International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

URL Feature Analysis for Effective Phishing Detection using Machine Learning

Dr. S. Subashini[1], A. MohammedNavas[2], T. S. P. Ksheerabdinath[3], M. H. Ksheerabdhinath[4], T. S. Muneeshwara Venkatesh[5]

[12345]Department of Computer Science and Engineering, KLN College of Engineering, Tamil Nadu, India.

Abstract - In the digital age, phishing websites pose a significant threat to online security by mimicking trusted platforms to steal sensitive user information. The Phishing Website Detection system leverages Machine Learning (ML) to automate the classification of URLs as either phishing or benign. By analyzing a dataset of 10,000 URLs, consisting of 5,000 phishing and 5,000 legitimate links, the system extracts 17 key features related to address bar, domain, and HTML/JavaScript indicators. Among the algorithms tested, including Decision Tree, Random Forest, and SVM, the XGBoost classifier achieved the highest accuracy of 86.4%. Developed with Flask as the backend, the web-based application ensures real-time detection, providing users with phishing alerts while maintaining memory efficiency. The tool contributes to enhancing web security, offering reliable protection in various digital environments.

***Keywords*:** Machine Learning, Phishing Detection, URL Feature Extraction, XGBoost Classifier.

1. INTRODUCTION

In today's digital era, phishing attacks have become one of the most common threats to cyber security, targeting users through deceptive websites designed to steal sensitive data. Manually identifying and filtering such malicious sites is inefficient and error-prone, especially with the increasing volume of online activity. To tackle this issue, a machine learning-based phishing detection system has been developed that analyses the structure and behaviour of URLs to determine their legitimacy. The system uses a balanced dataset of 10,000 URLs—5,000 phishing URLs from PhishTank and 5,000 legitimate ones—and extracts 17 key features categorized into address bar, domain-based, and HTML/JavaScript-based features. These features offer a comprehensive understanding of each URL, enabling more accurate classification.

Multiple machine learning algorithms, including Decision Tree, Random Forest, SVM, and XGBoost, are trained and evaluated on the dataset using an 80-20 train-test split. Among them, the XGBoost classifier achieves the highest accuracy of 86.4%, minimizing both false positives and false negatives. The system is optimized for real-time detection, making it suitable for integration into browsers or cyber security tools. Its lightweight design ensures fast and memory-efficient operation, capable of handling both real-time and batch URL analysis. By focusing solely on machine learning techniques, the system offers a reliable and scalable solution for early phishing detection, enhancing user safety in a rapidly evolving digital environment.

2. METHADOLOGY

2.1 Modules:

2.1.1 Data Collection

The Data Collection module is responsible for gathering URLs from credible sources such as PhishTank, Kaggle, and publicly available cyber security datasets. These datasets contain both phishing and benign URLs, exposing the model to a variety of phishing patterns. After gathering the data, duplicates and malformed URLs are removed to ensure data quality. Additionally, URLs are normalized by converting them to lowercase and handling special characters for consistency.

2.1.2 Data Preprocessing

Data preprocessing involves cleaning and transforming the collected data into a structured format suitable for feature extraction and model training. The process includes:

Text Cleaning: Removing unnecessary elements such as duplicates and malformed URLs.

Normalization: Converting URLs to lowercase and handling special characters for uniformity.

Feature Extraction: Extracting lexical and domain-based features (e.g., URL length, subdomains, WHOIS information) to enable the model to distinguish between phishing and benign URLs.

2.1.3 Feature Extraction

The Feature Extraction module is crucial for converting raw URL data into numerical features. The module extracts:

Lexical Features: Includes URL length, subdomain count, presence of special characters, and URL entropy.

Content-Based Features: Identifies suspicious HTML/JavaScript elements like <iframe>, <script>, and event handlers used in phishing attacks.

Domain-Based Features: Retrieves WHOIS data, such as domain registration date and registrar information, to detect newly registered domains often used in phishing.

These features enable the machine learning model to better classify URLs based on patterns indicative of phishing attempts.

