

Machine Learning-Based Trust Scoring and Detection of Phishing Websites Using URL Intelligence

Author: **KOTHAPALLI HEMALATHA¹** (MCA student), **Mr.M. SREENIVASU²** (Assistant Professor)

Master Of Computer Applications, GIET ENGINEERING COLLEGE (A), Rajahmundry, AP.

Corresponding Author: Kothapalli Hemalatha

email-id: hemalatha.muthyala.chem@gmail.com

ABSTRACT

Phishing attacks have become one of the most significant cybersecurity threats, causing financial losses and unauthorized access to sensitive information. Traditional phishing detection methods mainly rely on blacklists and rule-based techniques, which are ineffective against newly generated phishing websites. This paper presents a Machine Learning-Based Trust Scoring and Real-Time Detection of Phishing Websites Using URL Intelligence framework to identify malicious websites efficiently.

The proposed system analyzes various URL-based features, including URL length, domain age, SSL certificate status, suspicious keywords, and subdomain information. These features are processed using ensemble machine learning algorithms such as Random Forest and XGBoost to classify websites as legitimate or phishing. In addition to classification, the system generates a trust score ranging from 0 to 100, helping users assess the credibility of a website. The results are displayed through a color-coded interface indicating safe, suspicious, or dangerous websites.

The framework supports real-time URL analysis and provides fast, accurate predictions. Experimental results demonstrate that the proposed approach effectively detects phishing websites, including previously unseen attacks, while improving user awareness and online security.

KEYWORDS: Phishing Detection, URL Intelligence, Machine Learning, Trust Scoring, Real-Time Analysis, Random Forest, XGBoost, Cybersecurity, Feature Engineering, Ensemble Classification.

I. INTRODUCTION

people rely heavily on the Internet for activities such as online banking, shopping, social networking, education, and communication. While these services have made life more convenient, they have also created opportunities for cybercriminals to exploit users through various online attacks. Among these threats, phishing has become one of the most common and dangerous cybersecurity attacks.

Phishing is a fraudulent technique used by attackers to trick users into sharing sensitive information such as

usernames, passwords, credit card details, and banking credentials. Attackers often create fake websites that closely resemble legitimate websites and distribute malicious links through emails, social media messages, advertisements, or text messages. Since these websites appear authentic, many users unknowingly provide their personal information, resulting in financial loss, identity theft, and data breaches.

People rely heavily on the Internet for activities such as online banking, shopping, social networking, education, and communication. While these services have made life more convenient, they have also created opportunities for cybercriminals to exploit users

through various online attacks. Among these threats, phishing has become one of the most common and dangerous cybersecurity attacks.

Phishing is a fraudulent technique used by attackers to trick users into sharing sensitive information such as usernames, passwords, credit card details, and banking credentials.

Attackers often create fake websites that closely resemble legitimate websites and distribute malicious links through emails, social media messages, advertisements, or text messages. Since these websites appear authentic, many users unknowingly provide their personal information, resulting in financial loss, identity theft, and data breaches.

To address this challenge, this project proposes a Machine Learning-Based Trust Scoring and Real-Time Detection of Phishing Websites Using URL Intelligence framework. The system analyzes URL-based features and uses machine learning algorithms such as Random Forest and XGBoost to detect phishing websites in real time. Unlike traditional systems that simply classify websites as safe or unsafe, the proposed framework generates a trust score ranging from 0 to 100, allowing users to better understand the credibility and risk level of a website.

The system provides instant analysis results through a user-friendly interface and categorizes websites into Safe, Suspicious, or Dangerous groups using color-coded indicators. By combining URL intelligence, machine learning, and trust scoring, the proposed solution aims to improve phishing detection accuracy and help users make safer decisions while browsing the Internet.

The main objective of this research is to develop an intelligent, efficient, and scalable phishing detection system that can identify both known and previously unseen phishing websites. The proposed framework contributes to enhancing cybersecurity awareness and reducing the risks associated with online phishing attacks.

Objectives

- Analyze URL and domain-based security features.
- Provide real-time phishing predictions.
- Improve online security awareness among users.
- Reduce risks associated with credential theft and cyber fraud.
- Deliver an easy-to-use web-based security solution.

II. Literature Survey:

2.1 Existing Phishing Detection Systems

Several phishing detection techniques have been developed over the years, including blacklist-based systems, heuristic approaches, and browser security extensions. Blacklist methods compare URLs against known phishing databases, while heuristic techniques analyze predefined website characteristics.

