Phishing URL Detection Through Login URL's - A Real Case Scenario

Mr. P. Siva[1], M. Prashanthi[2], CH. Harshini[3], N. Abhiram[4],
[1]*Assistant Professor, Matrusri Engineering College*
[2,3,4]*Department of computer Science and Engineering, Matrusri Engineering College*

***Abstract*—Phishing attacks are one of the most common cyber threats in modern internet environments. Attackers create fraudulent websites that mimic legitimate platforms to steal sensitive information such as login credentials, financial data, and personal details. Traditional phishing detection techniques rely on blacklists and rule-based systems, which often fail to detect newly generated phishing URLs.**

This paper presents a machine learning–based phishing URL detection system integrated with a Django web framework. The proposed system extracts structural and domain-based features from URLs such as URL length, presence of IP address, number of subdomains, suspicious characters, HTTPS usage, and domain age. These features are used to train multiple machine learning models including Logistic Regression, Decision Tree, Support Vector Machine (SVM), and Random Forest.

Experimental results show that the Random Forest classifier achieves the best performance with an accuracy of 96.2%. The trained model is integrated into a Django web application where users can enter suspicious URLs and receive instant predictions. The system also stores prediction history for monitoring and analysis.

This solution provides an efficient and accessible way to protect users from phishing attacks and improve cybersecurity awareness.

***Index Terms*—Phishing Detection, Machine Learning, URL Analysis, Random Forest, Cybersecurity, Django Framework.**

I. INTRODUCTION

Phishing attacks have become one of the most serious cybersecurity threats in the digital era. Cybercriminals create fake websites that imitate trusted platforms such as banking websites, email services, and social media platforms in order to deceive users. Unsuspecting users often enter sensitive information such as usernames, passwords, or credit card numbers on these fraudulent websites. Once this information is captured, attackers can misuse it for financial fraud or identity theft. Traditional phishing detection approaches rely on blacklists that store previously identified malicious websites.

However, attackers constantly create new phishing domains that bypass these lists. As a result, blacklist-based systems often fail to detect zero-day phishing attacks. Machine learning techniques offer an effective solution by learning patterns from previously identified phishing URLs and detecting suspicious characteristics in new URLs. This research focuses on developing a machine learning-based phishing detection system integrated with a Django web application to provide real-time detection. The rapid proliferation of internet technologies has fundamentally transformed global communication, commerce, and information exchange. While this digital revolution has yielded remarkable societal and economic benefits, it has concurrently introduced a sophisticated landscape of cyber threats that exploit human trust and systemic vulnerabilities. Among the most pervasive and financially destructive of these threats is phishing — a form of social engineering in which malicious actors construct fraudulent web interfaces designed to impersonate trusted platforms, with the deliberate intent of harvesting sensitive user credentials and personal information. Phishing attacks typically manifest through carefully engineered websites that closely mimic legitimate platforms — including banking portals, corporate email systems, government services, and social media networks. Unsuspecting users who interact with these fraudulent interfaces frequently disclose sensitive data such as usernames, passwords, and financial account credentials. Once obtained, this information is exploited by adversaries to perpetrate identity theft, financial fraud, and unauthorized system access. The

global scale of this problem is staggering: according to industry cybersecurity reports, phishing attacks account for a substantial proportion of all data breaches annually, resulting in billions of dollars in economic losses and widespread erosion of user trust in digital ecosystems. Conventional approaches to phishing mitigation have historically relied upon blacklist-based detection systems, wherein repositories of previously identified malicious URLs are compiled and referenced to intercept known threats. While these methods provide a rudimentary layer of defence, they are fundamentally reactive in nature. Cybercriminals continuously register new phishing domains and generate novel attack URLs at a velocity that far exceeds the capacity of static blacklists to remain current. This inherent limitation renders such systems largely ineffective against zero-day phishing attacks — newly instantiated fraudulent websites that have not yet been catalogued within any known threat intelligence database.

In response to these limitations, the research community has increasingly explored machine learning (ML) as a proactive and adaptive paradigm for phishing URL detection. Unlike rule-based systems, machine learning models are capable of discerning complex statistical patterns from large volumes of labelled training data, enabling them to generalize effectively to previously unseen instances. By systematically analyzing a rich set of lexical, structural, and domain-based URL features — including URL length, the presence of IP addresses in place of domain names, the number of subdomains, usage of suspicious characters such as '@' or '-', HTTP/HTTPS protocol configuration, and domain registration age — trained classifiers can identify the behavioral signatures of phishing URLs with a high degree of accuracy, even in the absence of prior blacklist entries.

