International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 2, 2026

Machine Learning – Based Detection Of Phishing Websites

Dr. P. Rajendra Prasad[1], P. Akhila[2], P. Joshna[3], P. Vijaya Sowmya[4], P. Hima Bindhu[5]

[1]Associate Professor; Department Of Computer Science And Engineering Vignan's Institute Of Management And Technology For Women Hyderabad India

[2,3,4,5]B.Tech Students; Department Of Computer Science And Engineering Vignan's Institute Of Management And Technology For Women Hyderabad India

Mail Id; pallaakhilareddy8@gmail.com[2], vijayasowmyapushadapu@gmail.com [4]

Abstract

The widespread adoption of the internet has significantly increased exposure to cyber threats, with phishing attacks emerging as one of the most prevalent forms of online fraud. Phishing websites are designed to deceive users into disclosing confidential information such as login credentials, financial details, and personal data. Detecting such malicious websites at an early stage is critical for improving cybersecurity and protecting users.

This study proposes a machine learning–driven framework for identifying phishing websites through the analysis of URL characteristics. A dataset containing more than 101,000 URLs was utilized to train and evaluate the proposed system. From each URL, sixteen discriminative features were extracted, including URL length, HTTPS usage, entropy score, domain length, top-level domain (TLD) popularity, presence of IP address, digit ratio, and other structural attributes.Three supervised machine learning algorithms—Decision Tree, Random Forest, and XGBoost—were implemented and compared to determine the most effective model for phishing detection. Experimental results indicate that the Random Forest classifier achieved the best performance, reaching an accuracy of ***99.98%****, highlighting the effectiveness of feature-based URL analysis in identifying phishing threats.*

To demonstrate the practical applicability of the proposed model, a web-based interface was developed using the Flask framework. The application allows users to submit URLs and receive real-time predictions regarding potential phishing risks through a user-friendly interface. The results of this research suggest that machine learning techniques combined with URL feature engineering can serve as an efficient solution for automated phishing detection and online threat mitigation.

Keywords

Phishing Detection, Machine Learning, Cybersecurity, URL Feature Extraction, Random Forest, XGBoost, Decision Tree, Web Security, Flask Application

Introduction

The rapid expansion of internet technologies has significantly changed the way individuals, businesses, and government institutions communicate, conduct transactions, and manage information. Services such as online banking, electronic commerce, social networking platforms, and cloud-based applications have become integral parts of everyday life. As a result, a large volume of sensitive information—including personal data, authentication credentials, and financial records—is transmitted and stored online. While these advancements have improved accessibility and efficiency, they have also created opportunities for cybercriminals to exploit vulnerabilities in digital systems and human behavior.Among the various cyber threats, phishing has emerged as one of the most prevalent and damaging forms of online fraud. Phishing attacks involve the creation of deceptive websites or messages that impersonate legitimate organizations with the objective of tricking users into disclosing confidential information such as usernames, passwords, credit card numbers, or other personal data. Attackers typically distribute phishing links through emails, social media platforms, or messaging applications. Once users interact with these malicious links, they may unknowingly provide sensitive information that can later be used for identity theft, financial fraud, or unauthorized system access.Recent cybersecurity reports highlight the increasing scale of phishing activities worldwide. According to the Internet Crime Report released by the Federal Bureau of Investigation's Internet Crime Complaint Center (IC3), cybercrime losses in the United States exceeded 10 billion dollars in 2022, with phishing being the most frequently reported attack category. Similarly, the Anti-Phishing Working Group (APWG) documented more than 1.6 million phishing incidents during the first quarter of 2023 alone, representing the highest number recorded in a single quarter. These alarming statistics demonstrate that phishing remains a persistent and rapidly evolving cybersecurity challenge that affects individuals, organizations, and governments globally.One of the primary reasons phishing attacks are difficult to prevent is the increasing sophistication of modern phishing websites. Attackers are now capable of designing fraudulent websites that closely resemble legitimate platforms in terms of layout, branding, and functionality. The availability of low-cost domain registration services, free SSL certificates, and URL shortening tools allows cybercriminals to quickly generate convincing phishing links. In many cases, these malicious websites appear authentic even to experienced users, making manual identification extremely difficult. As a result, traditional cybersecurity measures such as user awareness programs, email filtering mechanisms,

and blacklist-based detection methods are often insufficient to fully address the problem.

