International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Real-Time Automatic Phishing Detection System

Bhakti Pokale
Electronics and Telematics engineering G.Narayanamma Institute of Technology and Science hyderabad, India

Abstract- — Phishing attacks have become one of the most serious cybersecurity threats worldwide, causing identity theft, financial loss, and data breaches. Attackers use fake websites, emails, and malicious links to trick users into revealing sensitive information. Traditional security mechanisms such as antivirus software and browser filters are often unable to detect newly generated phishing URLs, making users vulnerable to attacks. To address this issue, this project proposes a Real-Time Automatic Phishing Detection System that identifies and blocks phishing links instantly. The system uses Machine Learning techniques, specifically the Random Forest Classifier, to analyze URL features such as length, domain age, and special characters. It operates silently in the background without requiring user intervention, ensuring continuous and seamless protection. The system is developed using Python, Java, JavaScript, Node.js, MongoDB, HTML, and CSS to support multi-platform functionality. It provides real-time alerts and maintains logs of detected threats for further analysis. The proposed solution aims to enhance cybersecurity by offering proactive protection and ensuring a safer digital environment for individuals and organizations.

Keywords— Phishing Detection, Cybersecurity, Machine Learning, Random Forest, URL Analysis, Real-Time Security.

I. INTRODUCTION

Phishing is a cyberattack technique in which attackers deceive users into sharing sensitive information such as passwords, bank details, and personal data. These attacks are commonly carried out through emails, social media platforms, and fake websites that appear legitimate. With the increasing reliance on digital systems, phishing attacks have grown significantly and continue to evolve in complexity.

Traditional security solutions like antivirus software and browser-based filters are not sufficient to handle modern phishing attacks. These systems mainly rely on blacklists and predefined rules, which fail to detect newly created phishing URLs. As a result, users often fall victim to attacks due to a lack of awareness and delayed detection mechanisms.

Therefore, there is a need for a proactive and intelligent system that can detect phishing attempts in real time. This project focuses on developing a solution that automatically identifies and blocks suspicious URLs before users interact with them. The system ensures continuous protection and enhances user safety in the digital environment.

II. LITERATURE REVIEW

Existing phishing detection techniques primarily rely on blacklist-based approaches and heuristic methods. These methods compare URLs against known databases of malicious links. While effective for previously identified threats, they fail to detect newly created phishing websites, limiting their effectiveness in real-world scenarios.

Recent research has introduced Machine Learning techniques for phishing detection, including Decision Trees, Support Vector Machines, and Neural Networks. These models analyze patterns in URLs and website features to classify them as legitimate or phishing. Among these techniques, Random Forest has shown better performance due to its ability to combine multiple decision trees and reduce overfitting.

Despite advancements, many existing systems lack real-time detection and cross-platform compatibility. Most solutions are reactive rather than proactive, detecting threats only after damage has occurred. This project addresses these limitations by implementing a real-time phishing detection system using Machine Learning with improved accuracy and efficiency.

III. METHODOLOGY / PROPOSED SYSTEM

The proposed system is designed to detect phishing URLs in real time using Machine Learning algorithms. It continuously monitors URLs accessed by the user through browsers, applications, or manual input. The system ensures that all links are analyzed before they are opened, preventing potential threats at an early stage.

Feature extraction is a crucial step in the detection process. The system analyzes various URL characteristics such as length, use of special characters, domain age, and IP address presence. These features are used to train the Machine Learning model,

enabling it to distinguish between phishing and legitimate URLs effectively.

The Random Forest Classifier is used for classification due to its high accuracy and robustness. Once a URL is analyzed, the system predicts its nature and takes appropriate action. If the URL is identified as phishing, it is blocked immediately, and a warning is displayed to the user. The system operates silently in the background, ensuring uninterrupted user experience.