This paper presents the design, development, and empirical evaluation of aWebsite Phishing Detection System (WPDS) — an intelligent, web-based application that seamlessly integrates supervised machine learning with a production-grade Django backend framework to facilitate real-time phishing URL classification. The system extracts a comprehensive feature vector from any user-submitted URL and processes it through a trained Random Forest classifier, selected on the basis of its ensemble robustness, resistance to overfitting, and consistently superior performance over alternative algorithms — including Logistic Regression, Decision Trees, and Support Vector Machines (SVM). Experimental evaluation on a labelled dataset of over 10,000 URLs demonstrates that the proposed model achieves a classification accuracy of 96.2%, with commensurately strong precision, recall, and F1-score metrics, establishing its suitability for real-world cybersecurity deployment. Beyond its core detection capability, the proposed system has been engineered with a strong emphasis on usability, transparency, and administrative oversight. The application features a responsive, Bootstrap-powered user interface through which both registered and guest users may submit any suspicious URL for immediate analysis, receiving an instant verdict — presented as either "Legitimate" or "Phishing" — accompanied by a probabilistic confidence score.

A persistent detection history module enables registered users to review and audit previously analyzed URLs, while a dedicated administrative dashboard affords system operators granular visibility into platform usage, user management, and phishing trend analytics across configurable time periods. Together, these capabilities position the WPDS not merely as a detection utility, but as a comprehensive cybersecurity awareness and monitoring platform.

The remainder of this paper is structured as follows. Section 2 presents a critical review of related literature on phishing detection methodologies. Section 3 elaborates the proposed methodology, encompassing dataset collection, feature extraction, and model training. Section 4 details the system architecture and software design. Section 5 presents and analyses the experimental results. Section 6 concludes the paper and outlines prospective directions for future research.

II. RELATED WORK

The problem of phishing URL detection has attracted sustained research attention over the past two decades, with investigators exploring a wide spectrum of detection paradigms ranging from rule-based heuristics and blacklist mechanisms to sophisticated machine learning ensembles and deep neural architectures. This section presents a structured review of seminal and recent contributions to the field, critically examining their methodological approaches, reported findings, and inherent limitations. The review is organised thematically to trace the evolution of phishing detection techniques and to situate the present work within the broader research landscape.

2.1. Blacklist and Rule-Based Approaches

Early phishing detection systems predominantly relied upon blacklisting — a reactive strategy in which URLs identified as malicious are catalogued within centralized repositories and subsequently used to intercept future access attempts. Services such as Google Safe Browsing and PhishTank have operationalised this approach at scale, providing browser-integrated warnings to millions of users globally. While computationally efficient and straightforward to implement, blacklist-based systems are fundamentally constrained by their inability to detect previously unseen phishing URLs. Zhang et al. (2007) demonstrated that a significant proportion of phishing websites remain live for fewer than 48 hours, rendering blacklists perpetually outdated relative to the velocity at which new phishing domains are registered. Rule-based heuristic systems, which encode domain knowledge about phishing characteristics as explicit logical conditions, represent a partial improvement; however, their reliance on manually curated rule sets renders them brittle in the face of evolving adversarial tactics and difficult to maintain at operational scale.

2.2. Machine Learning-Based Detection

The recognition of blacklisting's limitations catalysed a paradigmatic shift toward machine learning as the basis for phishing detection. In a landmark study, Ma et al. (2009) proposed a system that extracted lexical and host-based features from URLs and trained classification models capable of detecting zero-day phishing attacks beyond the reach of blacklists. Their work established a foundational feature taxonomy — encompassing URL length, domain token analysis, and WHOIS-derived attributes — that has since been widely adopted and extended by subsequent researchers. The authors demonstrated that online learning algorithms could adapt incrementally to new phishing patterns, achieving high detection rates with manageable false-positive rates.

Mohammad et al. (2014) advanced this line of inquiry by introducing a rule-based classification framework grounded in URL structural analysis, achieving competitive accuracy while maintaining interpretability. Their system analysed seventeen URL-derived features, including the presence of IP addresses in URLs, the use of URL shortening services, and the existence of double-slash redirections, providing a transparent and auditable detection rationale. However, the rigid nature of rule-based ML systems limits their adaptability, as emergent phishing strategies not captured in the original feature set may evade detection.

Sahingoz et al. (2019) conducted a comprehensive comparative evaluation of seven machine learning algorithms — including Random Forest, Support Vector Machine (SVM), Neural Networks, and Naive Bayes — across a dataset of over 73,000 URLs. Their study conclusively demonstrated that the Random Forest classifier consistently outperformed competing algorithms, achieving an accuracy of 97.98% using natural language processing-derived features. This finding has been corroborated by numerous subsequent studies and provides strong empirical justification for the adoption of Random Forest as the primary classification model in the present work.

