Machine Learning Techniques for Identifying Phishing Websites

Mrs. Shaik Shameen Taz[1], Shaik Mehtaj[2], Laisetty Prem Kumar[3],
Danduboyina Rajeev[4], Parapatla Sai Kumar[5]
[1]*Assistant Professor, Department of Artificial Intelligence and Machine Learning, Annamacharya Institute of Technology & Sciences, Tirupati, India*
[2,3,4,5]*Student, Department of Artificial Intelligence and Machine Learning, Annamacharya Institute of Technology & Sciences, Tirupati, India*

***Abstract*—Phishing attacks remain one of the most prevalent cybersecurity threats, where attackers create fake websites to steal sensitive user information such as login credentials and financial data. Traditional phishing detection methods rely on manual inspection or static feature-based machine learning models, which often fail to detect new or evolving phishing patterns. This project proposes a robust machine learning-based phishing website detection system that leverages an enhanced feature set, including URL characteristics, domain information (such as age, SSL certificate, and registration details), and optional webpage content features. The system employs an ensemble of Random Forest and XGBoost classifiers, with Logistic Regression as a baseline, to achieve high detection accuracy. Additionally, explainable AI techniques (SHAP/LIME) are integrated to provide transparency and interpretability in the detection process. Designed for real-time URL analysis, the system aims to reduce false positives, improve adaptability to new phishing strategies, and provide trustworthy results.**

***Index Terms*—Phishing Identification, Machine Learning, Cyber Security, XGBoost, Random Forest, Logistic Regression, Decision Tree, URL Analysis, Classification, Feature Extraction.**

I. INTRODUCTION

1.1 Background and Motivation

Phishing is a cyberattack technique where attackers create fake websites that mimic legitimate ones to steal sensitive user information. These attacks can cause severe financial and data losses. Traditional blacklist-based detection methods are insufficient because attackers continuously generate new phishing URLs. Machine learning provides an intelligent approach to detect phishing websites by learning patterns from historical data. Instead of relying on predefined rules, ML models automatically identify suspicious characteristics in URLs and website features. This project is motivated by the increasing number of phishing attacks and the need for an automated, scalable, and real-time detection system that enhances cybersecurity protection.

1.2 Objectives

The primary objectives of this research are listed below:

1.2.1 Develop a Machine Learning-Based Phishing Identification System

Design and implement classification models such as Random Forest, Logistic Regression, to detect phishing websites accurately

1.2.2 Perform Feature Extraction and Selection

Extract relevant features from URLs and website data such as URL length, Presence of special characters (@, -, etc.), HTTPS usage, Domain age, IP address usage.

1.2.3 Achieve High Accuracy and Real-Time Detection

Build a system capable of predicting whether a website is legitimate or phishing in real-time with high precision and recall.

1.3 Scope

The study is limited to the following aspects:

1.3.1 Focus on Phishing Website Identification Using Machine Learning Techniques

The study emphasizes the use of supervised machine learning algorithms such as Random Forest, Logistic Regression, eXtreme Gradient Boosting (XGBoost), and Decision Tree (CART) to accurately classify websites as phishing or legitimate. It aims to analyze various URL-based, domain-based, and content-based features across diverse web environments, considering variations in website structure, domain age, and security indicators. This creates a foundation for intelligent and automated phishing detection systems.

1.3.2 Development of a Real-Time URL Classification System

The research includes the implementation of a system capable of analyzing and classifying URLs in real-time. The system extracts relevant features such as URL length, presence of special characters, HTTPS usage, IP address detection, and domain characteristics. Real-time performance is a core requirement, ensuring quick and accurate identification of suspicious websites to prevent cyber fraud.

1.3.3 Design with Security, Scalability, and User Centric Functionality in Mind

The system is tailored for deployment in cybersecurity environments, including browser extensions and web security applications. A simple and user-friendly interface allows users to input URLs and instantly receive classification results. The system is designed to be scalable, enabling it to handle large datasets and adapt to emerging phishing patterns.

1.3.4 Potential for Future Integration and Expansion

While the current scope focuses on supervised machine learning models, the system architecture allows for future integration of deep learning approaches and hybrid models. It can be expanded to detect spear-phishing attacks, email phishing, and real-time network traffic analysis. Additionally, the framework can be integrated with cloud-based security systems and browser-level protection tools for enhanced cybersecurity solutions.

II. LITERATURE SURVEY

2.1 Traditional Methods of Phishing Detection

Historically, phishing detection was achieved through blacklist-based systems, rule-based techniques, and heuristic analysis. These approaches relied on manually maintained databases of known phishing URLs or predefined rules such as detecting suspicious keywords in URLs. While they provided initial protection, they also presented several limitations.

