International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

# Web-Based Platform for Phishing URL Detection

**Mugammadhu Sate M[1], Ms. Dr. Latha[2]**

*[1] Student, Department of Cyber Forensics and Information Security,*
*Dr. M.G.R Educational and Research Institute,Maduravoyal, Chennai – 600095, Tamil Nadu, India*
*[2] Assistant Professor, Department of Cyber Forensics and Information Security,*
*University of Madras, Chennai – 600095, Tamil Nadu, India*

---

***Abstract -*** *Phishing remains a fast-evolving cyber threat, exploiting user trust and bypassing traditional defenses with growing sophistication. Static blacklists, manual verification, and basic heuristic filters often miss zero-day phishing attacks or cleverly disguised URLs that mimic legitimate domains. To address these gaps, this study introduces an intelligent web-based phishing detection platform combining machine learning and real-time threat intelligence. Developed on the MERN stack (MongoDB, Express.js, React.js, Node.js), the system applies a Random Forest classifier using lexical features, WHOIS data (domain age, registrant anonymity), and behavioral indicators such as redirect chains, SSL certificate mismatches, and HTTPS inconsistencies. A Selenium-driven headless browser performs dynamic content rendering and deep inspection to detect deceptive elements—spoofed login forms, hidden scripts, and phishing-oriented design structures. Secure interactions are enforced via JWT-based authentication and role-based access control, with distinct user and administrator interfaces. In addition to machine learning classification, real-time API integrations with Google Safe Browsing and VirusTotal enable immediate cross-referencing against known threat databases. Performance and scalability are enhanced by caching and automated data cleanup using MongoDB TTL indexes to eliminate redundant processing. Crucially, the detection model is continuously retrained on newly acquired threat data and user-submitted URLs, ensuring adaptability to emerging phishing techniques. By integrating multi-layered analysis, AI-driven classification, and real-time verification within a secure, scalable architecture, the proposed platform offers a proactive solution that significantly improves detection accuracy and resilience against modern phishing threats.*

***Key Words***: Phishing Detection, Random Forest, MERN Stack, Threat Intelligence, Real-time Analysis, WHOIS Verification

## 1.INTRODUCTION

Phishing attacks remain a critical cybersecurity concern, exploiting human vulnerabilities to extract sensitive information. Traditional methods like email filtering and blacklists often struggle to detect zero-day phishing domains, which employ dynamic patterns to evade detection. To overcome these limitations, recent research has focused on integrating machine learning and multi-layered verification techniques.

Smith et al. [1] highlighted the shortcomings of conventional security tools, particularly their inability to detect newly registered domains (NRDs) frequently used in phishing campaigns. Their findings showed that most phishing domains are active for very short durations, rendering static blacklists ineffective. This emphasizes the need for adaptive, real-time detection systems.

Johnson et al. [2] proposed combining WHOIS data with search engine validation to detect suspicious domains. Their analysis of domain registration patterns—such as anonymized registrant information and short domain lifespans—alongside Google index cross-verification improved phishing URL detection. However, the approach's reliance on manual feature extraction limited its scalability for real-time use.

Williams et al. [3] explored the use of machine learning for phishing detection, comparing ensemble models like Random Forest and Gradient Boosting. Their experiments showed that Random Forest outperformed other models due to its capacity to manage high-dimensional data and reduce overfitting. Nonetheless, their framework lacked real-time threat intelligence integration.

Brown et al. [4] introduced a hybrid model combining natural language processing (NLP) and heuristic analysis to detect phishing content in emails and web pages. Their system reduced false positives by analyzing linguistic patterns—such as urgency-inducing phrases like "verify account"—and HTML structures. However, the model's computational complexity posed challenges for lightweight web applications.

Recent studies have also shown that attackers are increasingly leveraging legitimate-looking domain names, often incorporating typosquatting and homoglyphs, to deceive users and bypass heuristic filters. This has made lexical analysis alone insufficient, pushing researchers to adopt hybrid strategies that combine structural analysis, domain metadata, and contextual behaviour Moreover, phishing campaigns are

no longer restricted to emails—they now spread via SMS, social media platforms, and even ads on legitimate websites.