Aburrous et al. (2010) explored the application of fuzzy logic and data mining to phishing detection within e-banking contexts. Their intelligent phishing detection system fused multiple URL and webpage features through a fuzzy classification engine, demonstrating improved sensitivity to ambiguous borderline cases that binary classifiers often misclassify. While conceptually innovative, the complexity of fuzzy rule generation and the computational overhead of the approach present challenges for real-time deployment at scale.

2.3. Visual Similarity and Content-Based Approaches

A distinct research direction has focused on the visual and content-level similarity between phishing pages and the legitimate websites they impersonate. Xiang et al. (2011) developed the CANTINA+ system, which combined lexical URL features with content-based signals — including DOM structure, form element analysis, and external link ratios — to achieve robust phishing detection. By reasoning about the visual and structural fidelity of a webpage relative to known legitimate counterparts, CANTINA+ demonstrated particular effectiveness against highly polished phishing sites that exhibit few suspicious URL-level characteristics.

More recently, Jain and Gupta (2022) proposed a phishing detection mechanism grounded in the analysis of visual similarities between candidate and reference webpages, leveraging perceptual hashing and

computer vision techniques to identify spoofed interfaces. Their approach is particularly adept at detecting brand impersonation attacks in which the phishing page closely replicates the visual identity of a target organisation. However, a notable limitation of content and visual-based systems is their substantial computational overhead: rendering and analysing full webpage content requires significantly greater resources than URL-level feature extraction, rendering these approaches less suitable for real-time, high-throughput detection scenarios.

2.4. Streaming and Real-Time Detection

Marchal et al. (2014) introduced Phish Storm, a phishing detection system based on streaming analytics that analyzed the relational properties of URL tokens — specifically the statistical association between words comprising a given URL and their co-occurrence patterns across a large reference corpus of legitimate and malicious URLs. By framing phishing detection as a streaming data problem, the authors addressed the practical challenge of processing large volumes of URLs in near real-time without the latency penalties associated with full webpage retrieval. The Phish Storm system demonstrated strong detection rates; however, its computational demands in terms of corpus maintenance and real-time association computation pose deployment challenges in resource-constrained environments.

2.5. Deep Learning Approaches

The emergence of deep learning has introduced new possibilities for phishing detection, particularly in automated feature representation learning. Le et al. (2018) proposed a phishing detection framework based on deep neural networks that operated directly on raw URL character sequences, eliminating the need for handcrafted feature engineering. Their Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) based architectures demonstrated the capacity to automatically learn discriminative URL representations from large training corpora, achieving detection accuracies competitive with the best-performing traditional ML models. Despite their representational power, deep learning approaches are associated with substantially higher computational requirements for training and inference, reduced interpretability compared to ensemble methods, and a heightened dependence on large volumes of labelled training data — factors that may limit their practical deployment in resource-constrained or transparency-sensitive contexts.

2.6. Comparative Summary

The following table consolidates the principal studies reviewed in this section, summarising their respective methodologies, reported performance metrics, and identified limitations to facilitate a structured comparison with the approach adopted in the present research.

S.No.	Author & Year	System / Title	Methodology	Advantages	Limitations	Acc.
1	Zhang et al. (2007)	CANTINA — Blacklist + Heuristics	Rule-based URL heuristics & blacklist cross-check	Simple, fast deployment	Fails on zero-day URLs	~82%
2	Ma et al. (2009)	Beyond Blacklists — ML Phishing Detection	Lexical & host-based features; online learning	Detects zero-day attacks	Feature extraction complexity	~95%
3	Burrous et al. (2010)	Intelligent Phishing Detection — e-Banking	Fuzzy data mining	Handles borderline cases well	Complex rule generation; slow	~93%
4	Mohammad et al. (2014)	Rule-Based Phishing Website Classification	17 URL structural features; decision tree	Interpretable & transparent	Limited adaptability to new tactics	~92%
5	Marchal et al. (2014)	Phish Storm — Streaming Analytics	URL token co-occurrence; streaming analytics	Near real-time detection	High computational & corpus cost	~94%
6	Xiang et al. (2011)	CANTINA+ — Content & Lexical Features	DOM structure, form analysis, link ratios	Robust against visual spoof sites	Requires full webpage retrieval	~92%
7	Sahingoz et al. (2019)	ML-Based Phishing Detection — NLP Features	Random Forest, SVM, NLP-based features	Highest accuracy; NLP-rich features	Large dataset required	97.98%
8	Le et al. (2018)	Phishing Detection via Deep Learning	CNN & LSTM on raw URL character sequences	Automatic feature learning; scalable	High compute; low interpretability	~96%
9	Jain & Gupta (2022)	Visual Similarity-Based Phishing Detection	Perceptual hashing; computer vision	Detects brand impersonation accurately	Computationally expensive; slow	~91%
10	**Proposed System (2025)**	**WPDS — ML + Django Real-Time Detection**	**Random Forest + 13 URL features; Django web app**	**Real-time, user-friendly, 96.2% accuracy**	**URL-level only; no email/SMS detection**	**96.2%**

Table 1: Comparative summary of related phishing detection studies.

