www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 1582-1587

e-ISSN :
2583-1062
Impact
Factor :
7.001

IMPROVING ONLINE SAFETY WITH MACHINE LEARNING-BASED PHISHING DETECTION

Vasanth Kalyan. C[1], B. V. V. Satyanarayana[2], A. V. V. Laxman[3], A. V. S. Amarnath[4], Dr. G. Hariharan[5]

[1,2,3,4]Department of AI & ML Malla Reddy University, Hyderabad, India.

[5]Associate Professor Department of AI & ML Malla Reddy University, Hyderabad, India.

ABSTRACT

The ability to detect phishing websites accurately is crucial for online security. Traditional detection methods rely on blacklists, which are often ineffective against newly generated phishing sites. To address this challenge, this study presents a machine learning-based approach that integrates feature extraction with ensemble classification. The dataset is created by collecting phishing URLs from PhishTank and legitimate ones from the University of New Brunswick. Key features such as Address Bar, Domain-based, and HTML & JavaScript-based characteristics are extracted. The classification models, including Random Forest, Support Vector Machines, and XGBoost, are trained to improve phishing detection accuracy. To enhance robustness, a stacking ensemble technique is employed, combining predictions from multiple classifiers. The proposed framework effectively identifies phishing websites while reducing false positives. By automating feature extraction and classification, this approach improves detection accuracy, reduces response time, and enhances scalability for real-time threat analysis. The integration of machine learning establishes a high-performance phishing detection system, enabling faster and more reliable cybersecurity measures. Future extensions include deploying the model as a browser extension or a user-friendly Flask-based web application for real-time phishing detection.

Keywords— Phishing Detection, Machine Learning, URL Classification, Feature Extraction, Random Forest, Support Vector Machines (SVM), XGBoost, Cybersecurity, Browser Extension, Real-Time Threat Detection.

1. INTRODUCTION

Phishing attacks have become a major cybersecurity threat, deceiving users into revealing sensitive information through fraudulent websites. Traditional rule-based detection methods, relying on predefined characteristics like suspicious URLs and deceptive domain names, struggle against evolving phishing tactics. Attackers continuously refine their techniques, making it difficult for static systems to keep up. These limitations highlight the need for adaptive approaches capable of identifying sophisticated phishing attempts beyond predefined patterns.

Machine learning (ML) and deep learning (DL) have revolutionized phishing detection by enabling systems to analyze large datasets and recognize intricate patterns. Unlike rule-based methods, ML and DL models learn from vast amounts of data, identifying subtle differences in HTML structure, user behavior, and traffic flow. Neural networks process high-dimensional data to enhance detection accuracy. The adaptability of these models allows them to detect novel phishing strategies that traditional methods fail to recognize.

However, several challenges hinder real-world implementation. Scalability is a concern as phishing attacks vary significantly in structure and appearance. Many models rely on limited datasets, leading to reduced effectiveness against new phishing tactics. Additionally, the fast-evolving nature of cyber threats demands continuous retraining to maintain accuracy. Another issue is the interpretability of ML models, particularly deep learning, which often functions as a "black box," making it difficult for cybersecurity professionals to understand the decision-making process. This lack of transparency raises concerns about biases, errors, and vulnerability to adversarial attacks.

Phishing detection models also struggle with dataset limitations. Many datasets contain only known phishing sites and fail to capture the full range of emerging threats. Phishing techniques vary based on region, language, and targeted demographics, requiring more diverse training data for improved robustness. Addressing these limitations necessitates the development of sophisticated models incorporating multiple features, such as URL analysis, content inspection, and behavioral patterns. Scalable and adaptable models capable of handling emerging threats are essential. Enhancing interpretability ensures trust in automated phishing detection.

This project aims to build an intelligent phishing detection system using machine learning, improving classification accuracy through models like Random Forest, Support Vector Machines (SVM), and XGBoost. Stacking these models minimizes false positives and negatives. Feature-based analysis, including URL, domain, and HTML attributes, enables the detection of subtle patterns beyond traditional blacklisting.

Despite advancements, certain limitations persist. The model's accuracy depends on training data quality, requiring frequent updates to detect new phishing tactics. False positives and false negatives remain challenges, potentially causing inconvenience or security risks. Black-box models like XGBoost and Random Forest lack interpretability, making it difficult to justify decisions. Scalability is another concern, especially for real-time detection in high-traffic environments. Attackers employ domain generation algorithms, content cloaking, and adversarial attacks to evade detection, necessitating continuous model improvements. Additionally, reliance on internet connectivity limits usability in restricted network environments.