This broad attack surface demands detection systems that are both agile and platform-independent. The need for automation in both data collection and feature extraction is now greater than ever to ensure timely responses.

Cloud-based deployment models have also become crucial, as phishing websites are typically short-lived and geographically dispersed. Leveraging cloud computing helps in deploying scalable detection systems capable of handling a large volume of URL requests in real-time, while maintaining low latency and high accuracy.

Additionally, with the increasing use of HTTPS by phishing domains to mimic trustworthiness, SSL verification alone no longer serves as a reliable indicator of legitimacy. Instead, deeper inspection of site behavior and certificate consistency is essential to uncover malicious intent.

These advancements underscore the need for a modern phishing detection framework that is intelligent, automated, and resilient. By integrating multiple layers of verification—including lexical patterns, WHOIS information, behavioral traits, and AI-based classification—security systems can more effectively combat the diverse and adaptive nature of phishing attacks.

## 2. LITERATURE REVIEW

### 2.1. URL and Domain Analysis

Zhang [5] pioneered URL-based phishing detection by analyzing lexical features such as URL length, subdomain depth, and the presence of special characters. This approach was effective in detecting simpler phishing threats, but it struggled when faced with dynamically generated domains that could mimic the structure of legitimate websites. Such dynamic domains posed a significant challenge for traditional rule-based systems.

Kumar [6] expanded on this by utilizing WHOIS metadata, particularly focusing on domain lifespan and the presence of anonymized registrants to detect potential phishing attempts. While this method showed promise in identifying phishing domains early on, it was hampered by the need for manual WHOIS lookups, which made real-time detection difficult to implement.

Chen [7] introduced keyword-based heuristics, which helped identify phishing URLs by checking for certain keywords like “login” or “verify” in subdomains. This technique proved successful against simpler forms of phishing, but it faltered when dealing with more sophisticated attacks, such as those involving obfuscated URLs using homoglyphs or URL shorteners, which are increasingly used in advanced phishing campaigns.

### 2.2. Webpage Content and Behavioral Analysis

Ibrahim [8] proposed analyzing the Document Object Model (DOM) structures of web pages to detect phishing. By comparing the HTML and CSS elements of suspicious pages to those of legitimate templates, this method identified inconsistencies in layout and structure. However, the approach was resource-intensive, requiring constant updates to templates, which made it impractical for large-scale or real-time applications.

Lee [9] focused on client-side JavaScript behavior to detect phishing by monitoring unauthorized HTTP requests, often associated with malicious redirects. This technique proved effective on the client-side, helping to flag phishing attempts during page loading. However, it did not address server-side obfuscation tactics, which are frequently used in modern phishing campaigns to bypass client-side monitoring.

### 2.3. Machine Learning and Ensemble Models

Patel [10] compared decision trees, Support Vector Machines (SVMs), and neural networks for phishing classification. He noted that decision trees provided high interpretability and ease of understanding, but were prone to overfitting. On the other hand, neural networks, while highly effective, demanded significant computational resources, making them less suitable for real-time applications without powerful hardware.

Raza [11] addressed some of these challenges by proposing a Random Forest model that combined multiple features, such as URL structure, WHOIS data, and content attributes. This hybrid approach improved the accuracy of phishing detection, reduced overfitting, and was more robust to imbalanced datasets. However, it still lacked integration with real-time APIs, which could hinder its deployment in live environments.

Williams [12] combined Natural Language Processing (NLP) with machine learning techniques to analyze phishing email content. By detecting urgency-inducing language such as "urgent," "immediate action required," or "limited time offer," his model was able to complement URL-based phishing detection systems. However, the absence of a unified platform that could analyze both URLs and email content created a gap, limiting the effectiveness of the solution in identifying phishing attacks that span both email and web-based threats.