2.1.4 Model Training and Classification

This module focuses on training machine learning models to classify URLs. The process includes:

Model Selection: Multiple algorithms such as Random Forest, Decision Tree, and XGBoost are evaluated for their effectiveness in classifying phishing URLs.

Training: The dataset is split into training and validation sets, and techniques like cross-validation are applied to prevent overfitting.

Optimization: The model is trained using categorical cross-entropy to minimize prediction errors.

The goal is to build an accurate and robust model capable of classifying URLs as phishing or benign based on the extracted features.

2.1.5 Evaluation and Optimization

Once the model is trained, it is evaluated using metrics such as accuracy, precision, recall, and F1-score to assess its performance. The system also uses optimization techniques like:

Feature Selection: Identifying the most relevant features.

Dimensionality Reduction: Simplifying the model while retaining important information.

Ensemble Learning: Combining multiple models for improved accuracy.

The use of K-fold cross-validation ensures the model generalizes well to new data, enhancing its real-world application for phishing detection.

2.1.6 Real-Time Detection

The Real-Time Detection module allows users to input URLs for immediate classification. The process includes:

Input Handling: Users submit URLs via a web interface, and the system processes these inputs in real-time.

Classification: The preprocessed URL is classified by the trained model as phishing or benign.

This module ensures fast and reliable predictions, even under heavy user demand, by using lightweight models and efficient processing pipelines.

2.1.7 User Interface (UI)

The User Interface (UI) module provides a user-friendly platform for interacting with the phishing detection system. It features:

URL Submission: A simple form for users to input URLs.

Result Display: The classification result (Phishing or Benign) is displayed with clear visual cues.

Security Recommendations: The system offers tips on safe browsing practices when a phishing URL is detected.

This module ensures the system is accessible and informative for users, enhancing the overall user experience and promoting secure online practices.

2.1.8 Data Flow Diagram

The data flow of the Phishing URL Detection System is shown in Fig. 1. It covers user input, URL preprocessing, feature extraction, model training, and real-time classification, ensuring efficient and accurate phishing detection with instant alerts.

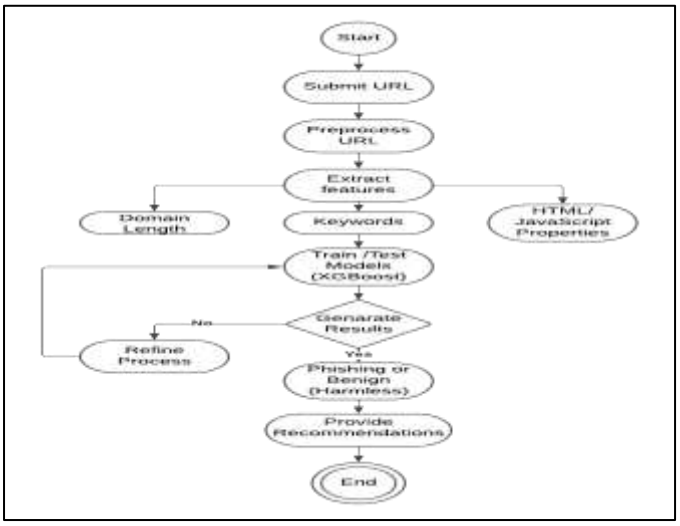

Fig -1 Data Flow Diagram of the Phishing Detection System

2.1.9 Architecture Diagram

The Phishing URL Detection System architecture is depicted in Fig. 2. It showcases the integration of data collection, feature extraction, machine learning models, real-time detection, and the user interface. This design ensures seamless processing, accurate classification, and fast user interactions.

