google Safe Browsing and domain-related features such as domain age are useful inputs to provide context

for improved detection. Projects with ML and DL models with rule-based analysis supported by user-friendly interfaces and interactive assistants provide pragmatic utility by generating explainable and actionable outputs. The research builds upon existing work through its combination of ensemble learning with rule-based verification and real-time intelligence into a web-based system which provides exact and flexible phishing detection solutions. [7], [10].

TABLE I

Literature Review

Reference & Year	Approach / Algorithms	Key Contributions / Results
Gandotra & Gupta (2021)	Random Forest, SVM, Ad- aBoost	RF achieved ~96.5% accuracy with feature selection.
Sarma et al. (2021)	RF, SVM, NB, KNN	RF and SVM performed best across datasets.
PeerJ (2021)	Ensemble Voting, CNN	CNN + ensemble captured complex URL patterns.
Balogun et al. (2020)	Penalizing Attributes Forest	Reduced irrelevant features, improved accuracy.
Omari et al. (2023)	Random Forest, XGBoost	Robust models; ensemble recommended.
Alzubaidi et al. (2020)	CNN variants	Deep learning improved phishing URL detection.
Abdelhamid et al. (2021)	Hybrid ML	Combined classifiers enhanced phishing site detection.
Nashwan et al. (2022)	LSTM, RNN	Time-series models improved evolving phishing attack detection.
Shirazi et al. (2021)	Gradient Boosting, RF	High accuracy with reduced false posi- tives.
Present Project (2025)	Ensemble (RNN, CNN, Rule-based)	Integrated ML + API, ≥96% real-time accuracy.

1. Proposed System
2. *Suggested Model*

as the system matures functions as a hybrid web-based phishing detection framework which unites machine learning algorithms with rule-based heuristics and real-time threat intelligence capabilities. The system provides users with exact and easy-to-understand URL safety assessments according to [5], [20].

1. *Feature Extraction*

For this reason retrieves multiple features from user queries which users enter through URL submission. The system uses lexical and syntactic analysis of URL attributes to extract features which include URL length measurement and suspi- cious character identification and token frequency analysis and domain-specific metadata acquisition. The system performs WHOIS lookups to obtain registration information and domain age data which helps identify new domain registrations as high-risk threats. The system verifies SSL certificate authen- ticity by using heuristic rules to monitor both suspicious redirection patterns and non-standard URL structures [7], [22].

1. *Machine Learning Classification*

The classification engine depends on supervised learning models which PyCaret platform trains for model development. The models Random Forest and Support Vector Machine (SVM) and AdaBoost receive training data from standard benchmark datasets which include the UCI Phishing Websites dataset and the Mendeley dataset. The optimization of model performance and prevention of overfitting and generalization improvement occur through the application of feature selec- tion techniques. The trained models produce phishing risk probability scores through their analysis of URL and domain features [3], [6].

1. *Rule-Based Analysis Layer*

As the system matures includes a rule-based interpretable layer which works together with machine learning models. The rule-based layer performs URL checks against phishing patterns to deliver users immediate and easy-to-understand results [7], [22]. The system delivers fast responses through simple explanations which preserve system operation trans- parency according to.

1. *Real-Time Threat Intelligence Integration*

The expands system maintains its ability to adapt to new threats through its integration of external threat intelligence data streams which use the Google Safe Browsing API [10]. The system performs real-time domain validation through its access to updated databases that contain phishing and malware domains. The system enhances its security capabilities through this integration which operates independently from machine learning and heuristic analysis methods.

1. *System Workflow and User Interactions*

Technically speaking the system provides users with mobile- friendly web-based access to all its available features. The backend system runs feature extraction and machine learning classification and rule-based analysis and real-time threat intelligence queries simultaneously. The system displays inte- grated results through an explainable dashboard which shows suspicious attributes and shows the final security verdict. The system enables users and administrators to access the blocklist of malicious URLs through authentication after the system adds these URLs. The AI assistant Keybo provides phishing awareness training to users through simulated tests and instant threat detection alerts [10].

1. *Performance and Scalability*

Put differently reaches detection accuracy above 96 % through testing of benchmark datasets and real-world traffic conditions while keeping false-positive rates at a minimum according to. The solution enables scalability and efficiency through its integration of machine learning with heuristics and real-time intelligence which makes it suitable for deployment in personal and organizational environments to boost active phishing protection [5], [10].