By leveraging machine learning and deep learning, this project enhances phishing detection beyond traditional methods, offering a scalable, adaptive, and high-accuracy solution. With continuous improvements, it aims to strengthen cybersecurity defenses and protect users from fraudulent online threats.

2. PROBLEM STATEMENT

Phishing websites are deceptive web pages designed to impersonate legitimate platforms to steal sensitive user information, including login credentials, credit card details, and personal data. Cybercriminals use various techniques, such as email spoofing, social engineering, and malicious links, to lure users into interacting with these fraudulent sites. Despite advances in cybersecurity, phishing remains a significant threat due to its evolving nature and ability to bypass traditional security measures. Conventional detection methods, such as blacklists and rule-based systems, are limited in identifying newly created phishing websites, as attackers frequently alter URLs and website structures. Additionally, manual analysis is time-consuming and ineffective for large-scale monitoring. AI-driven phishing detection solutions utilizing machine learning and deep learning offer promising approaches to identifying malicious websites based on URL patterns, content analysis, and website behavior. Automation is essential to enhance accuracy, speed, and security in real-world applications.

2.1 Existing System:

Phishing detection has been extensively studied, with various systems developed to identify fraudulent activities. Traditional methods rely on blacklists, heuristic filtering, and rule-based approaches, flagging malicious URLs based on predefined patterns. While effective, they struggle with zero-day attacks due to their reliance on continuous updates. Machine learning techniques leverage lexical, host-based, and content-based features to classify URLs, employing classifiers like Random Forest, Decision Trees, and SVMs. Deep learning models, including CNNs and LSTMs, enhance detection but demand high computational resources, making real-time implementation challenging. Some systems focus on email-based detection, analyzing sender behavior and content patterns. Despite advancements, existing systems face limitations such as high false positive rates, lack of interpretability, and adaptability issues due to evolving phishing techniques. Many approaches require frequent retraining, struggle with scalability, and are computationally expensive. An efficient, scalable, and adaptable phishing detection system is needed for real-world applications.

2.2 Limitations:

- **Dependence on Training Data** – The model's accuracy relies on dataset quality and diversity. Limited phishing samples can reduce its ability to detect new attacks, requiring frequent updates.
- **False Positives & Negatives** – Misclassifications may occur, flagging legitimate sites as phishing or missing sophisticated attacks, leading to security risks or user inconvenience.
- **Limited Explainability** – Models like XGBoost and Random Forest act as black boxes, making it hard for users to understand why a site was flagged.
- **Scalability Challenges** – Handling large volumes of URLs in real-time requires optimized processing for browser extensions or enterprise security.
- **Evasion by Attackers** – Cybercriminals use domain generation, cloaking, and adversarial attacks to bypass detection, requiring continuous model updates.
- **Internet Dependency** – The Flask-based web app requires an active connection, limiting usability in offline or restricted environments.