Existing System

Current phishing detection approaches largely depend on conventional security mechanisms and manual verification processes. Web browsers, antivirus software, and security gateways typically rely on blacklist databases that contain previously reported malicious URLs. When a user attempts to access a website, the system checks whether the URL is listed in the database and generates a warning if a match is found. While this approach can effectively block known phishing sites, it has a significant limitation: newly created phishing websites may remain undetected until they are reported and added to the blacklist.In addition to blacklist mechanisms, some detection systems employ rule-based techniques that examine predefined patterns within URLs, such as unusual domain names, excessive use of special characters, or suspicious link structures. Although these rule-based methods can detect certain phishing attempts, they often fail to identify more sophisticated attacks that are carefully designed to resemble legitimate websites. Consequently, these traditional approaches lack the adaptability required to address the constantly evolving nature of phishing threats.

Proposed System

To overcome the limitations of traditional detection methods, this research proposes an intelligent phishing detection system based on machine learning techniques. The proposed approach focuses on analyzing structural and behavioral characteristics of URLs in order to automatically classify websites as legitimate or malicious. By extracting meaningful features from URLs and training machine learning models on large datasets, the system can learn patterns associated with phishing activities and identify suspicious links with high accuracy.The proposed framework integrates machine learning algorithms with a web-based application to enable real-time phishing detection. The system utilizes a dataset of URLs from which multiple features are extracted and analyzed using classification algorithms such as Decision Tree, Random Forest, and XGBoost. Among these models, Random Forest demonstrates superior performance in terms of accuracy and reliability.To enhance usability, a web application is developed using the Flask framework, providing users with a simple interface to submit URLs for analysis. The platform processes the input URL, extracts relevant features, and uses the trained model to determine whether the website is safe or potentially malicious. By combining machine learning capabilities with an accessible web interface, the proposed system aims to provide an efficient and practical solution for detecting phishing websites and improving overall online security.

LITERATURE SURVEY

The problem of phishing website detection has attracted considerable attention in the cybersecurity research community. Numerous studies have explored machine learning and deep learning techniques to identify malicious websites by analyzing URL patterns, webpage characteristics, and behavioral attributes.Ma et al. proposed an early machine learning approach for detecting malicious websites by analyzing lexical characteristics of URLs. Their work emphasized the limitations of blacklist-based systems, which are only capable of detecting previously identified phishing websites. Instead of relying on blacklists, the authors extracted several URL-based attributes such as token patterns, URL length, and suspicious substrings. These features were used to train supervised learning models capable of distinguishing malicious websites from legitimate ones. The results demonstrated that machine learning-based detection systems can successfully identify previously unseen phishing websites with high accuracy, highlighting their effectiveness over traditional rule-based filtering techniques.Sahingoz et al. presented a lightweight phishing detection framework that relies exclusively on URL-based features. The primary objective of their research was to design a fast and computationally efficient system capable of real-time detection without requiring webpage content analysis. The authors extracted lexical features from URLs and evaluated multiple machine learning algorithms including Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbors (kNN). Experimental results revealed that the Random Forest classifier achieved the best classification performance among the evaluated models. Their findings confirmed that URL-based analysis can provide an efficient and practical solution for real-time phishing detection systems.Feng et al. introduced a phishing detection technique known as Web2Vec, which utilizes multi-view representation learning to improve classification accuracy. Unlike traditional approaches that rely on a single type of feature, the proposed system combines textual information, URL structure, and visual characteristics of webpages. Deep learning models are employed to generate vector representations that capture complex relationships between these features. The experimental evaluation demonstrated that integrating multiple feature views significantly improves detection accuracy compared to conventional machine learning approaches that rely solely on lexical URL features.Huang et al. proposed a deep learning-based phishing detection system using a hybrid architecture that combines Convolutional Neural Networks (CNN) with Hierarchical Recurrent Neural Networks (RNN). This approach focuses on learning character-level and word-level representations directly from URLs,