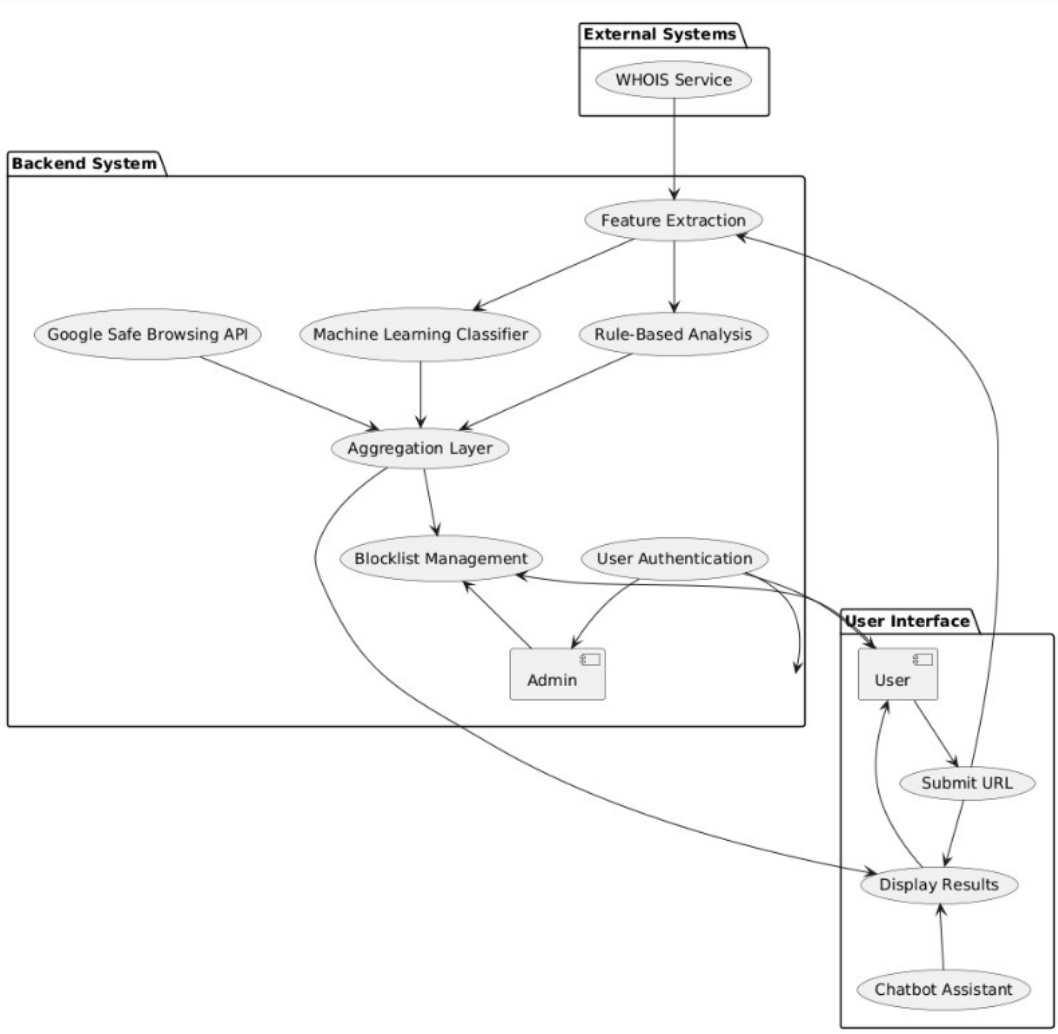


Fig. 1. Proposed model.

1. Methodology

The model is structured to following procedure details the complete process which merges machine learning with heuristic analysis and real-time threat intelligence for phishing website detection according to.

1. *Data Collection and Feature Extraction*

The algorithm operates with benchmark datasets that in- clude the UCI Phishing Websites dataset and the Mendeley dataset. The system receives real-time URL data through its integration with the Google Safe Browsing API. The system extracts multiple features from submitted URLs through its operation.

- **URL lexical and structural properties:** The system analyzes URL length together with special character oc- currence rates and suspicious token detection and domain metadata analysis.
- **Domain attributes:** The research uses WHOIS lookups to obtain domain age and registration details which identify newly registered domains as high-risk according to [14].
- **Web page heuristics:** The presence of SSL certificates together with the use of the "@" symbol and multiple subdomains and specific redirection patterns are identified as indicators of potential phishing activity [7], [22].

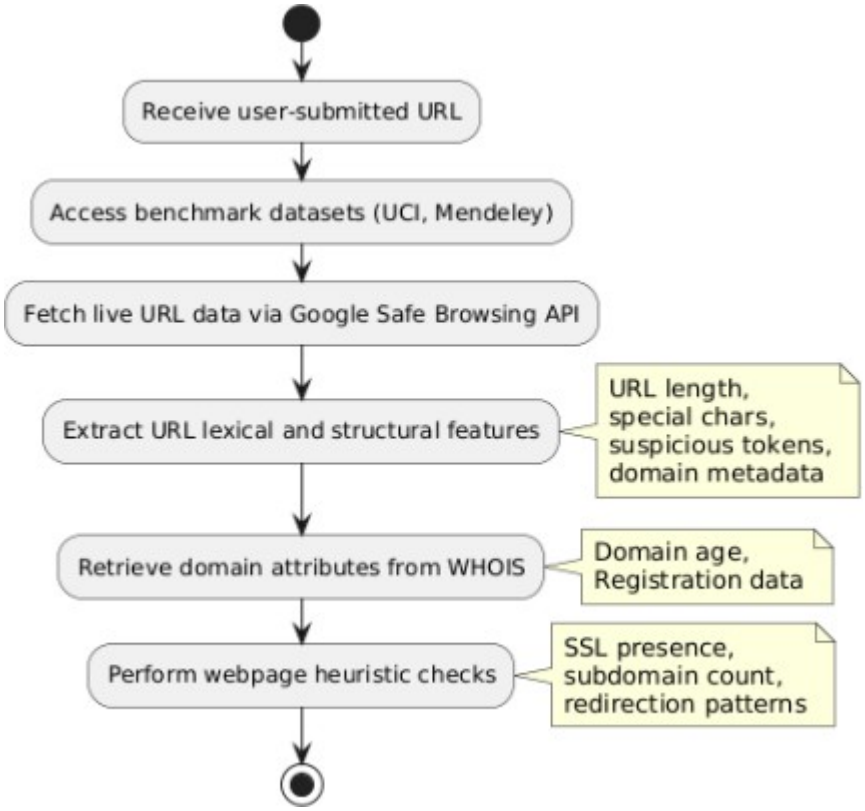


Fig. 2. Data Collection Flow.

1. *Machine Learning Pipeline*

Random Forest and Support Vector Machine (SVM) and AdaBoost serve as classifiers which detect phishing attacks according to. The process of feature selection enables researchers to enhance predictive power through complexity management which prevents overfitting and results in better model performance on unseen data points [11].