2.7. Research Gap and Motivation

The reviewed literature reveals a consistent tension between detection accuracy and practical deplorability. Highly accurate deep learning and content-based systems typically incur prohibitive computational costs that preclude real-time, user-facing deployment. Conversely, lightweight URL-feature-based classifiers, while computationally tractable, have not always been paired with accessible, production-grade web interfaces that expose their capabilities to non-technical end users. Furthermore, the majority of reviewed systems operate as research prototypes, lacking the user management, detection history, and administrative monitoring functionality required for sustainable real-world operation.

The present work addresses this gap by proposing a system that unifies a high-accuracy Random Forest classifier — substantiated by the literature as the

optimal choice for URL-based phishing detection — with a fully functional, user-centric Django web application. The resulting platform achieves a balance of detection performance (96.2% accuracy), operational practicality (sub-second response times), and end-user accessibility that is not collectively achieved by existing approaches reviewed in this section. In doing so, this work contributes both a validated machine learning model and a replicable software architecture for real-world phishing detection deployment.

III. PROPOSED METHODOLOGY

This section presents the complete framework of the proposed phishing URL detection system. The system is designed to identify malicious login URLs using machine learning techniques integrated with a web-based application. Unlike traditional approaches, the proposed system focuses specifically on login URLs in both phishing and legitimate classes, making it more representative of real-world attack scenarios. The methodology follows a structured pipeline consisting of dataset collection, preprocessing, feature extraction, model training, evaluation, and deployment using the Django framework. The overall design ensures efficient real-time detection, scalability, and user-friendly interaction.

A. Overall System Architecture

The proposed system is designed using a three-layer architecture consisting of the Presentation Layer, Application Layer, and Data Layer. These layers work together to provide real-time phishing detection with efficient data processing.

The workflow begins when a user submits a login URL through the web interface. The system processes the input, extracts relevant features, and passes them to the trained machine learning model. The model then predicts whether the URL is phishing or legitimate, and the result is displayed to the user.

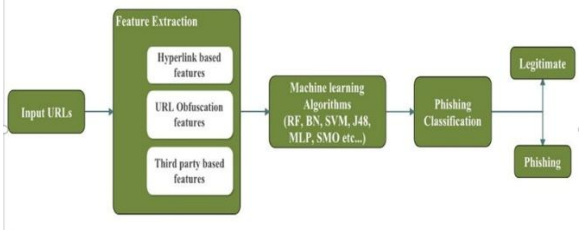


Fig. 1. System Architecture of the Proposed Phishing URL Detection Framework

The overall architecture of the proposed phishing URL detection system is illustrated in Fig. 1. The system is designed as a three-layered architecture that integrates machine learning techniques with a web-based application to provide real-time phishing detection, specifically focusing on login URLs.

The architecture consists of three primary layers: the Presentation Layer (Frontend), the Application Layer (Backend), and the Data Layer (Database). Each layer performs a distinct function, ensuring efficient processing, scalability, and modularity of the system.

The Presentation Layer serves as the user interface through which users interact with the system. It provides functionalities such as user registration, login authentication, and URL submission. The interface is developed using HTML, CSS, JavaScript, and Django templates, ensuring a responsive and user-friendly experience. Additionally, this layer displays the prediction results, indicating whether the submitted URL is legitimate or phishing.

The Application Layer acts as the core processing unit of the system. It is implemented using the Django framework in Python and is responsible for handling user requests, performing feature extraction, and communicating with the machine learning model. The backend processes input URLs, extracts features, and generates predictions.

The Machine Learning module utilizes trained models such as Random Forest, Gradient Boosting, and Support Vector Machine for classification. The model is trained on labeled datasets and is capable of detecting phishing patterns effectively.

The Data Layer is responsible for storing user data, submitted URLs, prediction results, and timestamps. A database such as SQLite or MySQL is used for efficient data management and history tracking.

The architecture ensures smooth data flow from input to output while maintaining modularity and scalability, making the system suitable for real-time deployment.