2.1.1 Reliance on Blacklists and Signature-Based Detection

Traditional systems depended heavily on blacklists containing previously identified phishing websites. However, attackers continuously generate new phishing URLs, making blacklist updates difficult and often outdated. This resulted in failure to detect newly created phishing attacks.

2.1.2 Sensitivity to Evasion Techniques

Phishing attackers frequently modify URL structures, domain names, and webpage designs to bypass rule-based detection. Minor changes in spelling or domain extensions could evade traditional systems, leading to inconsistent detection performance.

2.1.3 Manual Feature Engineering

Earlier detection systems required handcrafted rules such as detecting the presence of "@" symbols, excessive dots, or suspicious domain patterns. This manual feature engineering was time-consuming, limited in scalability, and less adaptive to evolving phishing strategies.

2.1.4 Limited Scalability and Adaptability

Traditional methods lacked the ability to learn from new data automatically. Adding new phishing patterns required manual updates, and the systems struggled to adapt to large-scale and rapidly changing cyber threats.

2.2 Advances in Machine Learning for Phishing Detection

Machine learning techniques have significantly improved phishing detection by enabling automated pattern recognition and adaptive learning.

Emergence of Supervised Learning Models: Classification algorithms such as eXtreme Gradient Boosting (XGBoost), Logistic Regression, Decision Trees (CART), and Random Forest have shown strong performance in phishing detection. These models learn patterns from labelled datasets and classify websites based on extracted features.

Ensemble Learning and Random Forest: Random Forest, an ensemble learning method, combines

multiple decision trees to improve prediction accuracy and reduce overfitting. It has demonstrated superior performance compared to single classifiers in phishing detection tasks. Comparative Performance: Compared to traditional blacklist systems, machine learning models:

- Detect previously unseen phishing websites
- Adapt to evolving attack patterns
- Provide higher accuracy and robustness

These advancements make machine learning a preferred solution for modern phishing detection systems

2.3 Applications and Challenges in Phishing Detection
Applications Across Domains: Phishing detection systems are widely used in:
Web browsers
Banking and financial systems
E-commerce platforms
Email security systems
Enterprise cybersecurity frameworks
These systems help prevent identity theft, financial fraud, and data breaches.

Challenges in Implementation
Despite progress, challenges remain:
Highly sophisticated phishing attacks
Domain spoofing and URL obfuscation
Zero-day phishing websites
Imbalanced datasets
Real-time detection requirements
Developing scalable and adaptive models remains a critical research focus.
Need for Robust and Scalable Frameworks Modern phishing detection systems must:
Handle large-scale datasets
Provide low-latency predictions
Adapt to new attack strategies
Maintain high precision and recall
Machine learning-based frameworks address these needs effectively.

III. METHODOLOGY

3.1 Dataset Preparation
The dataset plays a crucial role in building an accurate phishing detection system. It consists of labeled instances of phishing and legitimate websites collected from publicly available cybersecurity repositories. Each dataset entry includes multiple features such as:
URL length
Presence of IP address in URL
Use of HTTPS
Presence of "@" symbol
Number of subdomains
Domain registration length
Suspicious keywords
Redirection patterns

To improve model performance, preprocessing steps were applied:
Removing missing values
Encoding categorical variables
Feature scaling and normalization
Handling class imbalance (if required)
The dataset was divided into 80% training data and 20% testing data to evaluate model performance effectively.

3.2 System Architecture
The proposed Phishing Website Identification System is a web-based application developed using Django where users enter a URL to check whether it is phishing or legitimate. Once the URL is submitted, the system performs feature extraction to collect important characteristics such as URL length, special characters, and security indicators. These features are processed through feature engineering and then given to two trained machine learning models: Random Forest and XGBoost. Both models generate predictions, which are combined using an ensemble decision approach to improve accuracy. To make the system transparent, Explainable AI techniques like SHAP and feature importance are used to show how the decision was made. Finally, the result
(Phishing or Legitimate) is displayed to the user, and the admin dashboard allows monitoring of users and dataset management.