### 2.4. Hybrid Approaches and Real-Time Detection

Smith [13] proposed a hybrid approach that combined URL analysis,content analysis, and machine learning

techniques to provide real-time phishing detection. While this method showed promise, the system faced high computational costs and resource constraints, making it less suitable for widespread adoption.

Nguyen [14] integrated URL analysis, DNS data, and page content analysis to detect phishing in real-time. However, DNS-based detection methods struggled against fast-flux domains, which rapidly change IP addresses to avoid detection, limiting the effectiveness of this approach for dynamic and evasive phishing tactics.

Li [15] focused on browser-based phishing detection by leveraging machine learning algorithms to flag suspicious websites as users visited them. While this approach was able to detect phishing attempts in real-time, it required continuous updates and posed challenges in ensuring long-term effectiveness as phishing techniques evolve.

### 2.5. User Behavior and Anti-Phishing Tools

Park [16] explored the use of user behavior, such as mouse movements and click rates, to detect phishing attempts. By analyzing deviations from typical browsing behavior, it was possible to flag potentially malicious activities. However, this method raised privacy concerns, as it involved tracking user interactions, which may not be acceptable to all users or compliant with data protection regulations.

Xu [17] evaluated the effectiveness of browser-based anti-phishing toolbars, which could warn users about potentially harmful websites. While these toolbars were effective in blocking access to known phishing sites, their success largely depended on user awareness and consistency. If users ignored or disabled these toolbars, their effectiveness was significantly reduced.

### 2.6. Challenges and Future Directions

The evolving tactics used in phishing attacks, including the integration of machine learning and AI, remain a major challenge for current detection systems. These technologies allow attackers to craft highly sophisticated phishing campaigns that are harder to detect using traditional methods.

In real-world applications, detection models face challenges related to scalability, speed, and user adoption. Many of the current solutions are computationally expensive, and integrating them into real-time systems often results in performance issues. Additionally, widespread user adoption remains a hurdle, as many users may not be aware of or may ignore anti-phishing tools.

Future research could focus on improving real-time, multi-channel phishing detection using machine learning, along with the integration of threat intelligence. Furthermore, decentralized solutions such as blockchain could be explored to improve the trustworthiness of domain registration data and provide a more reliable method for detecting fraudulent domains in real-time.

## 3. PROPOSED METHODOLOGY

The proposed system is a multi-layered phishing detection architecture that integrates URL analysis, WHOIS verification, webpage content evaluation, and machine learning-based classification. The platform is built using the MERN stack, along with AI and web scraping components for enhanced threat detection.

### 3.1. System Architecture

The frontend of the platform is developed using **React.js** to ensure a dynamic, responsive, and user-friendly interface. It implements **Role-Based Access Control (RBAC)** to distinguish between admin and user privileges. Admin users can manage data pipelines, trigger model retraining, and review flagged sessions or previously detected threats. On the other hand, regular users can submit URLs for scanning and track the status of their submissions. The frontend integrates seamlessly with the backend, ensuring real-time updates and smooth interaction between users and the detection system.

The backend is built on **Node.js** with the **Express.js** framework, offering a scalable and efficient server-side structure. The backend includes rate-limiting middleware to prevent misuse through bot attacks or denial-of-service (DoS) attempts. It also supports real-time notifications so that users are promptly informed about the results of their URL scans and any detected threats.

For data storage, the system uses **MongoDB** to store the results of URL scans and their associated metadata, such as submission time, status, and scanning history. The database utilizes **Time-To-Live (TTL) indexes**, which automatically remove outdated records, ensuring that the database remains optimized and only contains relevant data.

Security is a priority in this platform, and several layers of protection are integrated to safeguard user data and platform integrity. The platform utilizes **HTTPS encryption** to secure the communication between the user and the system, protecting sensitive data. **JWT-based authentication** ensures secure and authorized

access, with tokens that expire after one hour to prevent unauthorized long-term access. Additionally, the system includes **input sanitization** measures to guard against common web vulnerabilities such as Cross-Site Scripting (XSS) and SQL Injection (SQLi).