2.2 Limitations of Existing Systems

Most traditional systems depend heavily on previously identified phishing websites. Newly created phishing domains often remain undetected until they are reported and added to blacklists. Furthermore, heuristic methods may generate false positives and require continuous updates.

2.3 Machine Learning in Cybersecurity

Machine learning techniques have become increasingly popular in cybersecurity applications. Algorithms such as Decision Trees, Random Forest, Support Vector Machines, and Logistic Regression can learn patterns from large datasets and identify phishing websites based on extracted features.

2.4 Research Gaps

Many existing phishing detection systems focus only on URL analysis and fail to incorporate broader security indicators. Some systems achieve high accuracy but require significant computational resources. Additionally, many solutions lack user-friendly interfaces for real-time phishing detection.

2.5 Proposed Contribution

The proposed system integrates machine learning-based phishing detection with real-time URL analysis and user-friendly visualization. It provides accurate predictions, security recommendations, and rapid response times while maintaining scalability and ease of use.

III. SYSTEM ANALYSIS AND DESIGN

3.1 Existing System

Traditional phishing detection systems primarily rely on blacklists and signature-based detection methods. These approaches compare URLs against known malicious databases and block suspicious websites.

Limitations

- Unable to detect newly created phishing websites.
- High dependency on blacklist updates.
- Limited adaptability to evolving threats.
- Increased false positive rates.
- Lack of intelligent decision-making.

3.2 Proposed System

The proposed system uses machine learning models trained in phishing and legitimate website datasets. It analyzes URL structure, domain features, SSL

information, and website attributes to classify websites.

Advantages

- Real-time phishing detection.
- High prediction accuracy.
- Detection of previously unseen phishing URLs.
- User-friendly interface.
- Improved cybersecurity awareness.
- Scalable architecture.

3.3 System Architecture

The architecture of the proposed Real-Time Phishing Risk Analyzer is designed as a multi-layered system that integrates user interaction, URL analysis, machine learning-based phishing detection, and results in visualization into a unified framework. The system enables users to analyze URLs in real time and receive risk assessments based on machine learning predictions.

I. User Interface Layer

This layer provides interaction between users and the system. Users can register, log in, submit single or multiple URLs, upload files containing URLs, and view phishing detection results through a web-based interface.

II. URL Processing Layer

Once a URL is submitted, the system validates its format and checks whether it is a properly structured web address. Valid URLs are forwarded for further analysis.

III. Feature Extraction Layer

The feature extraction module collects important security-related characteristics from the URL, including URL length, number of subdomains, presence of suspicious keywords, SSL certificate status, special characters, and domain age. These features form the input dataset for machine learning classification.

IV. Machine Learning Layer

The extracted features are processed using trained machine learning models such as Random Forest and XGBoost. The models analyze patterns associated with phishing attacks and generate prediction scores indicating whether the URL is legitimate or malicious.

V. Risk Assessment Layer

Based on the machine learning prediction, the system calculates a phishing risk score ranging from 0 to 100. The score is categorized into three security levels:

- Green – Safe Website
- Yellow – Suspicious Website

- Red – Dangerous Website

VI. Database Layer

The database stores user information, login credentials, analyzed URLs, prediction history, feedback data, and system logs. This information helps maintain historical records and improve system performance.

VII. Analytics and Dashboard Layer

This module presents URL analysis results, historical search records, phishing statistics, and risk trends through tabular format. Users can review previous analyses and monitor security patterns.

System Architecture Flow:



Fig1. This figure shows the system design



The above figure illustrates the overall workflow of the proposed phishing detection system. The process begins when a user logs into the application and submits a website URL for analysis. The system then extracts important URL intelligence features such as URL length, subdomains, special characters, suspicious keywords, SSL certificate status, HTTPS usage, and domain age. These features help identify patterns that are commonly associated with phishing websites. The generated score is used to classify websites into three categories: **Safe (Green), Suspicious (Yellow), and**

Dangerous (Red). The prediction results are then displayed to the user through an easy-to-understand interface.

The extracted features are processed using machine learning models, namely Random Forest and XGBoost, which have been trained to distinguish between legitimate and malicious websites. Based on the prediction results, the system generates a trust score ranging from 0 to 100, indicating the reliability of the submitted URL.

Finally, all analyzed URLs and prediction records are stored in the database, allowing users to access their previous analyses through the dashboard and history section. This methodology enables real-time phishing detection and helps users make safer browsing decisions.

