Journal of Science, Computing and Engineering Research (JSCER)
Volume-9, Issue-3, March 2026.
DOI: *https://do.org/10.46379/jscer.2026.080503*

PHISHING WEBSITE DETECTION TOOL

[1]Mohit Sharma,[2]Pulkit Gupta,[3]Nehal Choudhary,[4] Sahil Soni,[5]Abhijeet Singh, [6]Sonu Singh

[1]Asst. Professor, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India.

[2,3,4,5,6] UG Student, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received : 26 March 2026
Revised : 26 March 2026
Accepted: 27 March 2026
Published: 30 March 2026

Corresponding Author:

Pulkit Gupta

Abstract— Phishing websites are a major cyber threat designed to steal sensitive information by imitating legitimate sites. Traditional blacklist-based methods are ineffective against newly generated phishing domains. This project proposes a **machine learning–based phishing detection system** that analyzes URL and domain-level features to classify websites as legitimate or phishing. Using models such as Random Forest and Logistic Regression, the system achieved an accuracy on benchmark datasets. The trained model is integrated into a Flask application, enabling users to check URLs, and a **Chrome extension** provides real-time alerts while browsing. This dual approach offers both high detection accuracy and practical usability, making it a reliable solution for enhancing online security. Phishing attacks are one of the most common cyber threats that aim to steal sensitive user information such as login credentials, banking details, and personal data by creating fraudulent websites that imitate legitimate ones. Due to the rapid growth of internet usage, phishing websites have become increasingly sophisticated and difficult for users to identify manually. Therefore, an effective automated detection system is required to protect users from such malicious attacks. The developed tool provides users with a simple interface where they can enter a website URL and receive a prediction about its authenticity. The results demonstrate that the system can effectively detect phishing websites and help reduce the risk of cyber fraud.
.

Copyright © 2026: Mohit Sharma, Pulkit Gupta, Abhijeet Singh, Sahil Soni, Nehal Choudhary, Sonu Singh, This is an open access distribution, and reproduction in any medium, provided Access article distributed under the Creative Commons Attribution License the original work is properly cited License, which permits unrestricted use.

Citation: Mohit Sharma, Pulkit Gupta, Abhijeet Singh, Sahil Soni, Nehal Choudhary, Sonu Singh, "PHYSHING WEBSITE DETECTION TOOL", Journal of Science, Computing and Engineering Research, 9(03), March 2026.

I. INTRODUCTION

Phishing websites are designed to mimic legitimate websites such as banking portals, email services, or e-commerce platforms. Unsuspecting users often enter their personal details on these fake websites, which are then captured by attackers for malicious purposes. These attacks can lead to financial losses, identity theft, and serious security breaches.

Traditional phishing detection techniques mainly rely on blacklist databases that store known malicious URLs. However, attackers frequently create new phishing websites, making blacklist-based systems ineffective against newly generated phishing links.

To overcome these limitations, automated detection systems using machine learning and feature analysis have been proposed. These systems analyze multiple characteristics of a website, such as URL structure, domain age, security certificate information, and page behavior, to determine whether the website is safe or malicious.

This project focuses on developing a Phishing Website Detection Tool that helps users identify potentially harmful websites before interacting with them. The system analyzes website features and predicts whether the given URL is phishing or legitimate. The proposed tool improves user awareness and contributes to enhancing online security.

Phishing websites are malicious sites that mimic trusted platforms (banking, social media, e-commerce) to steal sensitive data. They trick users into sharing passwords, credit card numbers, and personal details. Traditional methods like blacklists fail to detect new or rapidly changing phishing sites. With the rise of digital transactions, phishing has become a major cybersecurity threat worldwide. There is a growing need for intelligent, real-time detection systems powered by machine learning and AI.

II. PROBLEM STATEMENT

Phishing attacks have become one of the most common and dangerous forms of cybercrime, targeting users through deceptive websites that mimic legitimate platforms. These fraudulent websites are designed to steal sensitive information such as login credentials, banking details, and personal data. Despite increasing awareness, many users still fall victim due to the difficulty in distinguishing between genuine and malicious URLs..

Phishing attacks have become one of the most common and dangerous forms of cybercrime, targeting users through deceptive websites that mimic legitimate platforms. These fraudulent websites are designed to steal sensitive information such as login credentials, banking details, and personal data.

To address this issue, a phishing website detection tool is required that leverages modern techniques to classify URLs as safe or malicious. The system should be capable of analyzing various URL-based features and providing accurate predictions to help users avoid fraudulent websites. Such a solution can enhance online security and reduce the risk of data breaches for individuals and organizations.

III. PROPOSED METHOD

The proposed system is designed to detect phishing websites by analyzing URL-based features and classifying them as legitimate or malicious. The system leverages machine learning techniques to provide real-time predictions, helping users avoid fraudulent websites. It ensures efficient detection by combining data preprocessing, feature extraction, and model prediction..