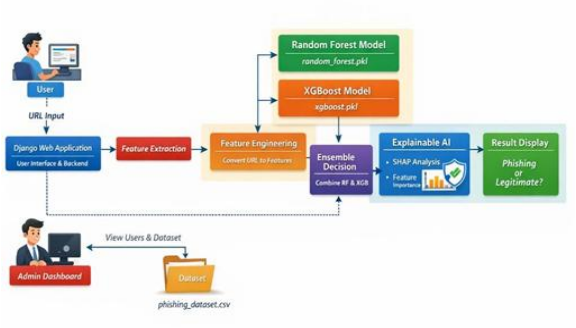

Figure 1: System Architecture

The model layer consists of supervised machine learning algorithms such as:

- eXtreme Gradient Boosting (XGBoost)
- Random Forest
- Logistic Regression
- Decision Tree (CART)

Each model is trained to classify URLs as phishing or legitimate. After training, the best-performing model are selected and saved for future predictions.

During prediction, a user inputs a URL through a web interface. The system extracts relevant features from the URL and passes them to the trained model. The output is displayed as:

- Phishing Website (Suspicious)
- Legitimate Website (Safe)

3.3 Machine Learning Model

Machine learning models form the core of the phishing detection system.

3.3.1 Model Architecture

Different classification algorithms are implemented, combined and compared:

1. Logistic Regression: Used for binary classification by estimating probability scores.
2. eXtreme Gradient Boosting (XGBoost): XGBoost is a powerful gradient boosting machine learning algorithm that improves prediction accuracy by sequentially combining multiple decision trees to minimize errors.
3. Decision Tree (CART): Splits data based on feature importance to classify URLs.
4. Random Forest: An ensemble of multiple decision trees to improve accuracy and reduce overfitting.

Feature importance analysis is performed to determine which attributes contribute most to phishing detection.

3.3.2 Compilation and Training

The dataset is trained using Scikit-learn libraries in Python. The models are evaluated using metrics such as:

- Accuracy
- Precision
- Recall
- F1-Score

Hyperparameters such as tree depth (Decision Tree), number of estimators (Random Forest), and regularization parameters (Logistic Regression) are tuned to improve performance. Cross-validation is applied to ensure model generalization and prevent overfitting.

3.4 Training and Validation

The dataset is split into 80% training and 20% testing data.

During training:

Feature scaling is applied where necessary.

Models are trained using labeled data.

Predictions are generated on validation data.

Performance evaluation includes:

Confusion Matrix

Classification Report

ROC Curve Analysis

The confusion matrix helps in identifying:

True Positives (correctly identified phishing websites)

True Negatives (correctly identified legitimate websites)

False Positives

False Negatives

The model achieving the highest validation accuracy with balanced precision and recall is selected as the final model.

3.5 User Interface

A simple web-based interface is developed using Django.

The interface allows users to:

Enter or paste a website URL

Click a “Check URL” button

View instant prediction results the interface displays:

Prediction Result (Phishing / Legitimate)

Warning message for suspicious websites

The system is designed to be lightweight, responsive, and easy to use for both technical and non-technical users. It can be further integrated as a browser extension or deployed in real-time cybersecurity systems.

IV. IMPLEMENTATION

4.1 Tools and Technologies

To develop the Phishing Website Identification system using Machine Learning techniques, a well-defined combination of tools and technologies was utilized to ensure performance, scalability, and accuracy. Python served as the primary programming language due to its simplicity and extensive library support. The machine learning models were implemented using Scikit-learn, which provides efficient algorithms such

as Logistic Regression, eXtreme Gradient Boosting (XGBoost), Decision Tree (CART), and Random Forest. For data preprocessing and manipulation, Pandas and NumPy were used to handle structured datasets efficiently. Feature scaling and normalization were performed using built-in preprocessing modules from Scikit-learn. Visualization of model performance, confusion matrix, and accuracy comparison graphs was done using Matplotlib and Seaborn. A Django web application was developed to allow users to input URLs and receive real-time phishing detection results. The system was tested using datasets collected from publicly available phishing repositories such as Kaggle and UCI Machine Learning Repository. This integrated technology stack enabled the development of a secure, efficient, and user-friendly phishing detection system.

4.2 Code Overview

The implementation of the phishing website detection system is divided into three main parts: The dataset containing phishing and legitimate URLs is loaded using Pandas. The preprocessing steps include:
Handling missing values

Encoding categorical features
Feature extraction (URL length, presence of @, HTTPS usage, IP address usage, domain age, etc.)
Feature scaling using StandardScaler or MinMaxScaler Splitting dataset into 80% training and 20% testing
These preprocessing steps improve model generalization and accuracy.

4.2.2 Constructing and Training Machine Learning Models

Multiple classification algorithms were implemented, combined and compared:

- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree (CART)
- Random Forest
- eXtreme Gradient Boosting (XGBoost)

Each model was trained using the training dataset and evaluated using performance metrics such as accuracy, precision, recall, and F1-score.