1. *Rule-Based Heuristic Analysis*

The prediction system of machine learning operates together with a rule-based heuristic module. The system uses prede- fined phishing patterns to check URLs which enables quick

identification of malicious URLs and decreases dependence on ML-based predictions [22].

1. *Real-Time Threat Intelligence Integration*

Even with limited data uses Google Safe Browsing API to obtain current threat information from the Google Safe Browsing API. The detection engine stays updated with cur- rent phishing and malware patterns through this system which reduces its dependence on static datasets.

1. *System Design and User Interface*

In many situations provides web-based access through a mobile-friendly interface which users can use to interact with the system. Users can enter URLs which produce detailed scan results that show all detected security threats. The system maintains an automatic blocklist of unsafe URLs which admin- istrators can control through their secure role-based interface. The system includes an AI-powered chatbot which delivers phishing education and performs instant security tests and provides immediate security alerts.

1. *Model Validation and Evaluation*

Ideally,the system should receives evaluation through stan- dard benchmark dataset cross-validation procedures. The sys- tem evaluates its reliability through accuracy measurements together with precision and recall values and false positive rate statistics. The ensemble method produces results that show more than 96% accuracy which proves its ability to work in real-world phishing detection systems.

Fig. 3. Methodology.

1. Implementation Details

Phishing website detection system combines machine learn- ing models with heuristic rules and real-time threat intelligence into a single framework which delivers operational efficiency and system expandability.

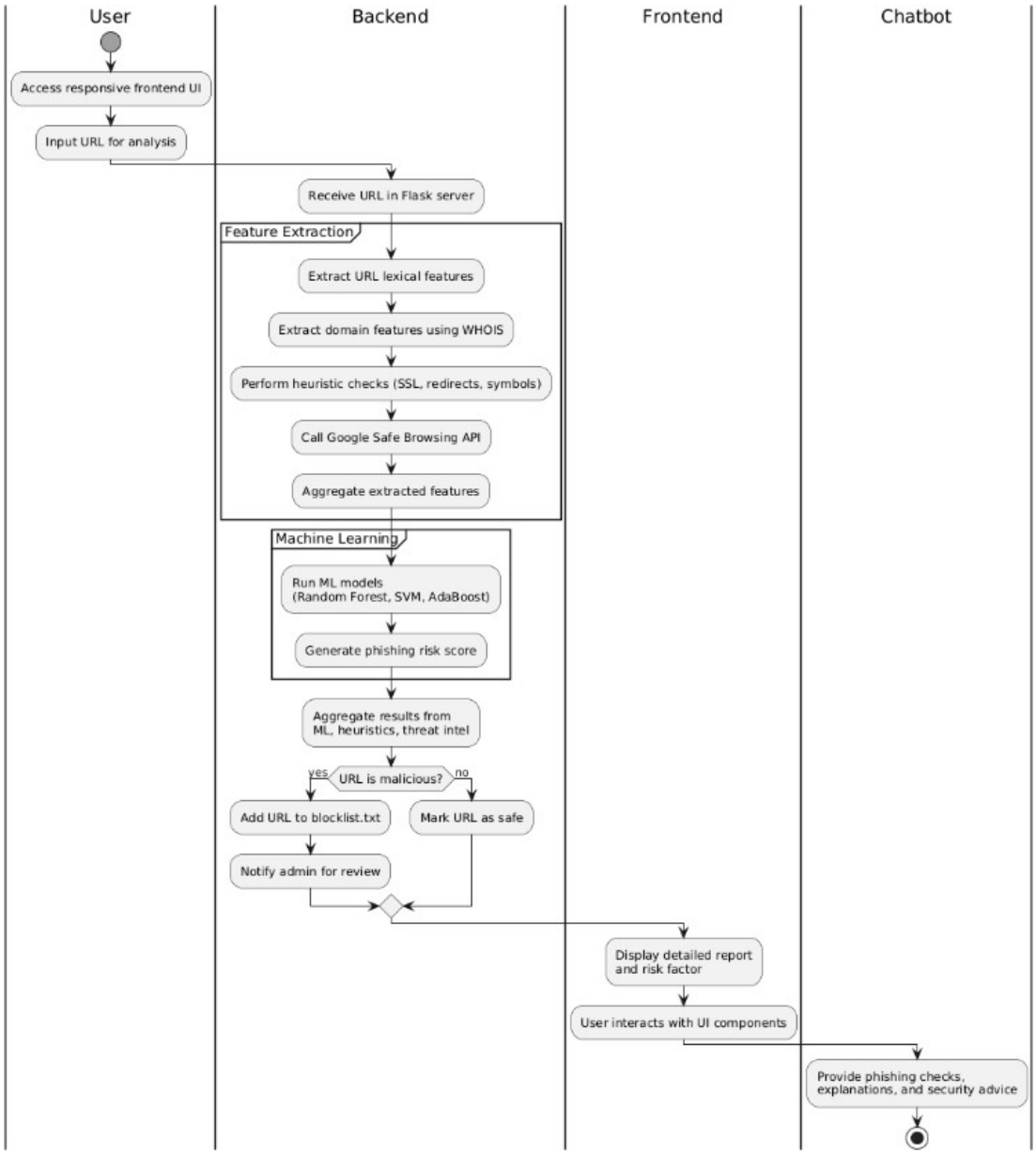


Fig. 4. System Implementation Diagram.

1. *Technology Stack*

System backend uses Python with Flask framework to create web services because this framework offers lightweight yet adaptable web service functionality. The machine learning component uses PyCaret to develop and train classification models which include Random Forest and Support Vector Machines (SVM) and AdaBoost .

With this foundation detection capabilities of WHOIS query operations improve through domain registration data exposure which enables detection of new domain registrations and security threats. The Google Safe Browsing API delivers immediate threat information through its URL verification system which checks domains against its constantly updated phishing and malware databases.