2.3 Architecture:

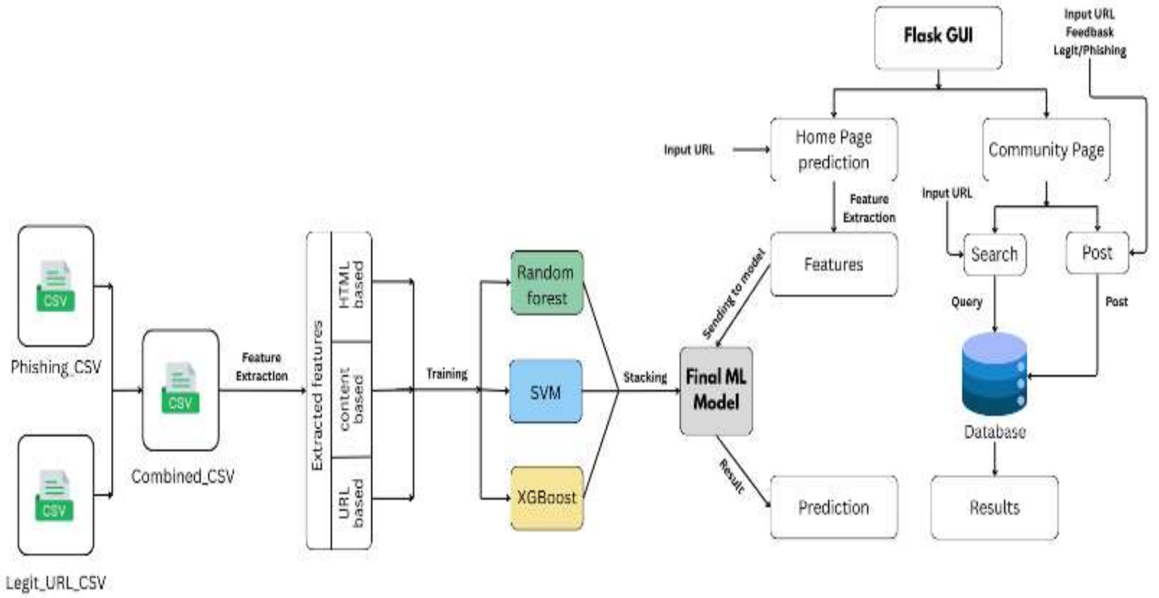

Fig 2.3.1 Full System Architecture

3 METHODOLOGY

The phishing detection system integrates machine learning with a community-driven feature for real-time threat detection. It employs Random Forest, SVM, and XGBoost for URL classification based on extracted features like domain attributes, lexical patterns, SSL validation, WHOIS data, and webpage structure. A stacking approach enhances accuracy by leveraging multiple classifiers. The dataset, sourced from PhishTank and the University of New Brunswick, undergoes feature engineering to refine data for better training. A Flask-based web interface allows users to enter URLs and receive real-time analysis with probability scores. Feature selection techniques optimize performance while preserving phishing indicators. Users can input a URL for instant classification via the trained model, displayed on a Flask GUI with alert mechanisms. A community-driven SQLAlchemy database lets users report, view, and search phishing links, promoting collaborative cybersecurity. The system is designed for lightweight deployment with potential integration into browser extensions or enterprise security tools.

Advantages include high detection accuracy through ensemble learning, reducing false positives compared to rule-based methods. Real-time detection enables instant phishing analysis via a user-friendly Flask interface. The model considers multiple feature types—URL, content, and HTML—improving adaptability to evolving attacks. A community-driven feature allows users to report phishing sites, enhancing shared security awareness. The scalable and modular design enables deployment as a browser extension or API, ensuring adaptability. SQLAlchemy efficiently manages phishing reports, creating a searchable repository for informed browsing. The system enhances security by reducing financial fraud, credential theft, and identity breaches, particularly in banking and e-commerce.

Limitations include vulnerability to zero-day attacks, reliance on predefined features, and possible false positives. The model depends on data sources like PhishTank and WHOIS, which may provide outdated information. Computational demands pose real-time processing challenges. Legal and ethical concerns, including compliance with privacy laws, must be addressed. The model assumes phishing sites share identifiable characteristics and that WHOIS/SSL data is always accessible. However, attackers may anonymize domain details, reducing feature extraction reliability. The system relies on a stable internet connection for real-time detection. Repeated false positives may lead to user distrust.

The system's novelty lies in its hybrid approach, combining machine learning with deep learning-based anomaly detection. Unlike rule-based filtering, it employs Random Forest, SVM, and XGBoost with handcrafted features like domain age, URL length, and WHOIS data. It continuously updates with real-time threat intelligence, enhancing adaptability. A unique aspect is the integration of a crowdsourced phishing database, ensuring responsiveness to emerging threats. While many solutions struggle with real-time detection, this model optimizes performance via feature selection and parallel processing. Anomaly detection enhances security by flagging suspicious behavior beyond known phishing patterns. The modular design enables seamless integration into browsers, email filters, or security tools. By incorporating user feedback and reinforcement learning, the system improves decision-making and resists adversarial attacks, ensuring long-term adaptability to cyber threats.

3.1 Modules:

A. Data Collection and Preprocessing (PhishTank & UNB Dataset) Module- The system collects phishing and legitimate URLs from PhishTank and UNB datasets, ensuring diverse data. Preprocessing involves cleaning, removing duplicates, and handling missing values. Feature engineering techniques refine raw data, converting URLs into structured inputs for machine learning models.

B. Feature Extraction Module (URL-based, Content-based, and Network-based Features)- Feature extraction is categorized into URL-based (domain length, special characters), content-based (HTML tags, JavaScript presence), and network-based (SSL, WHOIS data). Advanced statistical techniques help retain the most informative features, enhancing model efficiency.

C. Machine Learning Model Training (Random Forest, SVM, XGBoost, Autoencoder Neural Networks)- The system utilizes an ensemble approach with Random Forest, SVM, and XGBoost to classify URLs effectively. Additionally, Autoencoder Neural Networks identify anomalies by learning the distribution of legitimate websites and detecting deviations in phishing sites.

D. Model Stacking & Optimization (Ensemble Learning Approach)- Stacking combines predictions from different models, improving accuracy. A meta-classifier refines outputs by weighing individual model contributions, reducing false positives and negatives. Hyperparameter tuning enhances performance, adapting to new phishing tactics.