eliminating the need for manual feature extraction. The motivation behind this work was to address sophisticated phishing techniques that attempt to evade traditional detection mechanisms through URL obfuscation. Experimental results showed that the hybrid CNN-RNN model achieved high accuracy in distinguishing phishing URLs from legitimate ones, demonstrating the effectiveness of deep learning in cybersecurity applications.Large technology organizations have also contributed significantly to phishing detection research. Google developed systems such as PhishNet and the Google Safe Browsing API to protect users from malicious websites. These systems rely on continuously updated blacklists and reputation databases to identify known phishing domains. When a user attempts to access a suspicious website, the browser compares the URL against the database and warns the user if the website is identified as malicious. While this approach is highly effective in blocking previously reported phishing websites, its effectiveness is limited when dealing with newly created or zero-day phishing attacks that have not yet been added to the blacklist database. This limitation has motivated researchers to explore machine learning-based detection systems capable of identifying unknown phishing websites.

SYSTEM ANALYSIS

System analysis plays a crucial role in identifying the requirements, feasibility, and functional scope of the proposed phishing detection system. This stage focuses on evaluating existing solutions, defining system objectives, and ensuring that the proposed system can be successfully implemented using available resources.

Purpose

The primary objective of the proposed system is to design an intelligent and scalable phishing detection platform that overcomes the limitations of traditional detection methods. Conventional approaches often rely on manual verification or static rule-based filtering mechanisms that are unable to detect newly generated phishing websites effectively. The proposed system utilizes machine learning algorithms to automatically analyze URL characteristics and detect phishing websites in real time. By introducing automated analysis and AI-driven decision making, the system aims to improve detection accuracy, reduce manual intervention, and enhance the overall security experience for users.

Scope

The scope of the proposed system includes the design, development, and deployment of a web-based application capable of detecting phishing URLs using machine learning techniques. The system consists of several functional modules such as user authentication, data processing, machine learning-based analysis, and a dashboard interface for visualizing results. The platform is intended primarily for educational institutions, startups, and small organizations that require affordable cybersecurity tools. Advanced enterprise-scale features and specialized hardware integration are outside the scope of this project.

Feasibility Study

Economic Feasibility

The economic feasibility of the proposed system is favorable due to the use of open-source technologies and cloud-based infrastructure. The system can be developed using freely available programming frameworks such as Python, Flask, and open-source machine learning libraries. The estimated cost for deployment, including domain registration and cloud hosting services, remains relatively low. This cost-effective approach enables small organizations and educational institutions to adopt the system without significant financial investment.

Technical Feasibility

From a technical perspective, the system can be implemented using widely available web development and machine learning technologies. The backend is developed using Python and its associated machine learning libraries, while the frontend interface is created using modern web technologies such as HTML, CSS, and JavaScript. Cloud platforms can be used for deployment to ensure scalability and accessibility. Since the required technologies are well-established and supported by extensive documentation, the system is technically feasible to implement.