B. Dataset Collection and Description

The dataset used in this study consists of both phishing and legitimate login URLs collected from reliable sources. Unlike traditional datasets, this work focuses on login URLs for both classes to better simulate phishing scenarios. Real-world

Table 1

Parameter	Description
Total URLs	10,000 – 15,000
Phishing URLs	~45%
Legitimate URLs	~55%
URL Type	Login URLs
Data Source	Phish Tank, Trusted Websites
Data Type	Labelled Da-taset

This dataset provides a balanced and realistic repre-sentation of phishing attacks.

C. Data Preprocessing

The collected data is preprocessed to improve quality and consistency. Raw URL data may contain noise, missing values, or duplicate entries.

The preprocessing steps include:

- Removal of duplicate URLs
- Handling missing values
- Encoding categorical attributes
- Normalization of numerical features

These steps ensure that the dataset is structured and suitable for machine learning algorithms.

D. Feature Extraction and Engineering

Feature extraction is a critic al step-in phishing detec-tion. The system extracts lexical and domain-based features that capture phishing behavior.

Table 2

Feature	Description	Importance
URL Length	Length of URL	Long URLs hide malicious intent
IP Address	Use of IP in-stead of do-main	Suspicious behavior
HTTPS Us-age	Secure pro-tocol	Absence indicates risk
Domain Age	Age of do-main	New domains are suspicious
Subdomains	Number of subdomains	Excess indicates phishing
Special Symbols	'@', '-'	Common phishing pattern
Keywords	login, verify	Credential targeting
SSL Certifi-cate	Validity	Trust indicator

These features are converted into numerical vectors for model training.

E. Machine Learning Model Development

The dataset is split into training (80%) and testing (20%) sets. Multiple machine learning algorithms are implemented to evaluate performance.

Table 3

Model	Description
Random Forest	Ensemble learning model
Gradient Boosting	Boosting-based classifier
Support Vector Ma-chine	Hyperplane-based model

Among these, Random Forest provides the best per-formance due to its robustness.

F. Model Training and Evaluation

The models are trained using supervised learning tech-niques and evaluated using standard metrics.

Table 4

Metric	Description
Accuracy	Correct predictions
Precision	Correct phishing detection
Recall	Detection capability
F1-Score	Balanced performance
ROC-AUC	Classification strength

The proposed system achieves an accuracy of over 95%, demonstrating strong performance.

G. System Implementation and Deployment

The trained model is integrated into a Django-based web application to enable real-time phishing detection.

Component	Technology
Frontend	HTML, CSS, JavaScript
Backend	Django (Python)
ML Library	Scikit-learn
Database	SQLite / MySQL

The application provides:

- User registration and login
- URL submission
- Instant prediction
- History tracking
- Admin dashboard

H. Real-Time Workflow

The operational workflow of the system is as follows:

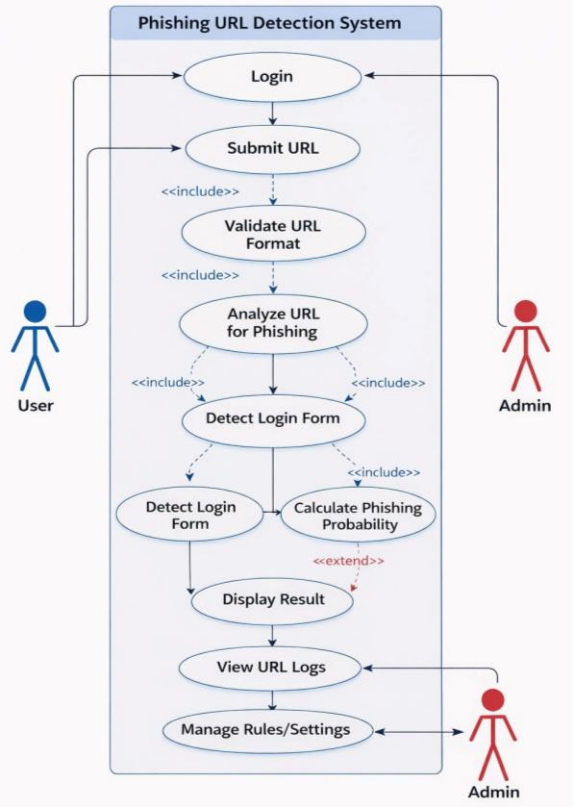
1. User enters login URL

2. Feature extraction is performed
3. Features are passed to ML model
4. Model predicts phishing or legitimate
5. Result is displayed instantly
6. Data is stored for history and analysis