E. Webpage Classification & Prediction (Flask API, Real-time Detection on Home Page)- A Flask-based API enables real-time classification. Users input URLs, and the system returns phishing probability scores with alerts. The detection mechanism dynamically updates with new data, improving its phishing detection capabilities over time.

F. Deployment & Web Interface (Flask, HTML, JavaScript, CSS for Home Page & Detection System)- A user-friendly Flask-powered web interface integrates HTML, CSS, and JavaScript to present phishing detection results visually. Probability scores and color-coded indicators ensure easy interpretation, even for non-technical users.

G. Community Page Module (User-Submitted URL Reports, Search, Likes, & Discussion)- Users contribute by reporting phishing links, enhancing collective threat awareness. A searchable database stores community reports, helping users verify website credibility. Interactive features like discussions and likes facilitate knowledge sharing, reinforcing cybersecurity awareness. This modular design ensures a scalable, intelligent, and interactive phishing detection system, leveraging machine learning, community-driven insights, and real-time analysis to combat online threats effectively.

4 RESULTS AND DISCUSSION

A. Introduction- The proposed Phishing Website Detection System was trained on a dataset containing legitimate (label 0) and phishing (label 1) URLs, sourced from PhishTank and UNB cybersecurity datasets. Legitimate URLs included domains from e-commerce, government portals, and news sites, while phishing URLs covered fake banking, e-commerce, and credential-stealing sites. The dataset featured diverse URL structures, including shortened links, subdomains, and redirects, ensuring real-world applicability.

B. Dataset Splitting- The Training Set (70%): Used for model learning with Random Forest, SVM, and XGBoost. Features like URL length, subdomains, phishing-related keywords, SSL inconsistencies, and suspicious registrations played a key role. Validation Set (15%): Optimized hyperparameters like tree count (RF), kernel type (SVM), and learning rate (XGBoost) while preventing overfitting. Testing Set (15%): Evaluated model performance on unseen URLs using accuracy, precision, recall, and F1-score.

C. Performance Evaluation

Confusion Matrix:

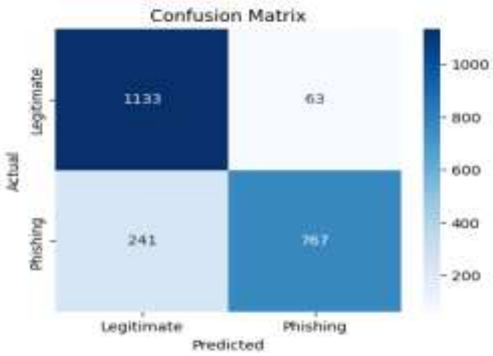

Fig 4.1 Confusion Matrix

TP (Phishing Detected): 767

TN (Legitimate Detected): 1133

FP (Legitimate Misclassified): 63

FN (Phishing Missed): 241

Classification Metrics:

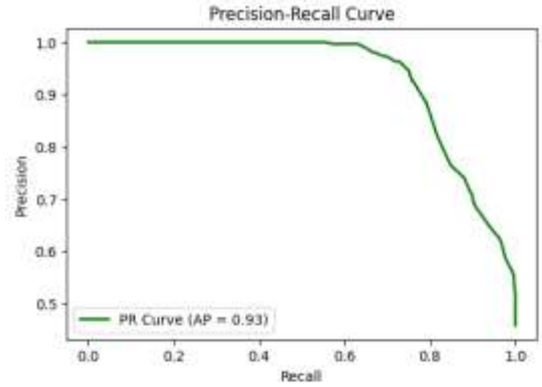

Fig -4.2 Precision-Recall

Accuracy: 86.2%

Precision: Legitimate: **82%**, Phishing: **92%**

Recall: Legitimate: **95%**, Phishing: **76%**

F1-Score: Legitimate: **88%**, Phishing: **83%**

ROC & Precision-Recall Curves:

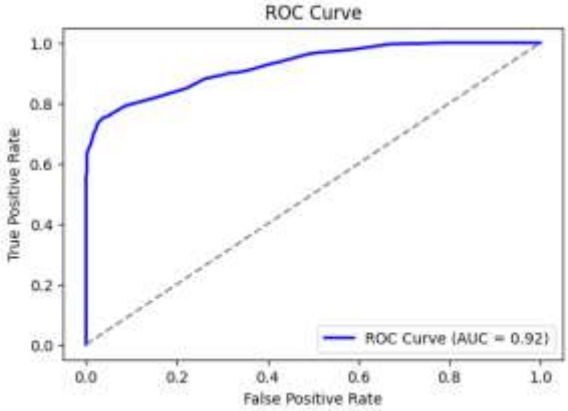

Fig 4.2 ROC Curves

- **AUC Score**: **0.92**, confirming strong class separation.
- **Average Precision (AP)**: **0.93**, demonstrating reliable phishing detection.