Social Feasibility

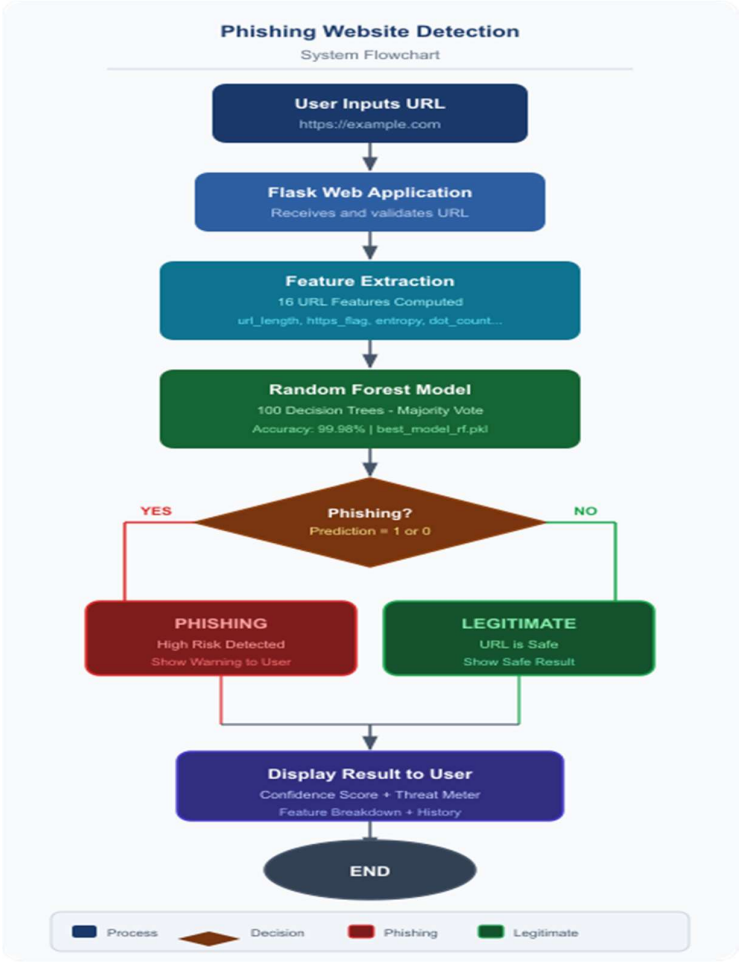
The proposed system contributes positively to cybersecurity awareness by providing users with a tool for identifying potentially malicious websites. It can be particularly beneficial for students, researchers, and professionals who frequently interact with online platforms. By promoting responsible and secure internet usage, the system supports broader initiatives related to digital safety and ethical use of artificial intelligence technologies.

Requirement Analysis

The system requires both functional and non-functional capabilities to operate effectively. Functional requirements include user authentication, real-time URL analysis, result visualization, and notification mechanisms for potential threats. Non-functional requirements focus on performance, reliability, and security. The system is designed to provide fast response times, high availability, and secure data handling practices while maintaining compatibility across different web browsers and mobile devices.

SYSTEM DESIGN

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



The Image Representing the flow of the System

The system design phase focuses on defining the architectural structure and operational workflow of the phishing detection system. The proposed design follows a modular and layered architecture commonly used in web-based artificial intelligence applications.

System Architecture

The architecture of the proposed system follows a three-tier design consisting of the presentation layer, business logic layer, and data layer. This architecture ensures clear separation between user interaction, processing logic, and data management.The presentation layer represents the user interface through which users interact with the system. It is implemented using web technologies such as HTML, CSS, and JavaScript to provide a responsive and interactive interface. The interface allows users to submit URLs for analysis and view the results generated by the machine learning model.The business logic layer performs the core processing tasks of the system. This layer is responsible for handling user requests, extracting features from submitted URLs, and performing classification using trained machine learning models. The backend logic is implemented using Python and the Flask framework, which enables efficient communication between the user interface and the machine learning model.The data layer manages data storage and retrieval processes. It stores trained machine learning models, feature configurations, and system logs. Databases and file-based storage mechanisms can be used to manage system data efficiently while maintaining security and reliability.

System Design Description

The system design adopts the Model-View-Controller (MVC) architecture to ensure modular development and maintainability. The model component represents the machine learning algorithms responsible for phishing detection. The view component corresponds to the web-based user interface that displays results to users. The controller component manages communication between the user interface and the machine learning model.Several functional modules are implemented within the system. The user management module handles authentication and role-based access control. The analysis engine module performs feature extraction, model inference, and classification of URLs. The administrative module provides monitoring capabilities for system performance and allows administrators to update machine learning models when new training data

becomes available.The system processes user requests through an API-based communication mechanism. When a user submits a URL, the request is validated and forwarded to the backend service for processing. The feature extraction module converts the URL into a numerical feature vector compatible with the trained model. The machine learning model then predicts whether the URL is legitimate or malicious. The result is returned to the user interface and displayed immediately.