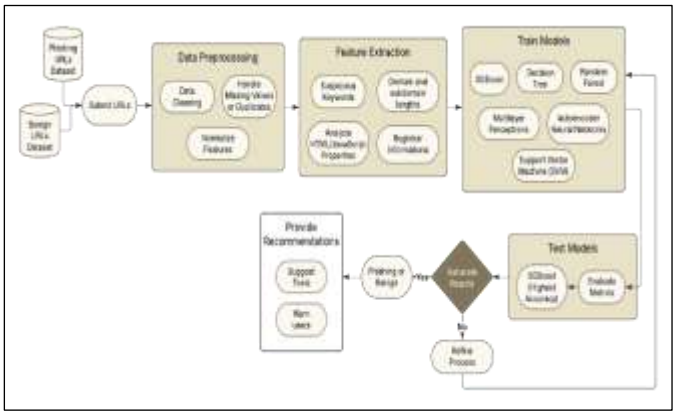

Fig -2 Architecture Diagram of the Phishing Detection System

2.2. EXPERIMENTAL FINDINGS

2.2.1 System Implementation

The Phishing URL Detection System is implemented using Python with Flask as the backend and Bootstrap for the frontend design. The core classification is performed using the XGBoost algorithm, chosen for its high accuracy in detecting phishing URLs. The system processes URLs, extracts features, and classifies them in real-time.

Upon receiving a URL input from users, the system applies preprocessing steps like data cleaning and feature extraction. After classification, results are displayed immediately, notifying users whether the URL is phishing or benign. The system is lightweight, responsive, and provides security alerts to ensure safe browsing.

2.2.1 Output Screens

The web application displays:

- Feature Correlation Matrix
- Phishing vs Legitimate Websites
- URL Input and Phishing Website
- URL Input and Legitimate Website

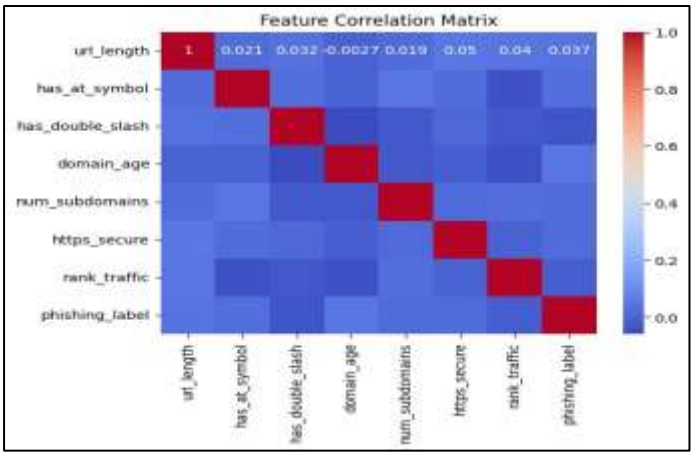

Fig -3 Feature Correlation Matrix of the Phishing Detection System

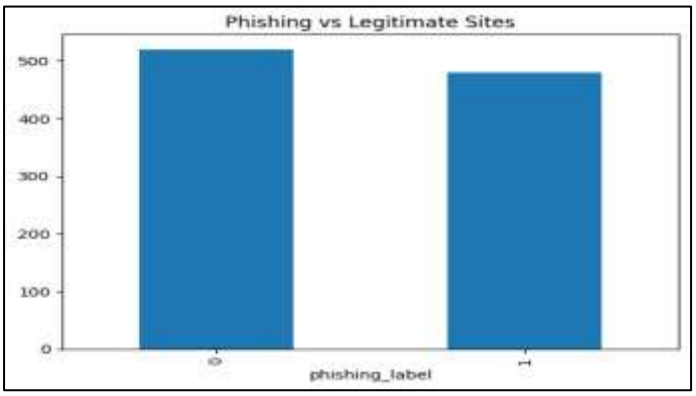

Fig -4 Phishing vs Legitimate Websites of the Phishing Detection System

Fig -5 URL Input and Phishing Website of the Phishing Detection System

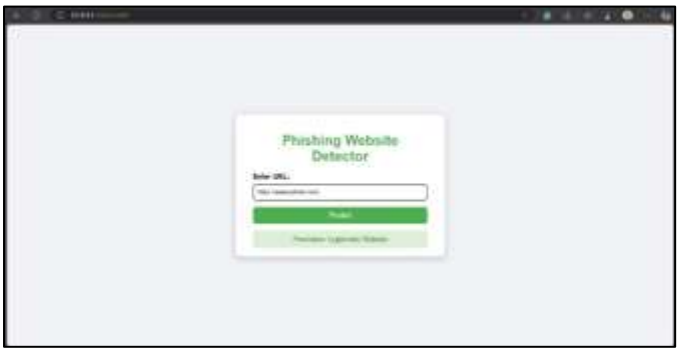

Fig -6 URL Input and Legitimate Website of the Phishing Detection System

2.3 APPLICATIONS AND LIMITATIONS

2.3.1 Applications

Real-Time Threat Detection: The system allows users to instantly verify whether a URL is safe or phishing, enabling timely decision-making and preventing data breaches. It is especially useful for individuals and organizations receiving suspicious links via email or social media.