A. Dataset

The dataset used for this system consists of a collection of legitimate and phishing URLs gathered from various online sources such as open phishing repositories and trusted websites. Each entry in the dataset includes the URL and its corresponding label (phishing or legitimate). This structured dataset enables the model to learn patterns and distinguish between safe and malicious websites effectively.

B. Data Preprocessing

In this step, the collected URLs are cleaned and prepared for analysis. Unnecessary characters and inconsistencies are removed, and URLs are normalized. Important features such as URL length, presence of special characters, use of HTTPS, number of subdomains, and suspicious keywords are extracted to create a meaningful representation of each URL.

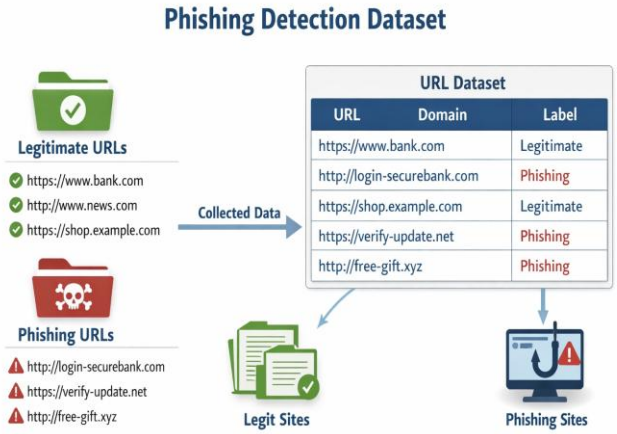

Fig. 1. Sample Phishing Detection Dataset

C. Feature Extraction

The system extracts multiple URL-based features that are indicative of phishing behavior. These features include lexical properties (length, symbols), domain-based features (age, DNS records), and security indicators (HTTPS usage). These extracted features form the input for the machine learning model.

D. Model Training

A machine learning model such as Logistic Regression, Random Forest, or Decision Tree is trained on the processed dataset. The model learns patterns from labeled data and identifies relationships between features and phishing behavior. The trained model is then evaluated using metrics such as accuracy, precision, and recall.

E. Prediction Mechanism

When a user inputs a URL, the system processes it through the same preprocessing and feature extraction pipeline. The trained model then predicts whether the URL is phishing or legitimate. The result is displayed to the user in real-time, ensuring quick and effective decision-making.

F. System Architecture

The system consists of a frontend interface developed using React for user interaction and a backend built with Node.js or Flask to handle requests and model predictions. The trained model is integrated into the backend, enabling seamless communication between the user interface and prediction engine.

G. Deployment

The application is deployed on a cloud platform to ensure accessibility and scalability. Containerization tools like Docker can be used for easy deployment and maintenance. The system is designed to handle multiple user requests efficiently while maintaining fast response times.

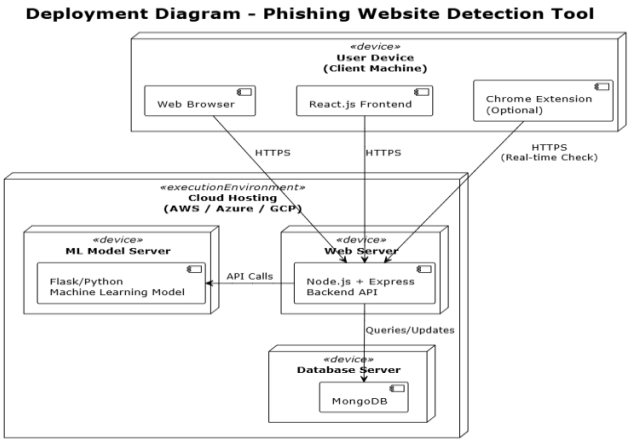

Fig. 2. Deployment Diagram

IV. TECH STACK

A. Frontend Technologies

The user interface of the phishing detection system is developed using React, enabling a responsive and interactive web application. It allows users to input URLs and instantly view prediction results. For styling and improved user experience, technologies such as HTML5, CSS3, and JavaScript are used, along with frameworks like Bootstrap or Material-UI to ensure a clean and responsive design. Axios is utilized for making HTTP requests to communicate with the backend services.

B. Backend Technologies

The backend of the system is built using Node.js with Express or Python with Flask, which handles API requests and integrates the machine learning model. The backend processes incoming URLs, performs feature extraction, and sends them to the trained model for prediction. It ensures smooth communication between the frontend and the model while maintaining efficient request handling.

C. Machine Learning

Machine learning plays a crucial role in detecting phishing websites. Models such as Logistic Regression, Decision Tree, or Random Forest are used to classify URLs as phishing or legitimate. The model is trained on a labeled dataset and evaluated using metrics like accuracy, precision, and recall. Libraries such as Scikit-learn and Pandas are used for model development, data preprocessing, and analysis.