IV. PROPOSED METHODOLOGY

The proposed Real-Time Phishing Risk Analyzer is designed to identify phishing websites by analyzing URLs and predicting their risk level using machine learning algorithms. The system helps users determine whether a website is safe, suspicious, or dangerous before visiting it.

4.1 Data Collection

The dataset used in this project consists of website links (URLs) along with their corresponding labels indicating whether they are phishing or legitimate. The data was collected from publicly available phishing detection datasets and cybersecurity sources. These labeled URLs served as the foundation for training and testing machine learning models. After collecting the data, relevant features were extracted from the URLs to create a structured dataset suitable for machine learning analysis.

4.2 Data Preprocessing

The collected data is cleaned to remove inconsistencies and missing values. Feature normalization and encoding techniques are applied to prepare the dataset for machine learning algorithms.

4.3 Feature Extraction

Several features are extracted from URLs, including:

- URL Length
- Presence of Special Characters
- Number of Dots in URL
- Domain Age
- SSL Certificate Status
- HTTPS Usage
- IP Address Usage
- Redirection Count

These features help distinguish phishing websites from

legitimate websites.

4.4 Model Training

The dataset is divided into training and testing sets. Machine learning models such as Random Forest and XGBoost are trained using the extracted features. During prediction, both models analyze the URL characteristics, and their accuracy-weighted prediction scores are combined to generate an average risk score. This risk score represents the likelihood of the URL being a phishing website and is used to classify the website as safe, suspicious, or dangerous.

4.5 Prediction and Classification

When a user enters a URL, the system extracts relevant features and passes them to the trained model. The model predicts whether the website is legitimate or phishing and returns the result instantly.

4.6 Result Visualization

Prediction results are displayed through an interactive dashboard showing security scores, risk levels, and recommendations. The results are presented using a color-coded interface, where green indicates a safe website; yellow indicates a suspicious website, and red indicates a potentially dangerous phishing website. Along with the prediction, the system provides the key reasons behind the classification. This helps users understand why a particular website was classified as safe or risky.

V. IMPLEMENTATION

5.1 Technologies Used

Python

Python is the primary programming language used in this project. It is used for implementing machine learning models, feature extraction, and backend processing. Its extensive libraries make development more efficient and reliable.

Flask

Flask is a lightweight web framework used to build the application's backend. It handles user requests and connects the frontend with the machine learning models. Flask also manages routing and server-side operations.

MySQL

MySQL is used to store user information and URL analysis records. It provides secure and efficient data management. The database helps maintain prediction of

history and user activities.

Scikit-Learn

Scikit-Learn is a machine learning library used for training and testing classification models. It provides tools for data preprocessing and model evaluation. The library helps identify phishing websites accurately.

Pandas

Pandas are used for data cleaning, processing, and analysis. It helps organize phishing datasets into structured formats. This simplifies feature extraction and model training.

NumPy

NumPy supports numerical computations and array operations. It improves the efficiency of mathematical calculations required in machine learning. The library is widely used for handling large datasets.

HTML

HTML is used to create the structure of web pages. It provides forms, tables, and layouts for user interaction. HTML serves as the foundation of the application's frontend.

CSS

CSS is used to style and design the web interface. It improves the appearance of pages through colors, fonts, and layouts. CSS helps create a responsive and user-friendly experience.

JavaScript

JavaScript adds interactivity to the web application. It enables dynamic content updates and improves user experience. The language is used for client-side validation and interface functionality.

XGBoost

XGBoost is an advanced machine learning algorithm used for phishing detection. It provides high prediction accuracy and efficient performance. The algorithm helps classify websites based on URL intelligence features.

Joblib

Joblib is used to save and load trained machine learning models. It reduces the need for retraining models every time the application runs. This improves system performance and prediction speed

5.2 Modules

1. User Authentication Module

The User Authentication Module allows users to register and log in to the system securely. It verifies user credentials and provides access to the phishing detection services. This module ensures that each user's analysis history is maintained separately



Fig2. This figure shows the login page

2. URL Prediction Module

After successful login, users are redirected to this page where they can enter a website URL for analysis. When the user clicks the Predict button, the system processes the URL, extracts relevant features, and sends them to the machine learning model for phishing detection.



Fig3. This figure shows the prediction page

3. Result Display Module

Displays the prediction outcome, trust score, and website status. The results are presented in a simple and user-friendly format to help users understand the risk level.