IMPLEMENTATION AND RESULTS

Dataset Description

The dataset used for training and evaluation consists of more than 101,000 URLs collected from publicly available repositories containing both phishing and legitimate websites. After removing incomplete records, the final dataset contains 101,218 URLs with sixteen extracted features and a binary class label representing whether the URL is legitimate or phishing. Approximately sixty-three percent of the dataset consists of legitimate URLs, while the remaining portion represents phishing URLs. The dataset is divided into training and testing sets using an 80–20 split to ensure unbiased evaluation of model performance.

Data Preprocessing

Before training the machine learning models, several preprocessing steps were performed. Missing values were removed to ensure data consistency. The dataset labels were verified and adjusted to follow the standard classification format, where zero represents legitimate websites and one represents phishing websites. The URL column was excluded from the feature matrix and retained only for reference purposes. Stratified sampling was used during the train-test split to maintain the original class distribution in both datasets.

Machine Learning Algorithms

Three machine learning algorithms were implemented and evaluated for phishing detection: Decision Tree, Random Forest, and XGBoost. The Decision Tree algorithm constructs a hierarchical structure of decision rules that classify URLs based on feature thresholds. Random Forest improves classification performance by combining multiple decision trees and aggregating their predictions through majority voting. XGBoost utilizes gradient boosting techniques to build a strong ensemble model that minimizes classification errors iteratively.Among these models, Random Forest demonstrated the highest accuracy during experimental evaluation. Its ability to combine multiple decision trees helps reduce overfitting and improve generalization performance.

Web Application Implementation

To demonstrate the practical applicability of the proposed detection system, the trained machine learning model was integrated into a web application using the Flask microframework. The application allows users to submit URLs for real-time phishing analysis. When a URL is submitted, the backend system extracts relevant features, applies the trained machine learning model, and returns the classification result along with a confidence score.The frontend interface was developed using HTML, CSS, and JavaScript with the Tailwind CSS framework for styling. The application provides an interactive interface where users can perform single URL scans or batch scans. The results are displayed dynamically without requiring page reloads, providing a seamless user experience.The backend application loads the trained model during system initialization and performs predictions using serialized model files. RESTful API endpoints handle communication between the frontend interface and backend processing modules, enabling efficient real-time analysis of user-submitted URLs.

Results and Discussion

Experimental evaluation indicates that the Random Forest classifier achieved the highest accuracy among the tested algorithms, reaching approximately 99.98 percent accuracy on the test dataset. This result demonstrates that URL-based feature engineering combined with ensemble machine learning models can effectively identify phishing websites. The web application implementation further validates the practical usability of the system by enabling real-time phishing detection through an accessible interface.

SYSTEM TESTING

System testing is an essential stage in the development process that ensures the entire phishing detection system operates correctly when all components are integrated. In this project, system testing was performed to verify the correct interaction between the URL input module, feature extraction process, machine learning model, and the Flask-based web interface responsible for presenting the prediction results. The main objective of this testing phase was to confirm that the application performs accurate phishing detection in real time while maintaining system stability and usability.The testing process focused on validating several operational aspects of the system. First, it ensured that the application correctly accepts user-submitted URLs and processes them without runtime errors. The testing phase also verified that the feature extraction module accurately converts each URL into the predefined set of structural and behavioral attributes used by the machine learning model. Another important aspect of testing involved confirming that the trained classifier correctly interprets the extracted features and returns the appropriate prediction label indicating whether the URL is legitimate or associated with phishing activity. Finally, the testing process evaluated whether the Flask web interface properly displays the classification result, confidence score, and