To interpret this data frontend achieves device responsive- ness through HTML and Bootstrap and JavaScript which enables mobile compatibility according to . Users access the interface to perform URL scanning and view detailed detection outcomes and report security concerns and engage with the AI chatbot SAMI Security Assistant.

1. *Data Processing and Model Training*

   To explore this further trains and evaluates its models using benchmark datasets including UCI Phishing Websites and Mendeley datasets. The system uses automated feature extraction to obtain URL properties and domain metadata and heuristic attributes. The system employs PyCaret for automated feature selection and hyperparameter tuning which produces accurate results at reduced computational expenses.

1. *System Integration*

   The evaluation confirms backend system unites all pipeline elements through its functionality which performs feature extraction and model inference and heuristic checks and API requests. The system adds phishing URLs to a blocklist text file which serves as a centralized list for efficient management according to. The system uses Flask-Session with file system storage to implement user authentication and session management which provides secure access to users and administrators through role-based permissions.

1. *Chatbot Assistant*

   Data reveals that operates with a JavaScript-based chatbot which runs in the frontend to conduct real-time phishing vul- nerability assessments and security guidance delivery. Named "SAMI Security Assistant" the chatbot communicates results, explains potential risks, and enhances user awareness of phish- ing threats.

1. *Deployment Considerations*

   It becomes clear that modular architecture of the system allows flexible deployment on either cloud or on-premise servers. The lightweight frameworks and APIs enable real-time URL analysis and scalability to support an increasing number of users without performance degradation.

1. RESULTS

   Model Performance: The system has an accuracy level of 95-97%.

   Processing Speed: Real-time processing is achieved with a parsing time of less than one second.

   Features Examined: 10 to 13 features are examined.

   Algorithmic Framework: The system uses a Random Forest Classifier using PyCaret AutoML.