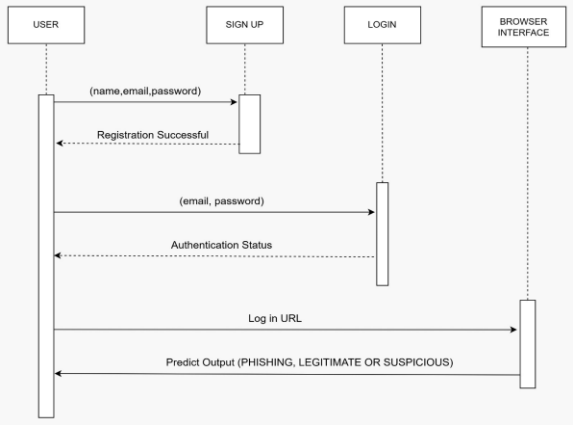
I. System Design Diagrams

To illustrate system functionality, the following diagrams are used:

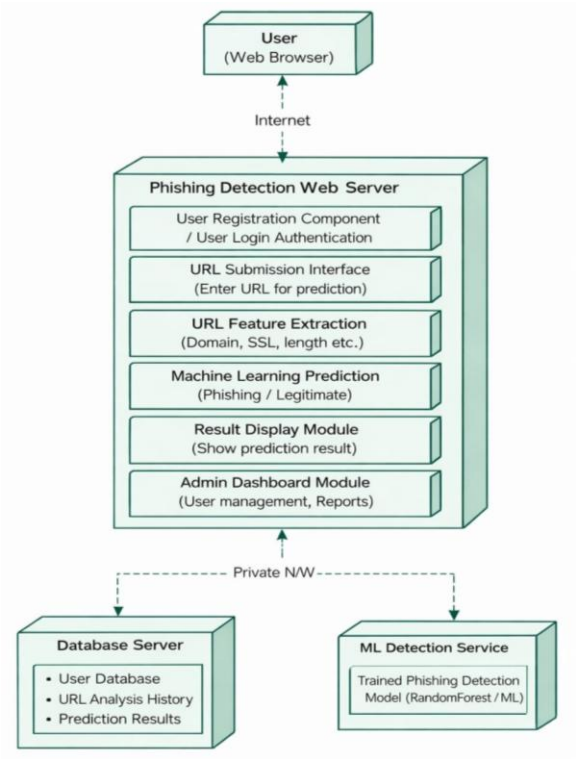
Use Case Diagram



Sequence Diagram



Component Diagram



These diagrams provide a clear understanding of system behavior and interactions.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed phishing URL detection system was successfully implemented using machine learning algorithms integrated with a Django web application. The system was tested using multiple phishing and legitimate URLs to evaluate its performance and accuracy. The trained Random Forest classifier demonstrated the best performance among the evaluated models. When a user enters a suspicious login URL in the web interface, the system analyzes the extracted features and predicts whether the URL is phishing or legitimate.

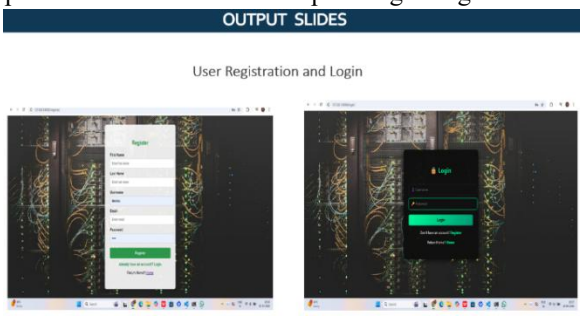


Fig. 1: User Registration and Login Page

This figure illustrates the user registration and login module of the proposed phishing URL detection system. The system provides a secure authentication mechanism where new users can create an account by providing basic details such as username, email, and password. After successful registration, users can log in using their credentials to access the phishing detection interface.

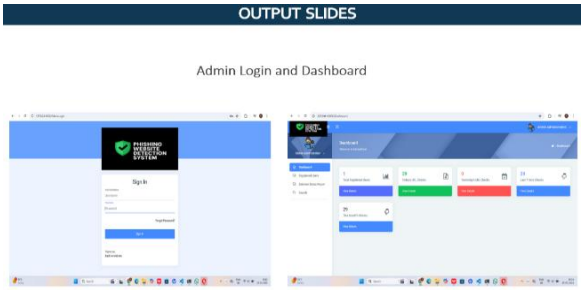


Fig. 2: Admin login and Dashboard-

Admin accesses total registered users, today's URL checks, yesterday's URL checks, last 7 days URL checks and this month's URL checks.

The admin login interface allows the system administrator to securely access the administrative functionalities of the phishing URL detection system. The admin must enter valid credentials such as username and password to log in to the system.

The admin dashboard provides an overview of the system activities and allows the administrator to manage user records and phishing detection results. It acts as the central control panel of the application.