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

related information in a clear and understandable manner for end users.Different testing approaches were used to evaluate the performance and reliability of the system. Functional testing was conducted to ensure that each component of the application performs its intended task correctly, including URL submission, feature extraction, model prediction, and result presentation. Integration testing was performed to verify that all modules interact seamlessly, particularly the communication between the feature extraction pipeline, machine learning model, and the web interface. Performance testing evaluated the response time of the system to ensure that the application is capable of processing URL scans in real time with minimal delay. Security testing examined the system's ability to handle invalid inputs, malicious strings, or unexpected characters without causing failures. In addition, usability testing was carried out to confirm that the user interface is simple, intuitive, and accessible for individuals with minimal technical knowledge.Several practical test scenarios were executed during system validation. The system was tested using valid URLs to confirm correct classification behavior. Known phishing URLs were also submitted to verify whether the detection model correctly identifies them as malicious. Additional tests included submitting empty inputs or incorrectly formatted URLs to ensure that appropriate error messages were generated. Stress testing was conducted by scanning multiple URLs repeatedly to observe the stability and responsiveness of the application. Another important verification step ensured that the trained machine learning model loads successfully during the application startup process.The results obtained from system testing indicate that the phishing detection system operates reliably and meets its design objectives. The application is capable of classifying URLs with high accuracy while maintaining fast response times for real-time scanning. Invalid inputs are handled gracefully, preventing unexpected crashes or system instability. Furthermore, the user interface effectively presents prediction results in a clear and informative manner. Overall, the system testing phase confirms that the developed phishing detection solution is stable, efficient, and suitable for practical deployment.