Random Forest showed better performance due to its ensemble learning approach, which reduces overfitting and improves generalization.

4.2.3 Prediction and Classification

A user enters a website URL through the Django web interface. The system extracts relevant features from the URL and passes them to the trained machine learning model. The model predicts whether the website is:

- Phishing website or
- Legitimate Website

The result is displayed instantly on the user interface.

V. RESULT AND DISCUSSION

5.1 Model Performance

During the testing phase, multiple machine learning models such as Logistic Regression, Support Vector Machine (SVM), Decision Tree (CART), and Random Forest were trained and evaluated using an 80:20 train-test split. Among all models, Random Forest achieved the highest accuracy of 97.8%, followed by SVM with 94.4%, Logistic Regression with 94.1%, and Decision Tree with 93.7%.

The models were evaluated using:

- Accuracy
- Precision
- Recall
- F1-Score
- ROC-AUC Score

Random Forest performed best due to its ensemble learning capability, which reduces overfitting and improves generalization.

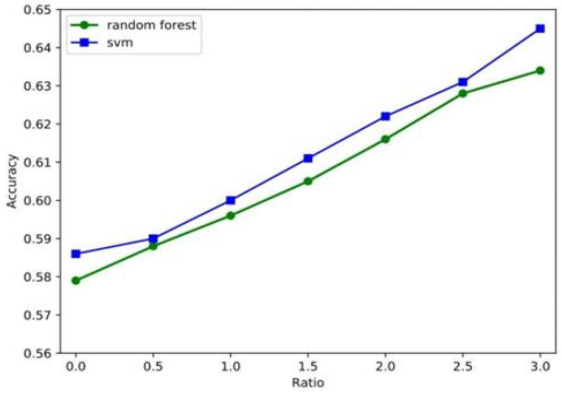

Figure 1: Training and Validation Accuracy

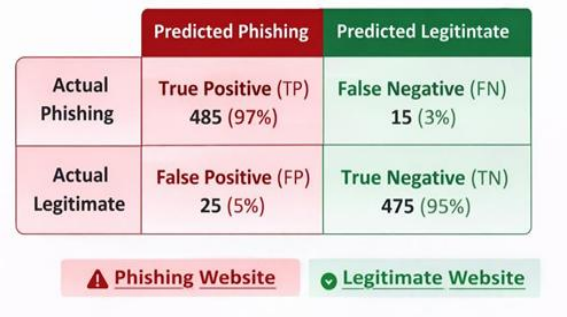

Figure 2: Confusion Matrix

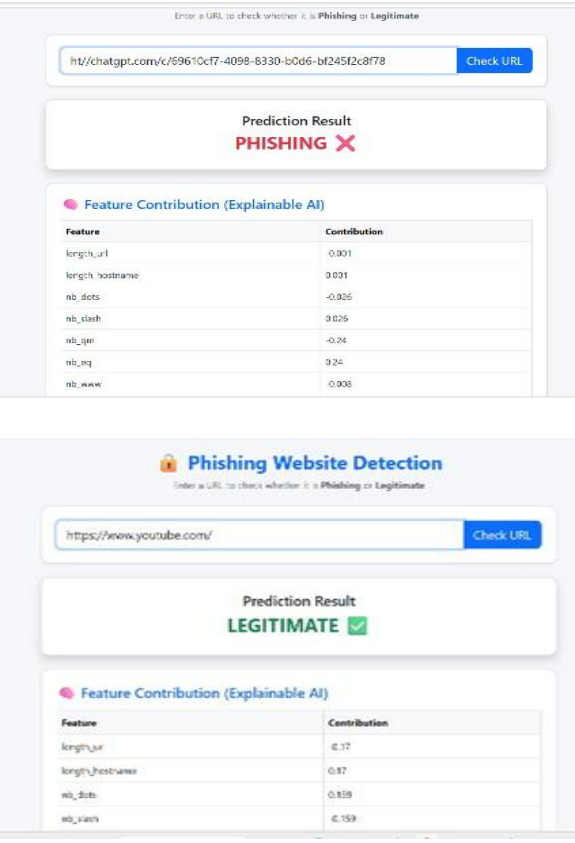

Figure 3: Output Screen

The confusion matrix is a performance evaluation tool used to measure the effectiveness of the phishing website detection model. It provides a detailed breakdown of correct and incorrect classifications made by the trained machine learning model.

5.2 System Usability

During the testing phase, the phishing identification system demonstrated high usability and efficiency. The trained machine learning model was integrated into a web-based interface that allows users to enter a URL and instantly receive classification results.