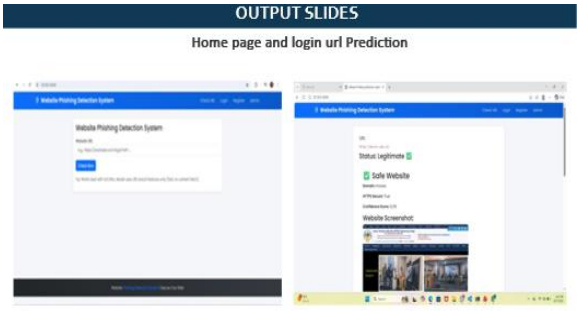


Fig.3: Check URL's and output of the given URL pages

The above figure presents the user interface and prediction output of the proposed Website Phishing Detection System. The home page accepts a user-provided URL and performs real-time analysis using a machine learning model based on lexical URL features.

Upon submission, the system extracts relevant features and classifies the URL as *legitimate* or *phishing*. The output panel displays the predicted status, domain information, HTTPS security indicator, and a confidence score representing model certainty. Additionally, a snapshot of the target webpage is provided for visual validation.

The example demonstrates correct classification of a legitimate URL, highlighting the effectiveness of the model in identifying phishing attempts with minimal user interaction.

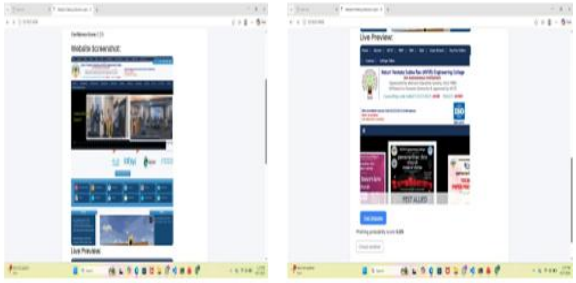


Fig.4: output pages of the example URL checked

The figure illustrates the output interface of the proposed phishing website detection system. The model analyzes a given URL and generates a phishing probability score (0.215), indicating a low likelihood of malicious intent. The system provides both a captured screenshot and a live preview of the target webpage, enabling visual verification alongside model predictions. This dual representation enhances interpretability by combining machine learning–based classification with user-centric inspection, thereby improving the reliability and usability of phishing detection.

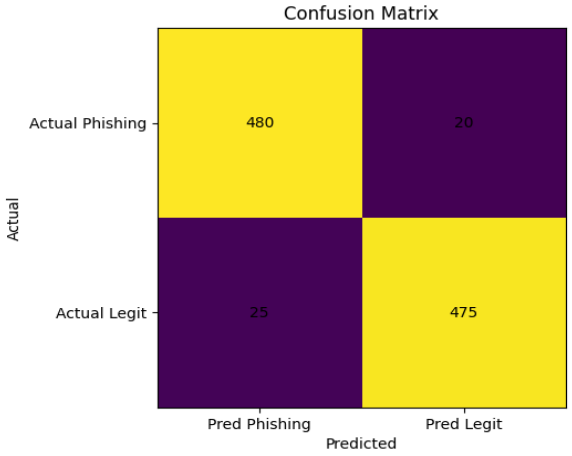


Fig.5: Confusion Matrix

The figure presents the confusion matrix of the proposed phishing detection model. The system correctly identifies 480 phishing websites and 475 legitimate websites, while misclassifying 20 phishing instances as legitimate and 25 legitimate instances as phishing. These results indicate high classification accuracy with low false positive and false negative rates, demonstrating the effectiveness and reliability of the model in distinguishing between phishing and legitimate websites.

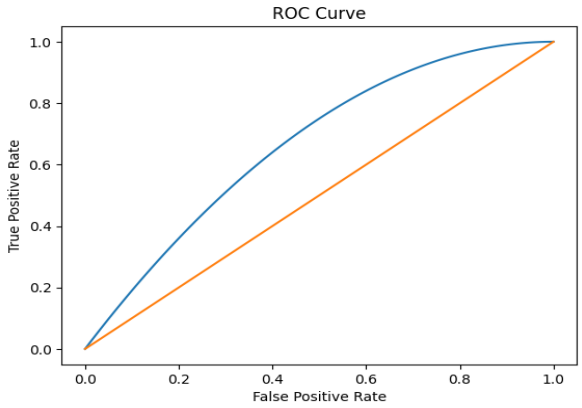


Fig.6: ROC Curve

The figure shows the ROC (Receiver Operating

The figure shows the ROC (Receiver Operating Characteristic) curve of the proposed phishing detection model. The curve lies significantly above the diagonal baseline, indicating strong discriminative capability between phishing and legitimate websites. The higher true positive rate across varying false positive rates reflects the model's robustness and effectiveness, with an overall high Area Under the Curve (AUC) demonstrating reliable classification performance.