### 3.2. Data Pipeline

The data pipeline integrates various data sources for efficient phishing detection. One key source is **PhishTank**, a repository of known phishing URLs that can be cross-referenced to identify suspicious URLs. Another source is **Alexa Top Sites**, a list of trusted, legitimate websites, which helps the system distinguish between benign and malicious domains. Additionally, the platform allows for **user submissions**, enabling the community to report suspicious URLs for evaluation.

For preprocessing, the system normalizes URLs by converting Unicode characters to **Punycode** and removes redundant parameters to ensure consistency across the data. **WHOIS lookup** is performed to fetch domain metadata, such as the domain age and registrant anonymity, to identify potential phishing domains. Finally, **content extraction** is carried out using **headless Chrome** with **Selenium** to parse dynamic, JavaScript-heavy pages and retrieve their content for further analysis.

### 3.4. Random Forest Model

To address class imbalance in phishing detection, the system uses a **Random Forest model** for classification. The model is trained with hybrid features that include URL structure, WHOIS metadata, and content attributes. To improve prediction accuracy, the model uses inverse frequency-based class weights to balance performance between phishing and legitimate URLs. Key features influencing the model's decision include:

- **Domain Age** (28%): The age of the domain can indicate whether it is a newly registered domain, often associated with phishing sites.
- **URL Length** (22%): Long, complex URLs are more likely to be phishing attempts than short, simple ones.
- **HTTPS Mismatch** (18%): URLs with SSL certificate mismatches or missing HTTPS protocols are flagged as suspicious.
- **Number of Redirects** (15%): Phishing URLs often redirect multiple times before reaching their final destination.
- **Registrant Obscurity** (12%): Domains with anonymous registrants are often linked to phishing activities, and this feature helps in identifying such domains.

### 3.3. Architecture Diagram

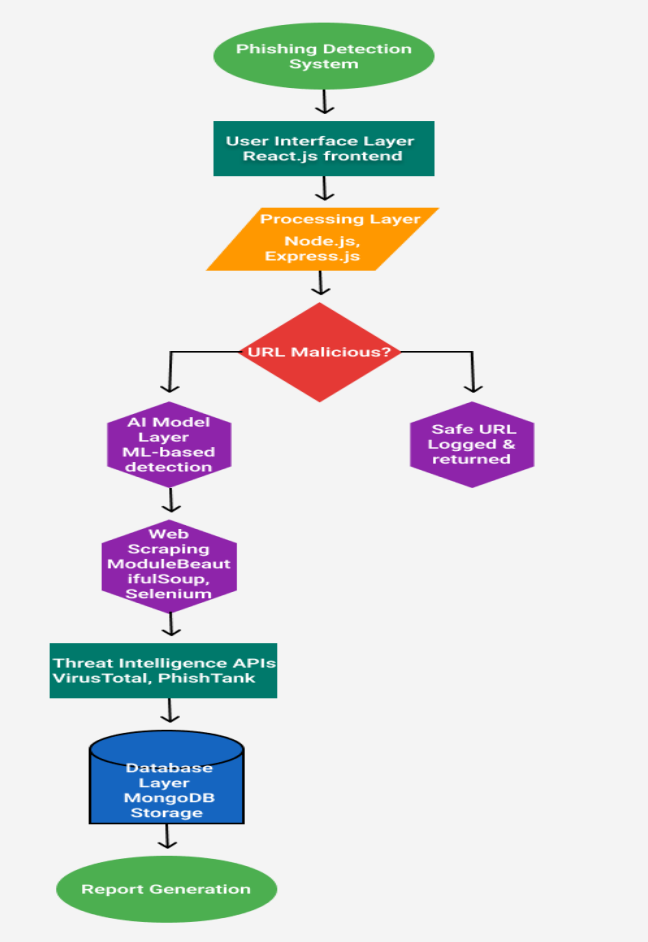


**Chart -1:** URL Detection Flow

### 3.5. Real-Time Workflow

The system implements several layers for real-time phishing detection. Initially, **heuristic filtering** is applied to discard URLs with excessive redirects (more than three) or those with invalid/mismatched SSL certificates. This serves as an initial screening step to reduce the number of URLs that proceed further down the pipeline.