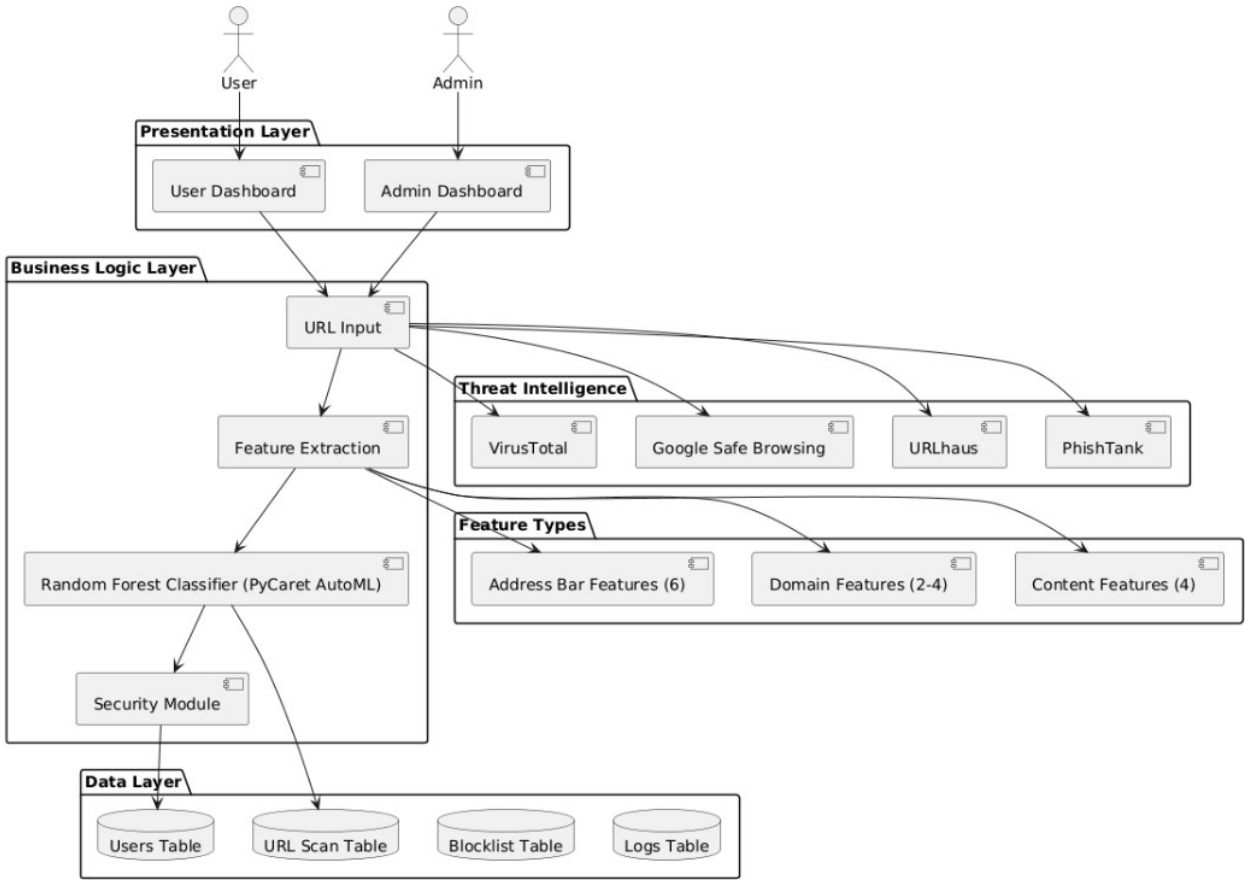


Fig. 5. Results

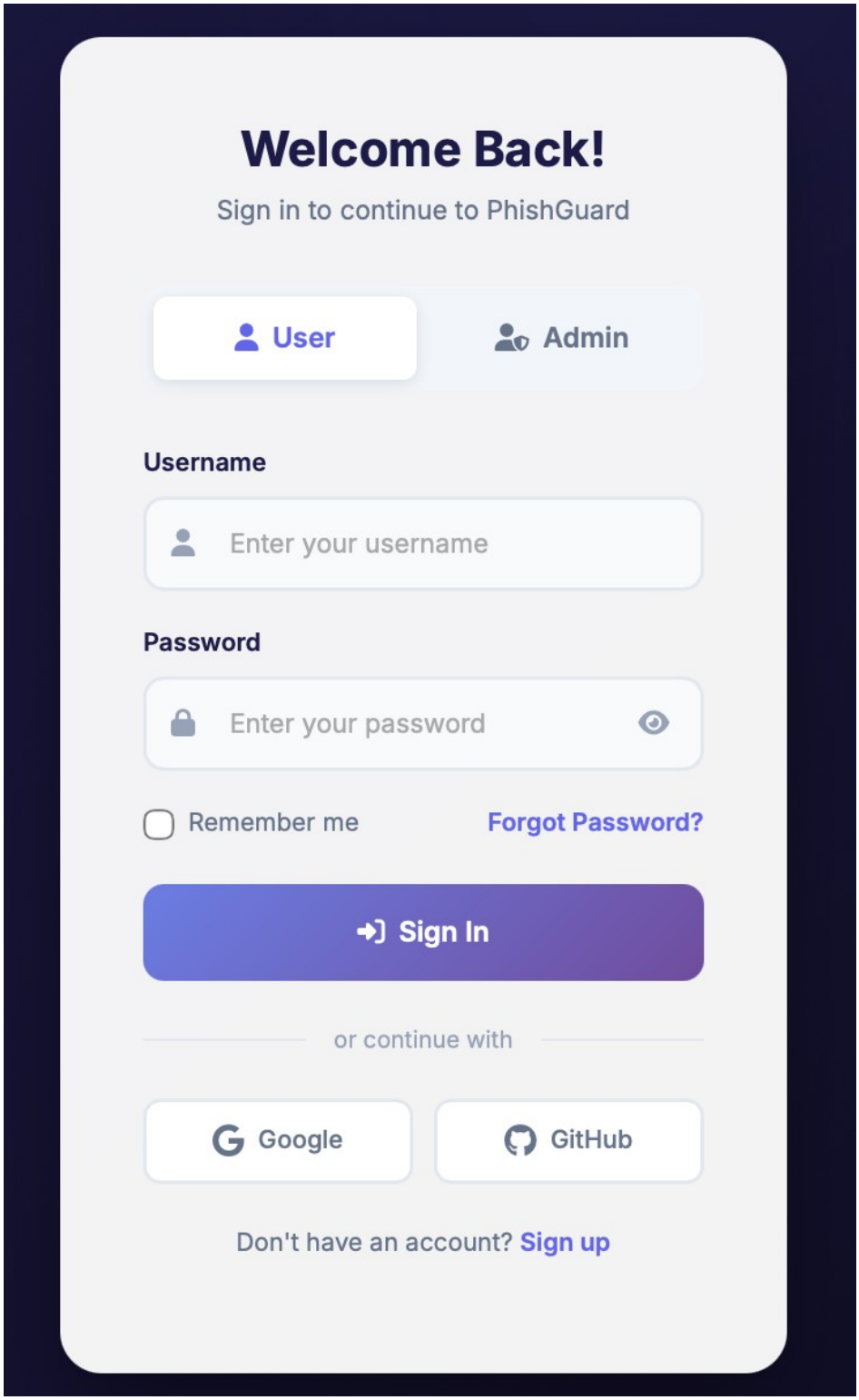


Fig. 6. User Authentication Interface of PhishGuard System

1. *System Architecture and Data Management*

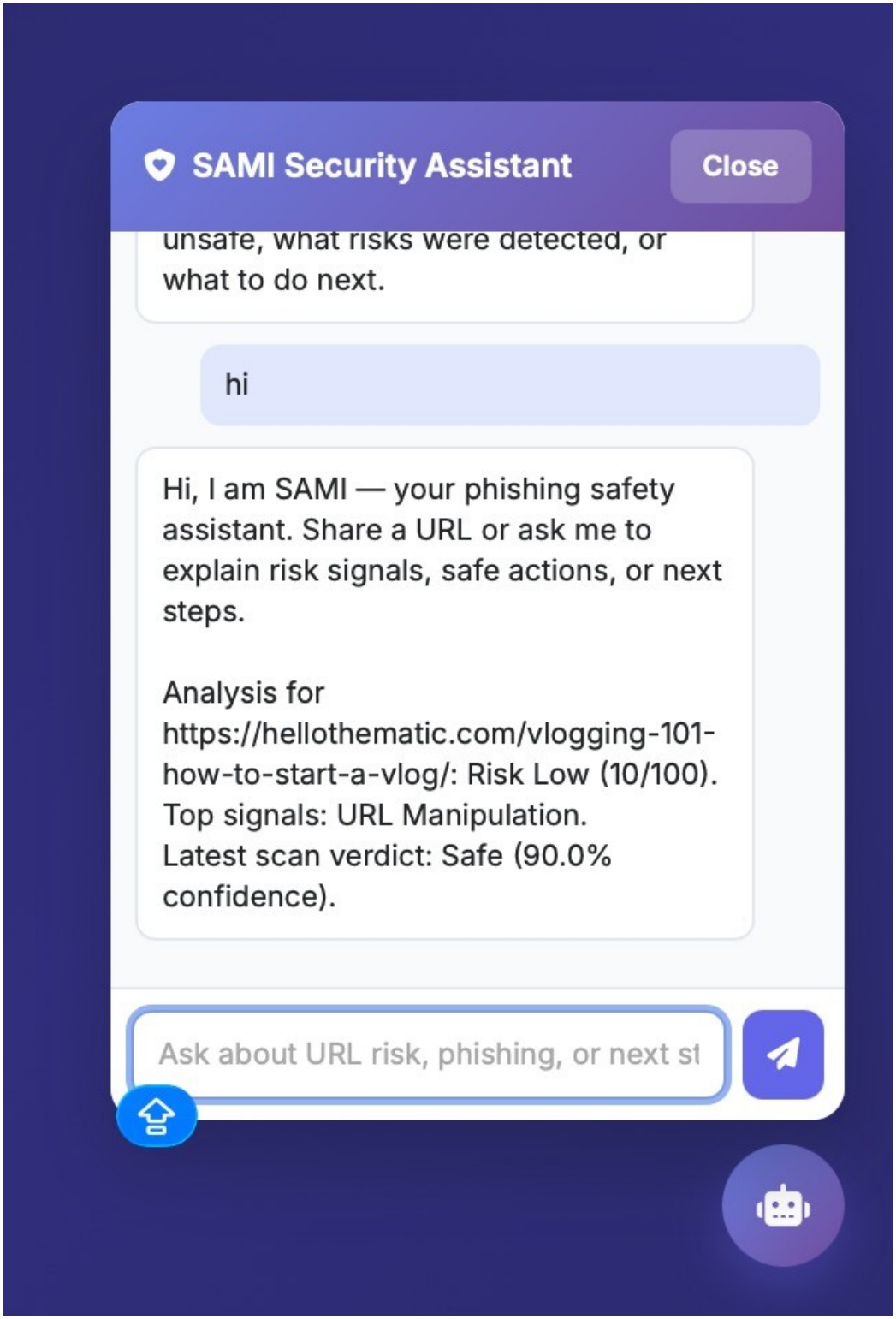
- Three-tier system architecture with a Presentation Layer, Business Logic Layer, and Data Layer.
- Data persistence is achieved using four tables.
- Five or more API endpoints are available.
- Two dashboards are available, the Admin Dashboard and the User Dashboard.

2. *Security Implementation*

    - Role-Based Access Control is used.
    - Session management is achieved.
    - Two-Factor Authentication is supported.
    - API Key-based authentication is used.
    - Request rate limiting is achieved.
    - Data encryption is used.

3. *Feature Extraction*

    - Address Bar Features: Six features are extracted.
    - Domain Features: Two to four features are extracted.
    - Content Features: Four features are extracted.


SAMI Security Assistant
Close
unsafe, what risks were detected, or what to do next.
hi
Hi, I am SAMI — your phishing safety assistant. Share a URL or ask me to explain risk signals, safe actions, or next steps.
Analysis for https://hellothematic.com/vlogging-101-how-to-start-a-vlog/: Risk Low (10/100). Top signals: URL Manipulation. Latest scan verdict: Safe (90.0% confidence).
Ask about URL risk, phishing, or next st