Screenshots

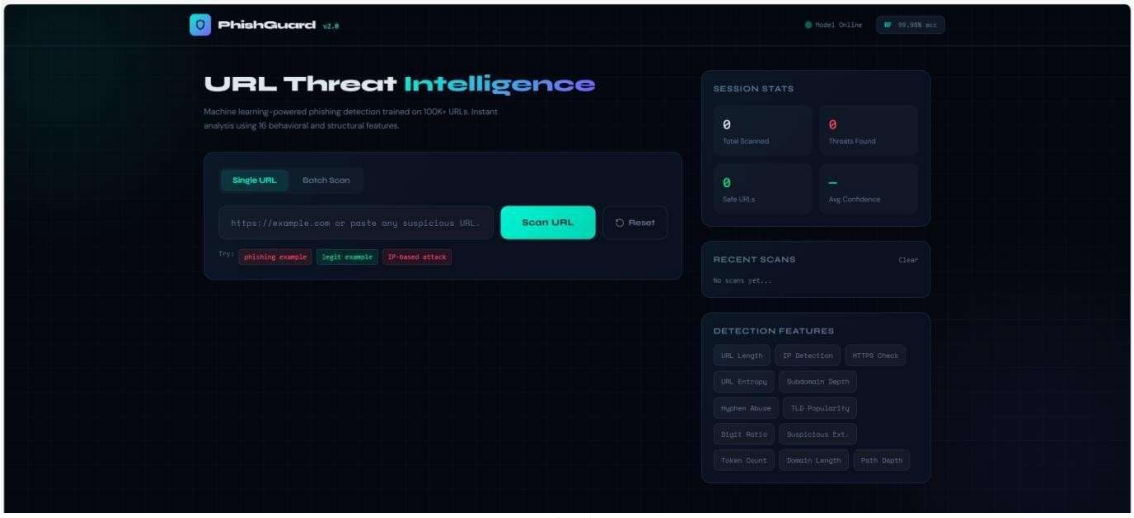


Fig-1

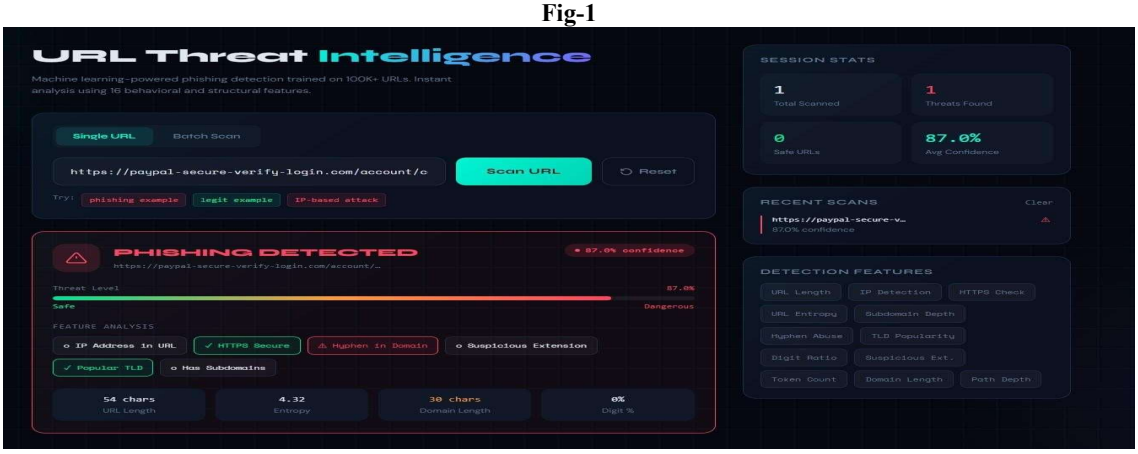


Fig-2

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

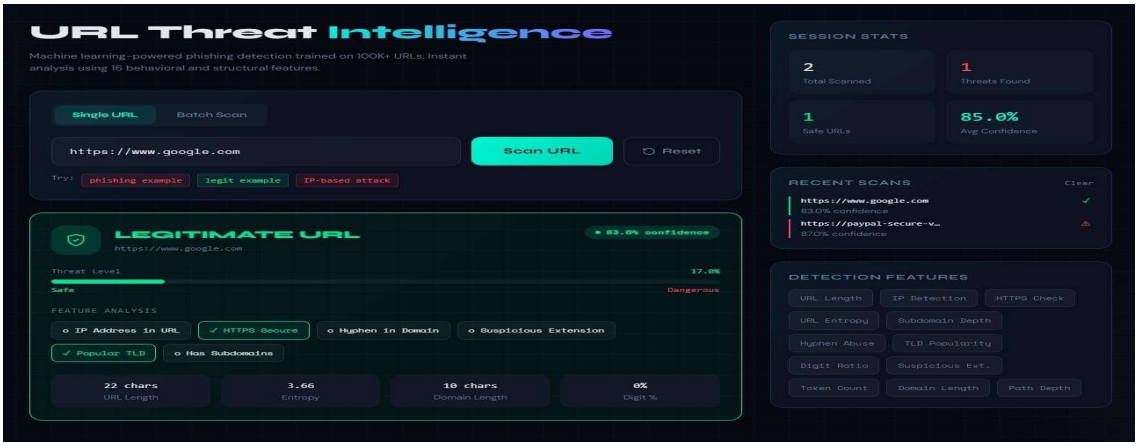


Fig-3

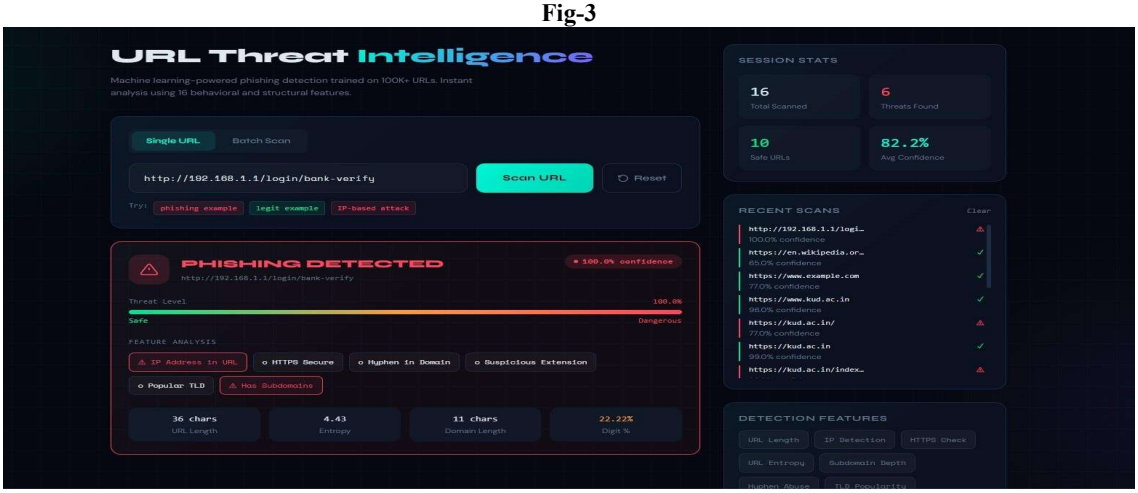


Fig-4

CONCLUSION

The results of this project demonstrate the effectiveness of machine learning techniques in detecting phishing websites using URL-based analysis. By extracting a set of sixteen structural and behavioral features directly from URLs, the system is capable of identifying suspicious patterns without requiring access to webpage content. This approach allows the detection process to remain both safe and efficient, since the system does not need to load potentially malicious web pages during analysis.Among the evaluated machine learning algorithms, the Random Forest classifier achieved the highest performance. Experimental evaluation conducted on a dataset containing more than 101,000 URLs showed that the Random Forest model achieved an accuracy of approximately 99.98 percent. This result indicates that ensemble-based machine learning models can effectively capture patterns associated with phishing URLs and provide reliable classification results.To demonstrate practical applicability, the trained model was integrated into a web application developed using the Flask framework. The application provides an intuitive interface that allows users to perform real-time URL scanning as well as batch analysis of multiple URLs. In addition to displaying prediction results, the system also provides confidence scores and feature analysis to help users understand why a particular URL was classified as phishing or legitimate.Despite its strong performance, the system has certain limitations. One limitation is its inability to fully analyze shortened URLs generated through services such as URL shorteners, which often redirect users to other websites. Another limitation is that the current detection model relies solely on URL structure and does not consider webpage content or dynamic behavior. These limitations highlight potential areas for improvement and motivate future research directions aimed at building more comprehensive phishing detection systems.

FUTURE SCOPE

Although the proposed phishing detection system demonstrates high accuracy and practical usability, several improvements can be implemented to enhance its capabilities and adaptability. Phishing techniques continue to evolve rapidly, and therefore detection systems must also evolve to address new attack strategies. Future enhancements can focus on improving detection accuracy, expanding analysis methods, and integrating additional cybersecurity