D. Database Technologies

MongoDB is used to store user data, analyzed URLs, and prediction results. It provides a flexible and scalable NoSQL database solution, enabling efficient storage and retrieval of large volumes of URL data for further analysis and improvement of the model.

E. Security

Security is an essential aspect of the system. HTTPS protocols are used to secure communication between client and server. JWT (JSON Web Tokens) can be implemented for authentication and secure API access. Input validation is also performed to prevent malicious data from affecting the system.

F. Version Control and Collaboration

Git is used for version control, with platforms like GitHub enabling collaborative development. It helps in tracking changes, managing code versions, and ensuring smooth teamwork during the development lifecycle.

G. Monitoring and Analytics

Basic monitoring tools and logging mechanisms are implemented to track system performance and errors. These logs help in identifying issues, improving model performance, and enhancing user experience over time.

.

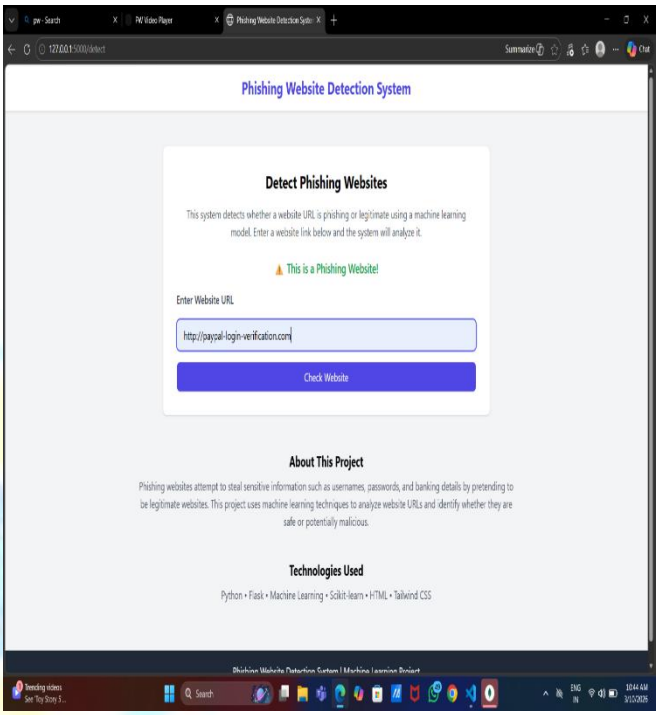

Fig. 3. Website Interface

V. RESULTS

The phishing website detection system demonstrates an effective approach to identifying malicious URLs using machine learning techniques. The model is capable of analyzing URL-based features and providing accurate predictions in real time. Experimental results show that the system achieves good accuracy and can successfully distinguish between legitimate and phishing websites.

However, there is scope for further improvement in terms of accuracy, scalability, and real-world adaptability. Future enhancements may include incorporating more advanced machine learning models, expanding the dataset, and integrating real-time threat intelligence sources. Additionally, improving the user interface and deploying the system as a browser extension can increase usability and reach.

Overall, the system provides a practical and scalable solution for enhancing cybersecurity awareness and protecting users from phishing attacks.

.

VI. CONCLUSION

In conclusion, the development of the phishing website detection system represents a significant step toward enhancing cybersecurity and protecting users from online fraud. By leveraging machine learning techniques and URL-based feature analysis, the system effectively identifies and classifies websites as legitimate or phishing. This approach overcomes the limitations of traditional blacklist-based methods by enabling real-time detection of previously unseen malicious URLs.

The system integrates efficient frontend and backend technologies to provide a seamless user experience, allowing users to quickly verify the safety of a website. Its scalable architecture and deployment on cloud platforms ensure accessibility and reliability for both individual users and organizations. Additionally, the use of secure communication protocols and validation mechanisms helps maintain data integrity and user trust.

Future work will focus on improving model accuracy by incorporating advanced algorithms and larger, more diverse datasets. Enhancements such as real-time threat intelligence integration, browser extension support, and detection of content-based phishing techniques can further strengthen the system. Overall, this project provides a practical and scalable solution for combating phishing attacks and contributes to safer internet usage.

REFERENCES

[1]. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications.

[2]. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. Proceedings of the ACM SIGKDD.

[3]. Sahoo, D., Liu, C., & Hoi, S. C. (2017). Malicious URL detection using machine learning: A survey. ACM Computing Surveys.

[4]. Verma, R., & Das, A. (2017). What's in a URL: Fast feature extraction and malicious URL detection. IEEE Conference on Computer Communications Workshops.

[5]. Scikit-learn Documentation. Machine Learning in Python. Retrieved from https://scikit-learn.org

[6]. Chollet, F. (2015). Keras: Deep Learning for Python.

[7]. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research.

[8]. OpenPhish Dataset. Phishing URL Data Repository. Retrieved from https://openphish.com