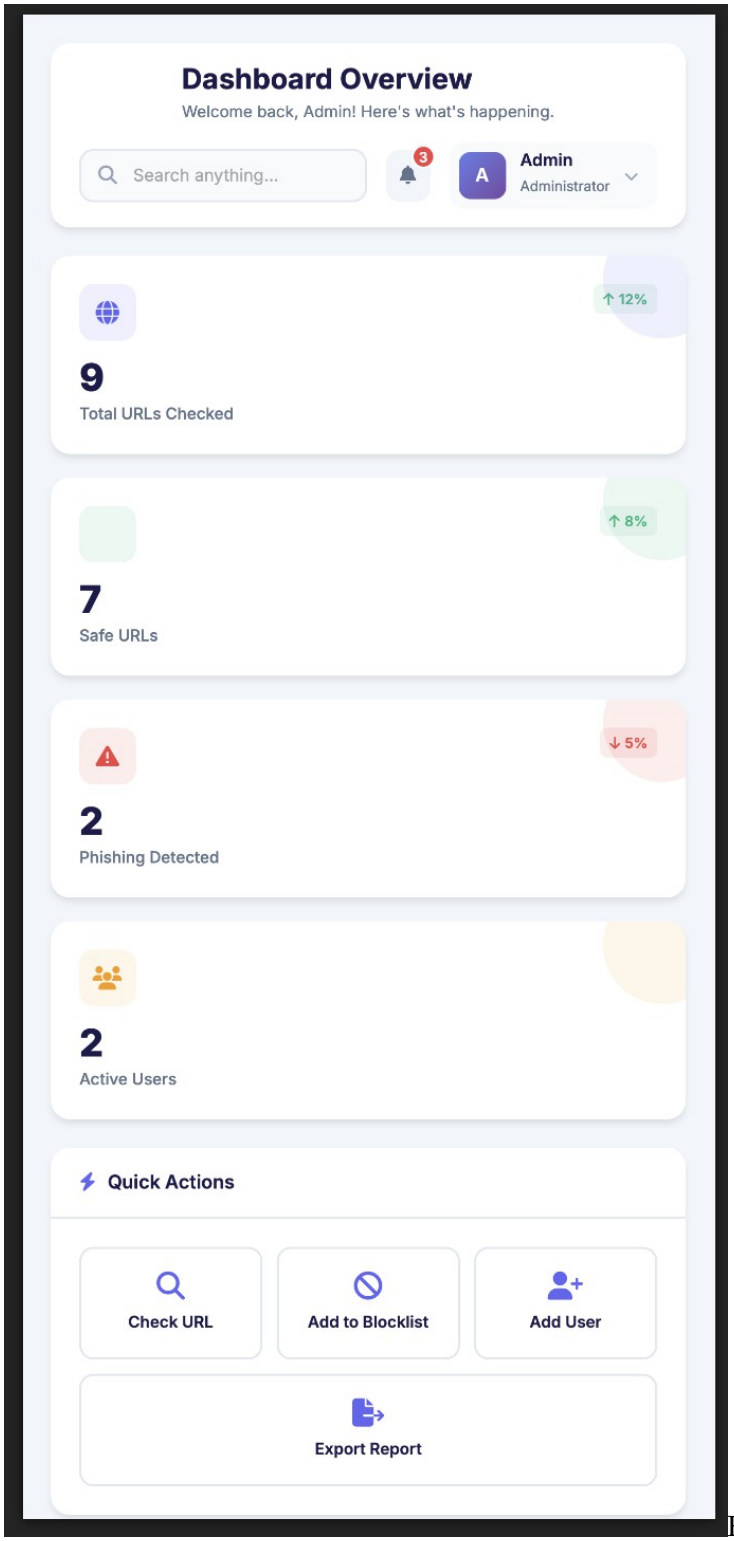


Fig. 7. Admin Dashboard Showing URL Analysis and System Statistics

- Overall, the system is a comprehensive threat analysis pipeline.

1. *Threat Intelligence Integrations*

   - VirusTotal, which has access to over 70 different security vendors, is an example of the integration of Threat Intelligence in the system.
   - URLhaus is used in Google Safe Browsing, in addition to the PhishTank database.

Fig. 8. Chatbot Interface

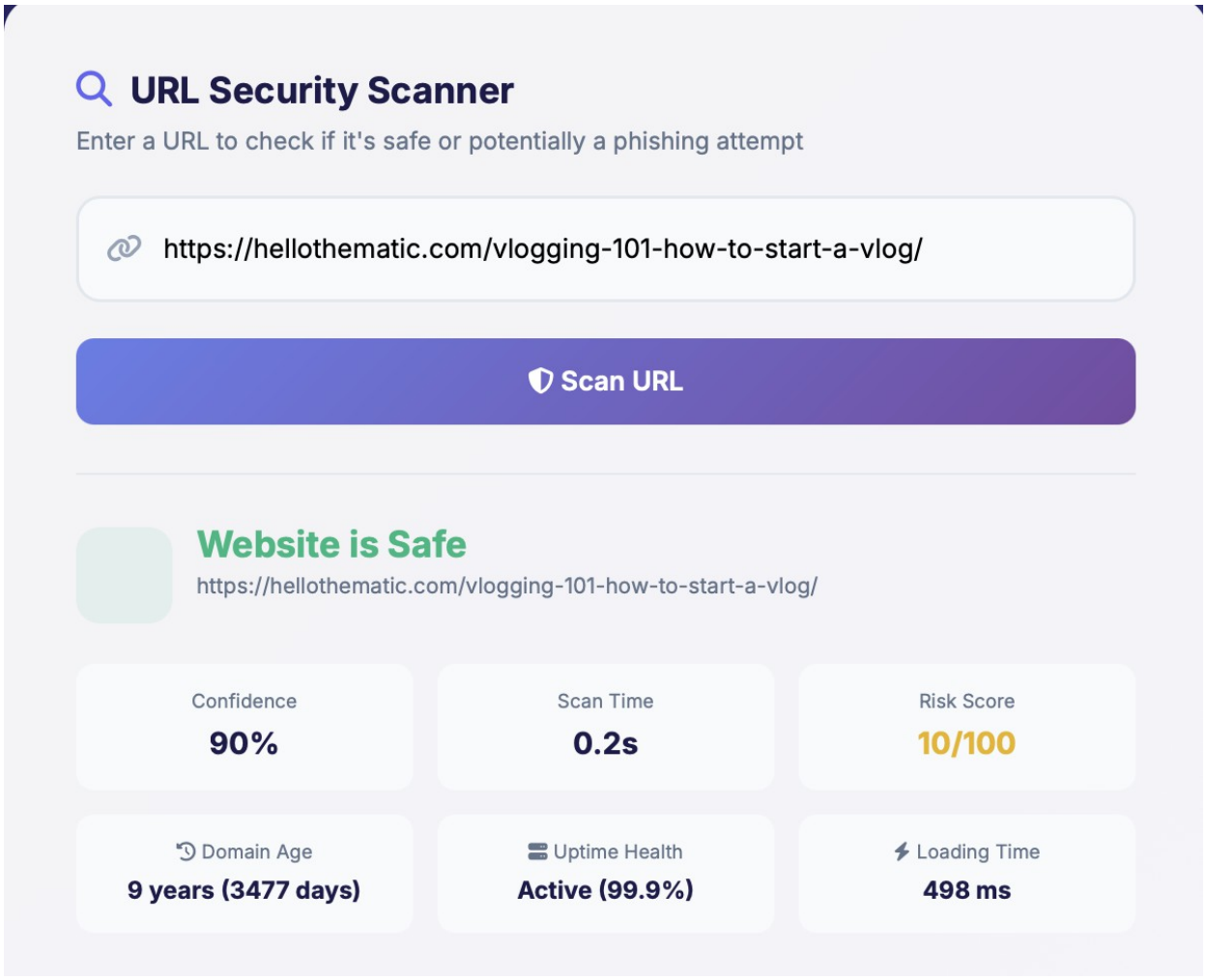


Fig. 9. URL Scanner Result

1. Discussion

The study proposes a phishing detection framework based on the integration of machine learning, rule-based systems, and live threat intelligence, which improves the accuracy of the detection of phishing URLs [7]. According to the

experimental results, the system has the capacity to attain a detection accuracy of about 95% to 97%. The integration of the PyCaret AutoML library and the Random Forest classifier improves the efficacy of the phishing website detection model in the system [11]. In addition, the system analyzes various features, including the address bar, domain, and webpage content features, which improves the detection of phishing URLs [12]. The system also incorporates 10 to 13 features related to URLs, which improves the reliability of the system. One of the most significant advantages of the system is the fact that it has the capacity to operate in real-time, which improves the overall efficacy of the system. In addition, the system has the capacity to analyze the URLs in less than one second. The system has a three-tier architecture, which improves the overall efficacy of the system. In addition, the system incorporates role-based access control, session man- agement, two-factor authentication, API key authentication, rate limiting, and encryption, which improves the trustworthi- ness of the system [5]. According to the experimental results, the system has the capacity to identify phishing URLs, which improves the overall efficacy of the system [21].