Email Security: It can be integrated into email gateways to automatically scan and flag phishing links in real-time, enhancing cyber security in professional and academic environments.

Educational Awareness: The system can be used in cyber security training programs to demonstrate phishing patterns and URL-based attacks, improving awareness and digital literacy among students and employees.

Browser Extension Integration: With further development, the system can power browser plugins that warn users when they encounter potentially malicious websites while browsing the internet.

Corporate IT Solutions: Organizations can embed the system within their internal networks to monitor and filter URLs, reducing the risk of employees accessing harmful websites and improving overall IT security.

Cybercrime Investigation: Cyber security experts can use the system to analyse large datasets of URLs collected during investigations, enabling efficient detection of malicious patterns and emerging phishing tactics.

2.3.2 Limitations

Accuracy Dependent on Dataset: The detection results are influenced by the quality and scope of the training dataset. Limited or outdated data can reduce the model's ability to detect new phishing techniques.

False Positives/Negatives: The system might incorrectly label legitimate URLs as phishing or fail to detect a phishing URL. Such misclassifications can affect user trust and security.

URL-Based Detection Only: The system analyses only the URL structure and features. It does not inspect the actual webpage content or behaviour, which could contain hidden phishing elements.

Scalability Issues: In high-traffic scenarios, especially during multiple simultaneous submissions, system performance may degrade if server resources are limited.

Bypass through Obfuscation: Attackers may use URL shortening, encoding, or redirection to bypass detection. Without advanced parsing, the system may not flag such URLs correctly.

Lack of Browser Integration: The system functions as a standalone web tool and does not offer real-time browser extension or email client integration for direct in-use detection.

User Input Dependency: The system depends on users to manually submit URLs. Mistyped or incomplete URLs cannot be analysed, limiting effectiveness without proper input validation.

2.4 Conclusion and Future Directions

2.4.1 Conclusion

The integration of advanced machine learning models, particularly XGBoost, enables the system to effectively classify phishing and benign URLs with an accuracy of 86.4%. By leveraging 17 URL-based features, including domain attributes, JavaScript elements, and suspicious keywords, the system ensures precise detection of malicious websites. This approach strengthens cyber security by identifying deceptive patterns in real time, allowing users to navigate the web with greater confidence.

The system dynamically processes incoming URLs, providing instant threat detection and immediate alerts to users. By continuously optimizing feature selection, it minimizes memory usage and processing time, ensuring smooth and efficient phishing detection

without compromising performance. Its adaptability makes it suitable for integration into web browsers, enterprise security tools, and online fraud prevention systems.

Beyond just detecting threats, the system enhances overall cyber security awareness by providing users with insights into why a URL is flagged as phishing. This educates individuals and organizations about common phishing techniques, helping them recognize suspicious patterns even without automated detection. By combining real-time classification with an informative approach, the system not only protects users but also empowers them with knowledge to prevent potential cyber threats proactively.

2.4.1 Future Directions

Integration with Web Browsers and Extensions: A potential future enhancement is to develop browser extensions for Chrome, Firefox, and Edge that embed the detection model directly into users' web experiences. This would provide real-time alerts and seamless protection while users browse the internet.

Continuous Learning Models: Implementing an online learning mechanism where the system updates itself based on new phishing techniques can greatly improve its adaptability. Incorporating feedback loops and updated datasets will ensure the model evolves with emerging threats.

Expanded Feature Set: Future versions could explore additional behavioural and contextual features such as webpage layout patterns, SSL certificate validation, and user interaction metrics to further improve classification accuracy and reduce false positives.