features.One potential improvement involves extending the system to detect phishing attempts within email messages. By analyzing email content, sender information, and embedded hyperlinks, the system could identify suspicious communication patterns commonly used in phishing campaigns. Another promising enhancement is the integration of webpage content analysis, which would allow the system to examine HTML structure, embedded scripts, and webpage behavior to identify fraudulent websites more effectively.Advancements in deep learning provide additional opportunities for improving phishing detection systems. Models such as Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and transformer-based architectures can be applied to learn complex patterns from large datasets. These models may improve the system's ability to detect sophisticated phishing URLs that attempt to evade traditional machine learning classifiers.Another important direction involves integrating the system into web browsers through extensions. A browser extension could provide real-time phishing warnings while users browse the internet, offering an additional layer of protection. Cloud deployment is also an important enhancement that would allow the system to scale efficiently and support remote access. Furthermore, developing a mobile application could enable users to perform phishing detection directly from smartphones and other portable devices.In the long term, the detection model can be improved by training it on larger and more diverse datasets collected from multiple regions and industries. Hybrid detection approaches that combine URL-based features, webpage content analysis, and blacklist verification could further improve accuracy and reliability. Another promising direction involves improving model explainability by providing detailed explanations for why a specific URL is classified as malicious.

REFERENCES

1. J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, ACM, 2009, pp. 1245–1254.
2. 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.
3. J. Feng, L. Zou, O. Ye, and J. Han, "Web2Vec: Phishing webpage detection method based on multi-view representation learning," *IEEE Access*, vol. 8, pp. 221214–221224, 2020.
4. Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL detection via convolutional neural networks and attention-based hierarchical recurrent neural networks," in *Proceedings of the 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2019, pp. 403–410.
5. X. Chen, Y. Zhang, J. Li, and H. Wang, "Visual similarity based phishing website detection using color histograms and wavelet hashing," *International Journal of Information Security*, vol. 22, no. 3, pp. 415–428, 2023.
6. Z. Wang, Y. Li, J. Zhang, and Q. Liu, "PDRCNN: A hybrid convolutional neural network and recurrent neural network model for phishing URL detection," *Computers & Security*, vol. 114, 2022.
7. Google Inc., "PhishNet and Google Safe Browsing API for phishing detection," *Google Security White Paper*, 2021.