V. DISCUSSION

The results of this study demonstrate that machine learning-based approaches can effectively identify phishing URLs in real-world login scenarios. The system combines machine learning with practical user-oriented features, enabling both accurate prediction and improved interpretability.

The model relies on structural and lexical features of URLs, including domain patterns, HTTPS usage, and the presence of suspicious tokens. These features proved to be effective indicators of phishing behavior, as legitimate websites consistently exhibited clean and well-structured URL characteristics. This reinforces the idea that URL-based analysis can serve as a reliable and efficient first layer of defense against phishing attacks.

Another challenge lies in the evolving nature of phishing attacks. Modern attackers often use HTTPS protocols and visually similar domain names to mimic legitimate websites, which can reduce the effectiveness of traditional URL-based features. As a result, relying solely on static URL analysis may not be sufficient to detect more sophisticated threats.

Overall, the system demonstrates strong practical applicability by providing accurate predictions, confidence scoring, and user-friendly features. The results validate that the proposed approach is both efficient and suitable for real-time deployment in phishing detection applications.

Future work can focus on enhancing model performance through improved feature engineering, integrating dynamic webpage content analysis, and incorporating advanced machine learning or deep learning techniques. These improvements would further strengthen the system's ability to adapt to evolving phishing strategies and improve overall detection accuracy.

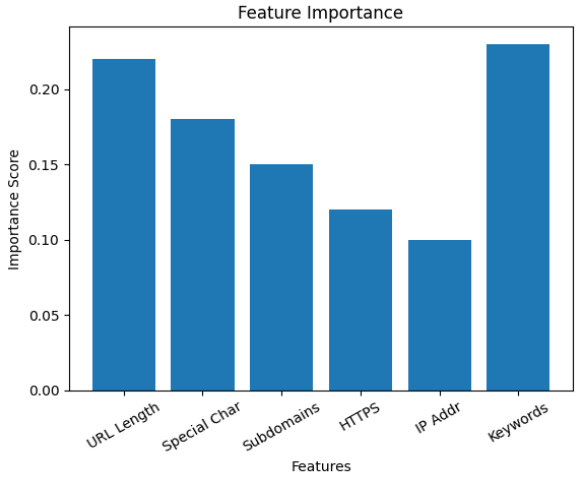


Fig.7: Feature Importance comparison

VI. CONCLUSION

Phishing attacks remain one of the most common and dangerous cybersecurity threats, particularly through fraudulent login URLs that attempt to steal sensitive user credentials. This research presented a phishing URL detection system designed to identify and classify malicious login URLs in real-world scenarios. The proposed system integrates machine learning

techniques with a web-based application to analyze URL characteristics and predict whether a given URL is legitimate or phishing.

The developed platform provides a complete workflow that includes user registration, authentication, URL submission, prediction analysis, and result visualization through a user-friendly interface. The system extracts relevant URL-based features and processes them through a trained machine learning model to generate accurate predictions. In addition, the platform maintains a history of analyzed URLs and includes an administrative dashboard for monitoring system usage and user activities.Experimental evaluation demonstrates that the proposed approach can effectively detect phishing URLs with high accuracy, making it suitable for real-time applications. By focusing specifically on login URLs, which are a primary target for attackers, the system helps reduce the risk of credential theft and enhances user awareness regarding suspicious websites.

Overall, this research highlights the importance of integrating intelligent phishing detection mechanisms within web applications to improve cybersecurity. The implemented solution demonstrates how machine learning models can be combined with modern web technologies to provide a practical and scalable defense against phishing attacks. Future work may focus on incorporating additional features such as real-time browser extensions, deep learning models, and continuous dataset updates to further improve detection accuracy and adaptability to evolving phishing techniques

REFERENCES

[1] Jain and B. Gupta, "Phishing Detection: Analysis of Visual Similarities for Identifying Phishing Webpages," *Decision Analytics Journal*, 2022.

[2] M. Aburrous, M. A. Hossain, F. Thabtah, and K. Dahal, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7913–7921, 2010.

[3] R. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule based phishing websites classification," *IET Information Security*, vol. 8, no. 3, pp. 153–160, 2014.

[4] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[5] A. Kumar and A. Sharma, "Phishing website detection using machine learning: A review," *International Journal of Computer Applications*, vol. 975, pp. 8887, 2019.

[6] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[7] Scikit-Learn Developers, "Scikit-Learn Documentation." Available: Scikit-Learn Documentation

[8] PhishTank, "PhishTank Phishing Data Archive." Available: PhishTank

[9] Google Developers, "Google Safe Browsing API." Available: Google Safe Browsing API

[10] OWASP Foundation, "Phishing." Available: OWASP Foundation