Mobile Platform Support: Extending the system to work on Android and iOS platforms would enhance mobile security. Integrating it with mobile browsers or as a background security app could help detect phishing attempts in SMS, email, and browser-based links.

Threat Intelligence Integration: Incorporating threat feeds from global cyber security databases can provide enriched context and increase the reliability of URL classification. This integration would allow the system to recognize newly identified phishing campaigns and blacklisted domains in real time.

Multilingual and Regional Adaptation: Supporting localized versions of phishing detection by adapting the system to recognize regional phishing patterns, domains, and languages can improve accessibility and make it more effective for global users.

3. CONCLUSIONS

The online version of the volume will be available in LNCS Online. Members of institutes subscribing to the Lecture Notes in Computer Science series have access to all the pdfs of all the online publications. Non-subscribers can only read as far as the abstracts. If they try to go beyond this point, they are automatically asked, whether they would like to order the pdf, and are given instructions as to how to do so.

REFERENCES

1. Abdelhamid N., Ayesh A., Thabtah F. – "Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features", Journal of Information Security and Applications, Vol. 54, pp. 102611, 2020.
2. Akinyelu A.A., Adewumi A.O. – "Classification of Phishing Email Using Random Forest Machine Learning Technique", Advances in Big Data and Cloud Computing, Vol. 1253, pp. 1–5, 2019.
3. Alzahrani A., Mahmmod B. – "Phishing Website Detection Using Machine Learning Techniques", Procedia Computer Science, Vol. 170, pp. 374–380, 2019.
4. Basit A., Zafar M., Liu X. – "PhishHawk: Building an Explainable Phishing Detection Framework Using Deep Learning and SHAP", IEEE Access, Vol. 9, pp. 160229–160243, 2021.
5. Bhandari A., Kumari R., Tripathi S. – "A Hybrid CNN-RF Model for Real-Time Phishing URL Detection Using Embedded Features", Journal of Cybersecurity and Privacy, Vol. 4, No. 1, pp. 73–89, 2024.
6. Jain A., Gupta B. – "Phishing Detection Using Machine Learning Techniques", International Journal of Engineering and Technology, Vol. 9, No. 2, pp. 95–102, 2020.
7. Khanna A., Saini P. – "Intelligent Phishing URL Detection Using Stacked LSTM with Statistical and Lexical Features", Proceedings of the International Conference on Machine Intelligence and Signal Processing (MISP), pp. 215–220, 2022.
8. Kulkarni A., Manekar V., Mane P. – "URL-Based Phishing Website Detection Using Machine

Learning", Proceedings of the International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1–6, 2019.

9. Patel H., Mehta H., Bhavsar H. – "DeepURL: A Deep Learning Approach for Phishing URL Detection", Procedia Computer Science, Vol. 185, pp. 207–214, 2021.
10. Rao R.S., Pais A.R. – "PhishTank: An Efficient Learning-Based URL Phishing Detection with Natural Language Processing Techniques", International Journal of Network Security, Vol. 22, No. 5, pp. 933–940, 2020.
11. Sharma A., Goyal M. – "Phishing Detection Using Neural Networks with Lexical and Domain Features", Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS), pp. 359–364, 2023.
12. Sahoo D., Liu C., Hoi S.C.H. – "Malicious URL Detection Using Machine Learning: A Survey", ACM Computing Surveys, Vol. 53, No. 6, pp. 1–51, 2020.
13. Verma C., Saxena P. – "URLNet++: A Lightweight CNN-LSTM Architecture for Phishing URL Classification", Proceedings of the International Conference on Data Science and Intelligent Applications (ICDSIA), pp. 120–126, 2023.
14. Wu C., Liu Y., Zhang Y. – "Lightweight Phishing Detection for Edge Devices Using URL-Based Feature Engineering and Ensemble Models", Journal of Information Security and Applications, Vol. 58, pp. 102771, 2021.
15. Zhang X., Zhang J., Xue Y. – "An Improved URL Phishing Detection Approach Using Attention-Based BiLSTM and Domain Features", Proceedings of IEEE GLOBECOM, pp. 1–6, 2022.