IV. SYSTEM DESIGN / ARCHITECTURE

The system follows a client–server architecture that enables efficient processing and real-time detection. The frontend acts as the client, collecting URLs from user activity, including browsing and application usage. This ensures that all links accessed by the user are monitored continuously.

The backend server is responsible for processing and analyzing the collected URLs. It performs feature extraction and sends the data to the Machine Learning model for classification. The Random Forest algorithm evaluates the features and determines whether the URL is safe or malicious.

Once the analysis is complete, the result is sent back to the frontend. If the URL is identified as phishing, an instant warning message is displayed. The system also stores all results in a MongoDB database for logging and reporting purposes. This architecture ensures high accuracy, low latency, and continuous protection.

V. IMPLEMENTATION

The implementation of the system involves the integration of multiple technologies to achieve efficiency and scalability. Python is used for developing the Machine Learning model, utilizing libraries such as Scikit-learn, Pandas, and NumPy. These tools help in training and testing the model for accurate predictions.

The backend is developed using Java and Node.js to handle server-side operations and real-time data processing. APIs are used to facilitate communication between the frontend and backend. The frontend is built using JavaScript, HTML, and CSS, providing a user-friendly interface for interaction and alerts.

MongoDB is used as the database for storing phishing datasets and user activity logs. The system also supports cloud integration for secure data storage and continuous updates. This multi-technology implementation ensures cross-platform compatibility and efficient performance.

VI. RESULTS & DISCUSSION

The system demonstrates effective performance in detecting phishing URLs in real time. It successfully identifies malicious links and provides instant alerts to users, preventing potential security breaches. Compared to traditional methods, the system shows improved accuracy in detecting newly generated phishing URLs.

The use of Machine Learning enhances the system's ability to adapt to evolving phishing techniques. The Random Forest Classifier provides reliable predictions by analyzing multiple features simultaneously. The system also maintains logs, allowing administrators to monitor and analyze phishing attempts.

However, the system has some limitations, such as occasional false positives where legitimate URLs may be flagged as phishing. Additionally, regular dataset updates are required to maintain accuracy. Despite these challenges, the system provides a strong and efficient solution for phishing detection.

VII. CONCLUSION

The Real-Time Automatic Phishing Detection System offers an advanced solution to combat phishing attacks using Machine Learning. It effectively detects and blocks malicious URLs in real time, reducing the risk of identity theft and financial loss. The system ensures continuous protection without requiring user intervention.

The solution is scalable and adaptable to evolving cyber threats, making it suitable for both individuals and organizations. It enhances user trust by providing proactive security and minimizes human error through automation. The lightweight design ensures efficient performance across multiple platforms. Overall, the system contributes to building a safer digital environment. It provides centralized monitoring, strong data security, and reliable performance. The project lays a solid foundation for future advancements in intelligent cybersecurity systems.

VIII. FUTURE SCOPE

The system can be further improved by integrating advanced Artificial Intelligence models such as Deep Learning for higher

accuracy. These models can analyze more complex patterns and improve detection capabilities for evolving phishing attacks.

Future enhancements may include the development of browser extensions and mobile applications for wider accessibility. The system can also be extended to detect other cyber threats such as smishing (SMS phishing) and vishing (voice phishing).

Additionally, the integration of real-time analytics dashboards and enterprise-level security systems can enhance monitoring and management. Continuous updates and adaptive learning mechanisms will ensure that the system remains effective against new and emerging threats.

References

1. IEEE Xplore Digital Library, Cyber Threat Analysis Papers.
2. ResearchGate, Real-Time Phishing Detection Studies.
3. Scopus Indexed Papers on AI-Based Phishing Prevention.
4. NIST, Cybersecurity Framework.
5. CISA, Cybersecurity Reports.
6. OWASP, Security Guidelines.
7. W3C, Web Security Standards.
8. Google Safe Browsing Documentation.
9. Scikit-learn Documentation.
10. TensorFlow Documentation.
11. MongoDB Official Documentation.
12. Node.js Official Documentation.