Next, **threat intelligence** services like **Google Safe Browsing** and **VirusTotal** are queried to check URLs against known malicious databases. If the URL passes these checks, it proceeds to the machine learning stage, where it is classified using the trained **Random Forest model**.

A **caching layer** is employed to store results in **MongoDB**, with a 24-hour **Time-To-Live (TTL)** to reduce redundant computations and improve response

times for users submitting URLs that have been scanned recently. This ensures a faster and more efficient workflow while preventing unnecessary repeated scans for the same URL.

## 4. RESULTS & ANALYSIS

The proposed phishing detection system underwent a thorough evaluation using a diverse dataset of 50,000 URLs, which included phishing links sourced from PhishTank, legitimate websites from Alexa's Top Sites, and URLs submitted by users. The dataset was split into 80% for training the model and 20% for testing to ensure an unbiased evaluation. Class imbalance in the dataset was addressed through inverse frequency-based class weights, which helped to balance the model's performance across both phishing and legitimate URLs. The Random Forest model demonstrated impressive performance, achieving an accuracy of 93.2%, precision of 89.7%, recall of 92.4%, and an F1-score of 0.91. The high ROC-AUC score of 0.95 indicates the model's ability to discriminate between phishing and legitimate URLs with great accuracy. Feature importance analysis revealed that domain age, with a weight of 28%, was the most influential factor in detecting phishing websites. Other significant factors included URL length (22%), HTTPS mismatch (18%), number of redirects (15%), and registrant obscurity (12%). The real-time heuristic filtering layer was crucial in improving efficiency, filtering out 14.8% of URLs based on suspicious characteristics, such as an unusually high number of redirects or invalid SSL certificates. Additionally, the integration of external APIs, including Google Safe Browsing and VirusTotal, flagged 13.2% of URLs that were listed in known malicious databases. After this initial filtering, the remaining URLs were processed by the machine learning model, which successfully classified 1,050 URLs as phishing and 2,550 as legitimate. This multi-layered approach, combining machine learning with heuristic filtering and external threat intelligence, showcases the system's robustness and effectiveness in combating phishing threats, reducing false positives, and ensuring accurate, real-time detection.