Feature Importance:

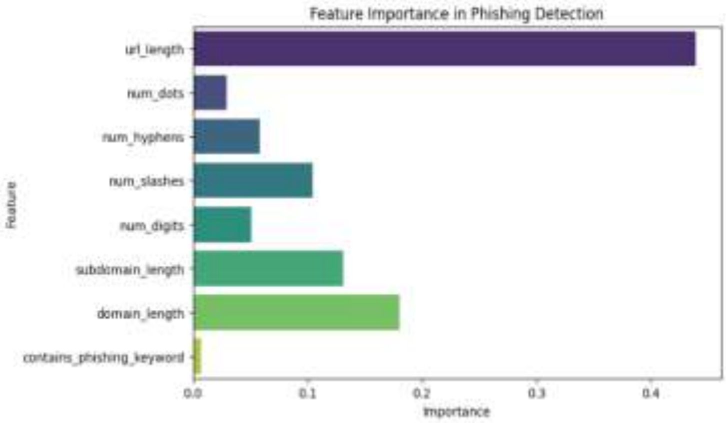

Fig 4.3 Feature Importance

1. **URL Length** – Longer URLs often indicate phishing attempts.
2. **Domain & Subdomain Count** – More subdomains suggest deception.
3. **Hyphens & Special Characters** – Used to mimic trusted domains.
4. **Phishing Keywords** – "Secure," "update," and "verify" frequently appear in phishing URLs.
5. **SSL Certificate & Domain Age** – Phishing sites often lack valid SSL and have short lifespans.

5 CONCLUSION

The phishing detection system demonstrates high accuracy (86.2%) and reliability in distinguishing phishing and legitimate websites using a stacked ensemble model (Random Forest, SVM, XGBoost). It effectively analyzes URL length, domain structure, SSL validity, and phishing-related keywords. With an AUC score of 0.92 and 92% precision, it minimizes false positives, making it practical for real-world use. However, the 76% recall for phishing sites suggests improvements are needed to reduce false negatives. Feature analysis confirms the significance of URL and domain attributes. Future work includes enhancing recall, integrating deep learning, real-time threat intelligence, and developing a browser extension or API.

6 FUTURE WORK

The phishing website detection system developed in this study has shown promising results, but several enhancements are needed to improve adaptability against evolving cyber threats. Integrating deep learning models like CNNs and RNNs could enable content-based analysis alongside URL-based features, while transformer architectures such as BERT or GPT could enhance linguistic pattern recognition in phishing URLs and webpages.

Real-time phishing detection can be achieved by deploying the model as a browser extension, web API, or cloud service, allowing users to verify URLs before access. Incorporating threat intelligence feeds from PhishTank, OpenPhish, and Google Safe Browsing could further improve detection capabilities.

To counter adversarial attacks, employing advanced adversarial training and combining heuristic-based methods with machine learning could enhance resilience. Expanding datasets with multilingual phishing sites and diverse attack strategies will improve generalization. Collaboration with cybersecurity organizations will ensure real-world validation, continuous updates, and sustained effectiveness, making phishing detection more robust and reliable.

7 REFERENCES

[1] Bhagwat, M. D., Patil, P. H., et al., 2021. A Methodical Overview on Detection, Identification, and Proactive Prevention of Phishing Websites.

[2] Singhal, S., Chawla, U., et al., 2020. Machine Learning & Concept Drift-Based Approach for Malicious Website Detection.

[3] Catal, C., Giray, G., et al., 2022. Applications of Deep Learning for Phishing Detection: A Systematic Literature Review.

[4] Rishikesh, M., et al., 2018. Phishing Detection Using Machine Learning Algorithms.

[5] Kiruthiga, R., Akhila, D., et al., 2019. Phishing Websites Detection Using Machine Learning.

[6] Kulkarni, A., et al., 2019. Phishing Websites Detection Using Machine Learning.

[7] Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Ul Haq, Q. E., Saleem, K., Faheem, M. H., 2023. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN.

[8] Salahdine, F., El Mrabet, Z., Kaabouch, N., 2022. Phishing Attacks Detection: A Machine Learning-Based Approach.

[9] Zhang, N., Yuan, Y., 2012. Phishing Detection Using Neural Networks.

[10] Zhang, N., Yuan, Y., 2012. Phishing Detection Using Neural Network AD Mentor.