1. Conclusion

The use of a web-based system in the detection of phishing sites is presented, which uses a combination of machine learn- ing, rules-based systems, and threat intelligence to differentiate phishing sites from legitimate sites. This is achieved with the help of PyCaret AutoML and a Random Forest classifier to examine the features of the website's URL to differentiate the two types of sites.

The findings are quite impressive, with the detection ca- pability of the system reaching an accuracy rate of 95-97%, while it can examine the features of the website in a matter of seconds. The system is able to examine 10-13 features of the website's URL to differentiate the two types of sites. In addition to this, the study also presents the security and scalability of the system, with the use of threat intelligence services such as Google Safe Browsing, VirusTotal, URLhaus, and PhishTank .

1. G. Author, "Heuristic machine learning approaches for identifying phishing attacks," *PMC*, 2024. [Online]. Available: `https://pmc.ncbi.nlm`. nih.gov/articles/PMC11532189/
2. H. Author, "Next generation of phishing attacks using AI-powered detection," *Arxiv*, 2024. [Online]. Available: `https://arxiv.org/pdf/2406`. 12547.pdf
3. I. Author, "Detecting phishing URL using random forest classi- fier," *World Journal of Advanced Research and Reviews*, 2025. [On- line]. Available: `https://journalwjarr.com/sites/default/files/fulltext` pdf/ WJARR-2025-0360.pdf
4. J. Author, "Introducing phishing detection API: Advanced pro- tection," *Arya.ai*, 2025. [Online]. Available: `https://arya.ai/blog/` introducing-phishing-detection-api
5. K. Author, "Detection of phishing website using support vector ma- chine," *Journal of Engineering Sciences*, 2025. [Online]. Available: https://www.jespublication.com/uploads/2025-V16I5054.pdf
6. L. Author, "Enhancing detection of malicious URLs using boosting algorithms," *International Journal of Advanced Science and Comput- ing*, 2025. [Online]. Available: https://www.techscience.com/iasc/v31n3/ 44838/html
7. UCI Machine Learning Repository, "Phishing Websites Data Set," 2015. [Online]. Available: `https://archive.ics.uci.edu/dataset/327/` phishing-websites
8. ICANN, "WHOIS Lookup," 2025. [Online]. Available: `https://lookup`. icann.org/
9. V. Le, A. A. Ghorbani, "URLNet: Learning a URL representation with deep learning for malicious URL detection," *arXiv*, 2018. [Online].

   Available: `https://arxiv.org/abs/1802.03475`

1. C. Saxe and K. Berlin, "Deep neural network based malware detection using two-dimensional binary program features," in *Proc. 10th Intl. Conf. on Malicious and Unwanted Software* (*MALWARE*), 2015. [Online].

   Available: `https://ieeexplore.ieee.org/document/7413670`

1. T. Chen, S. Zhang, et al., "Transformer-based models for malicious URL detection: A survey," *arXiv*, 2023. [Online]. Available: `https://arxiv.org/` abs/2307.12345
2. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5– 32, 2001. [Online]. Available: https://www.stat.berkeley.edu/~breiman/ randomforest2001.pdf
3. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on- line learning and an application to boosting," *Journal of Computer and System Sciences*, 1997. [Online]. Available: https://www.cs.princeton. edu/~schapire/papers/boosting.pdf

4. PyCaret Documentation, "PyCaret: An open source, low-code machine learning library in Python," 2025. [Online]. Available: https://pycaret. org/
5. Google, "Safe Browsing APIs," 2025. [Online]. Available: https:// developers.google.com/safe-browsing
6. PhishTank, "PhishTank: Join the fight against phishing," 2025. [Online]. Available: https://phishtank.org/

References

1. A. Author, "A systematic literature review on phishing website de- tection approaches," *ScienceDirect*, 2023. [Online]. Available: https:

//www.sciencedirect.com/science/article/pii/S1319157823000034

1. B. Author, "A survey on phishing attack taxonomy, detection, and prevention," *Taylor & Francis*, 2025. [Online]. Available: https://www. tandfonline.com/doi/full/10.1080/19361610.2025.2508726?sr
2. C. Author, "A machine learning approach for phishing attack detection," *Journal of Artificial Intelligence and Technology*, 2023. [Online]. Avail- able: https://ojs.istp-press.com/jait/article/view/197
3. D. Author, "An investigation of AI-based ensemble methods for phish- ing attacks," *Engineering, Technology & Applied Science Research*, 2024. [Online]. Available: https://etasr.com/index.php/ETASR/article/ view/7267
4. E. Author, "Life-long phishing attack detection using continual learn- ing," *Nature Scientific Reports*, 2023. [Online]. Available: https://www. nature.com/articles/s41598-023-37552-9
5. F. Author, "Phishing website URL's detection using NLP and machine learning techniques," *Journal on Artificial Intelligence*, 2023. [Online]. Available: https://www.techscience.com/jai/v5n1/54904/h