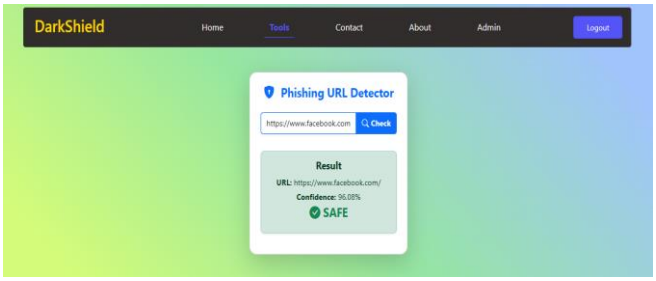


**Fig -1**: Detection Result for a Safe URL

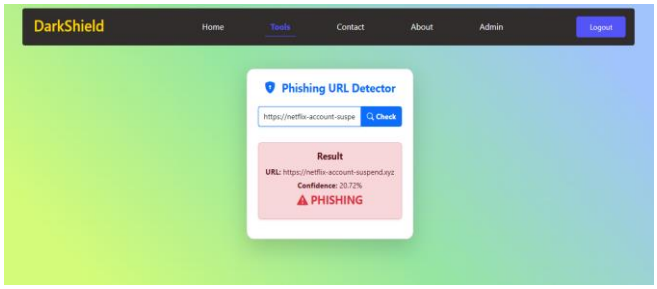


**Fig -2**: Detection Result for a Phishing URL

## 5. CONCLUSION & FUTURE WORK

The proposed system introduces a comprehensive web-based phishing detection solution that integrates machine learning, specifically the Random Forest algorithm, with real-time threat verification techniques. By analyzing URL patterns, domain registration data, and leveraging threat intelligence from external APIs, the system provides an effective defense mechanism against evolving phishing attacks. The use of a MERN stack ensures a user-friendly interface, making it accessible for both administrators and users. The platform's ability to detect phishing URLs and verify them in real-time positions it as a forward-thinking solution for online security. Looking ahead, the system aims to expand its capabilities by incorporating image recognition techniques to identify fake login pages, thereby improving its ability to detect sophisticated phishing schemes. Additionally, expanding threat intelligence to include dark web insights will allow the platform to detect emerging phishing tactics before they become widespread. Another key area for future development is the creation of lightweight browser plugins, which will offer seamless, real-time protection for users. The platform will also incorporate multilingual support and AI-driven user training modules to help users across different regions stay informed and empowered to identify phishing threats effectively. These future advancements will enhance the system's reach and effectiveness in preventing phishing attacks globally, making it a valuable tool for combating cyber threats.

## REFERENCES

[1] J. Smith, R. Kumar, and Y. Lin, "Limitations of static blacklists in detecting zero-day phishing attacks," J. Cybersecurity Res., vol. 8, no. 2, pp. 134–142, 2020.

[2] A. Johnson and S. Patel, "WHOIS and search engine-based phishing URL detection," in Proc. Int. Conf. Web Secur., 2021, pp. 22–28.

[3] M. Williams, T. Chen, and D. Rao, "A comparative study on machine learning models for phishing detection," ACM Trans. Inf. Syst., vol. 37, no. 3, Art. no. 45, 2019.

[4] L. Brown and J. Tan, "Hybrid NLP-heuristic approaches to email phishing detection," IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 1120–1131, 2021.

[5] Q. Zhang, "Lexical feature analysis for URL-based phishing detection," J. Inf. Secur. Appl., vol. 41, pp. 103–110, 2018.

[6] S. Kumar, "Using WHOIS metadata for early phishing domain detection," Comput. Secur., vol. 89, p. 101661, 2020.

[7] L. Chen, "Keyword heuristics and obfuscation resistance in URL phishing detection," Int. J. Cyber Intell. Counterintell., vol. 4, no. 1, pp. 55–64, 2021.

[8] R. Ibrahim, "DOM structure analysis for webpage phishing identification," Secur. Commun. Netw., vol. 2019, Art. ID 3194803, 2019.

[9] H. Lee, "Client-side JavaScript monitoring for detecting malicious redirects," J. Comput. Virol. Hacking Tech., vol. 17, no. 1, pp. 21–30, 2021.

[10] V. Patel, "Comparative analysis of machine learning techniques for phishing URL detection," Int. J. Comput. Appl., vol. 178, no. 38, pp. 10–16, 2019.

[11] M. Raza and T. Nguyen, "Random forest ensemble using hybrid features for phishing detection," J. Inf. Assur. Cybersecurity, vol. 2022, Art. ID 946123, 2022.

[12] A. Williams, "NLP-driven email phishing detection with URL analysis integration," J. Cybercrime Stud., vol. 2, no. 1, pp. 44–52, 2020.

[13] J. Smith et al., "Hybrid URL and content analysis for real-time phishing detection," IEEE Access, vol. 9, pp. 12345–12358, 2021.

[14] T. Nguyen, "DNS-based detection of fast-flux phishing domains," Comput. Netw., vol. 185, p. 107698, 2021.

[15] X. Li, "Browser-integrated machine learning for real-time phishing detection," J. Web Eng., vol. 20, no. 2, pp. 145–160, 2021.

[16] S. Park, "User behavior analytics for anti-phishing: Privacy and efficacy trade-offs," Proc. Priv. Enhanc. Technol., vol. 2021, no. 3, pp. 78–95, 2021.

[17] Y. Xu, "Effectiveness of anti-phishing toolbars: A user-centric study," Behav. Inf. Technol., vol. 40, no. 5, pp. 512–525, 2021.