Fig4. This figure shows the result

4. User History Module

The User History Module stores and displays previously analyzed URLs for the user. It allows users to review past predictions, trust scores, and analysis of results. This helps users track their URL analysis activities over time.

VI. RESULTS AND ANALYSIS

The proposed system was successfully developed and tested to detect phishing websites using machine learning and URL intelligence techniques. The application provides a simple and user-friendly way for users to check whether a website is safe before visiting it.

After logging into the system, users can enter a website's URL on the prediction page. Once the **Predict** button is clicked, the system analyzes the URL and processes it using trained machine learning models. Based on the analysis, the system generates a trust score and classifies the website as either safe or potentially malicious.

During testing, the system was able to identify phishing websites effectively by analyzing various URL characteristics. The prediction results were displayed instantly, allowing users to quickly understand the security level of the website. The trust score feature helped provide a clear indication of how reliable a website is, making the results easier to interpret.

The system was tested with different types of URLs, including both legitimate and phishing websites. The results showed that the machine learning models were capable of detecting suspicious links with good accuracy. Websites containing common phishing patterns were successfully identified and flagged as risky.

Another important feature of the system is the user history page. Every URL analyzed by a logged-in user is stored in the database and can be viewed later. This allows users to keep track of their previous analyses and review the results whenever needed.

Your Scan History

URL	Score	Status	Date
https://www.facebook.com/	99.97%	Safe	2025-05-11 10:37:10
protection.mcafee.com/cdr/cdr.html?app=	99.97%	Dangerous	2025-05-07 10:37:52
https://www.who.int/	99.97%	Safe	2025-05-07 10:37:52
https://www.who.int/	99.97%	Dangerous	2025-05-07 10:37:52
http://www.cnn.com/	99.97%	Safe	2025-05-07 10:37:52

[Click to view all](#)

Fig5.This figure shows the user activity

VII. CONCLUSION AND FUTURE SCOPE

Conclusion

The system successfully analyzes website URLs and uses machine learning techniques to determine whether a website is safe or potentially dangerous. By generating a trust score and displaying prediction results in an easy-to-understand format, the application helps users make informed decisions while browsing the internet.

The user authentication, URL prediction, result display, and history tracking modules work together to provide a smooth and reliable user experience.

The testing results showed that the system was able to identify phishing websites effectively and maintain a record of all previous analyses. Overall, the project demonstrates how machine learning and URL intelligence can be used to improve online security and protect users from phishing attacks.

Future Scope

The proposed system can be further enhanced by integrating advanced machine learning and deep learning techniques to improve phishing detection accuracy. Future versions may include a browser extension and mobile application for real-time website analysis.

A user feedback mechanism can be added to collect and learn from incorrect predictions, helping improve model performance over time. The system can also incorporate a sandbox environment to safely analyze suspicious URLs before users access them.

VIII. REFERENCES

- K. Omari, "Phishing Detection Using Gradient Boosting Classifier," *Procedia Computer Science*, vol. 230, pp. 120–127, 2023.
- M. Reggi Tresna Utami, M. H. Hilman, and S. Yazid, "Enhancing Phishing Detection: Integrating XGBoost with Feature Selection Techniques," *SSRN*, 2025.
- S. Banoth, B. Chandragiri, P. Ramadugala, H. Oraganti, and J. Jayanth, "Detecting Phishing URL Using Random Forest Classifier," *World Journal of Advanced Research and Reviews*, vol. 25, no. 2, pp. 762–769, 2025
- A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023

O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine Learning Based Phishing Detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

M. Almseidin, A. Alzubi, S. Kovacs, and M. Alkasassbeh, "URL-Based Phishing Detection Using Machine Learning Techniques," *IEEE International Conference on Intelligent Systems and Computer Vision*, 2024.

Y.Li, X. Wang, and H. Zhang, "A Deep Learning Framework for Real-Time Phishing Website Detection," *IEEE Access*, vol. 12, pp. 45621–45635, 2024.

S. Kumar and R. Gupta, "Trust Score Based Detection of Malicious URLs Using Ensemble Learning," *IEEE International Conference on Cyber Security and Resilience*, 2024.

A. Sharma, P. Singh, and V. Kumar, "Phishing Website Detection Using URL Intelligence and XGBoost Classifier," *IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2025.

J. Chen, K. Liu, and M. Zhao, "Machine Learning Approaches for Detecting Zero-Day Phishing Attacks," *IEEE Access*, vol. 13, pp. 11234–11248, 2025.