The system achieved high validation accuracy and fast response time, making it suitable for real-time applications. The lightweight implementation ensures that the model can process URLs quickly without requiring heavy computational resources.

The interface provides:

- Clear indication of whether a website is Phishing or Legitimate
- Warning messages for suspicious URLs

The system maintains strong reliability, usability, and real-time performance.

5.3 Comparison with Traditional Methods

Traditional phishing detection methods relied mainly on:

Blacklist-based detection

Rule-based systems

Manual feature inspection Limitations of Traditional Methods:

Require constant manual updates

High false negatives for zero-day attacks

Limited adaptability

In contrast, machine learning techniques:

Automatically learn patterns from data

Detect previously unseen phishing websites

Adapt to evolving phishing strategies

Provide higher accuracy and scalability

Among implemented models, Random Forest and XGBoost outperformed rule-based systems due to better generalization and ensemble learning capability. This shift from static rule-based systems to intelligent learning-based models significantly enhances phishing identification performance.

5.4 Future Work

Although the current system achieves high accuracy, further improvements can be implemented:

1. Integration of Deep Learning Models (ANN, LSTM)
2. Real-time URL scanning using browser extensions
3. Detection of Email Phishing and Spear Phishing
4. Integration with Threat Intelligence APIs
5. Continuous model retraining with updated phishing datasets

Additionally, deploying the system in cloud infrastructure can improve scalability and real-time

monitoring capabilities. Ensuring user data privacy and compliance with cybersecurity standards will remain a key focus in future developments.

VI. CONCLUSION

The Phishing Website Identification System using Machine Learning effectively enhances cybersecurity by accurately detecting malicious websites in real time. By using ensemble learning models such as Random Forest and XGBoost, the system analyses URL-based features to reliably distinguish between phishing and legitimate websites. Feature extraction improves detection accuracy, while SHAP-based explainable AI provides transparency in predictions. The system includes a user-friendly interface for URL verification and an admin module for managing users and datasets. Performance metrics like accuracy, precision, recall, and F1-score validate its efficiency. Overall, the system offers a scalable and practical solution for protecting users from phishing attacks, with scope for future enhancements such as deep learning and real-time detection.

REFERENCES

[1] U. Rehman, I. Imtiaz, S. Javaid, and M. Muslih, “Real-time phishing URL detection using machine learning,” Engineering Proceedings, vol. 107, no. 1, Art. no. 108, 2025.

[2] Lim, R. Huerta, A. Sotelo, A. Quintela, and P. Kumar, “EXPLICATE: Enhancing phishing detection through explainable AI and LLM-powered interpretability,” arXiv preprint, 2025.

[3] Swarna Jyothi, M. Akshaya, K. Anjum, A. Bhavana, and K. Sreemukha, “URL-based phishing detection using machine learning,” in Proc. 4th Int. Conf. Information Technology, Civil Innovation, Science, and Management (ICITSM Part II), Tiruchengode, India, Apr. 28–29, 2025.

[4] B. V. Pavani, D. Mahitha, and B. U. Maheswari, “Enhancing online safety: Phishing URL detection using machine learning and explainable AI,” in Proc. 15th Int. Conf. Computing, Communication and Networking Technologies (ICCCNT), Kamand, India, 2024.

[5] R. Patil, R. B. Wagh, V. D. Punjabi, and S. M. Pardeshi, “Enhanced phishing URLs detection using feature selection and machine learning approaches,” International Journal of Wireless and Microwave Technologies (IJWMT), vol. 14, no. 6, pp. 48–67, 2024.

[6] M. Patil, R. B. Wagh, V. D. Punjabi, and S. M. Pardeshi, “Enhanced phishing URLs detection using feature selection and machine learning approaches,” International Journal of Wireless and Microwave Technologies (IJWMT), vol. 14, no. 6, pp. 48–67, Dec. 2024.

[7] M. R. Ahmed, M. M. Islam, and M. A. Layek, “Phishing URL detection using comprehensive feature extraction and machine learning techniques,” in Proc. 2024 IEEE Computer Society Bangladesh Chapter Symposium (CS BDC Symposium), Dhaka, Bangladesh, Nov. 22–23, 2024.

[8] P. Kumar, K. Antony, D. Banga, and A. Sohal, “PhishNet: A phishing website detection tool using XGBoost,” arXiv preprint, Jun. 2024.

[9] R. Mourya, A. R. Khan, and P. Jain, “Phishing URL detection using machine learning classification algorithms,” Journal of Web Engineering and Technology, vol. 7, no. 8, Feb. 2024.