PHISHING URL DECTECTOR TOOL USING THE MACHINE LEARNING

Mrs J.Ranganayaki,Phd
Professor,
Department of school of computing
Bharath Institute of science and technology(BIST)
173chennai,selayur,Tambaram
Chennai-600073
jranganayaki.cse@bharathuniv.ac.in

A. Jagruth
School of computing
Bharath Institute of science and technology(BIST)
173chennai,seliayur,Tambaram
adapajagruth@gmail.com

B. Sai Venkat
School of computing
Bharath Institute of science and technology(BIST)
173chennai,seliayur,Tambaram
saivenkatbadugu@gmail.com

A. Navadeep Krishna
School of computing
Bharath Institute of science and technology(BIST)
173chennai,seliayur,Tambaram
adapanavadeep99@gmail.com

A. Jithendra Reddy
School of computing
Bharath Institute of science and technology(BIST)
173chennai,seliayur,Tambaram
jithendrareddy196@gmail.com

Abstract **: Phishing attacks have become one of the most critical and rapidly growing threats in cybersecurity. With increasing dependence on online services such as banking, e-commerce, and communication, attackers exploit user trust by creating deceptive websites that mimic legitimate platforms. These phishing sites aim to steal sensitive information such as usernames, passwords, and financial details .Traditional detection methods like blacklist-based systems fail to identify newly created phishing URLs. To overcome these limitations, this paper proposes a machine learning-based phishing URL detection system using structural and lexical features. The system employs a Random Forest classifier to achieve high accuracy and real-time detection capability. Experimental results demonstrate strong performance in terms of accuracy, precision, recall, and F1-score. The proposed system is efficient, scalable, and suitable for real-world cybersecurity applications.**

I. INTRODUCTION

The rapid growth of internet technologies has transformed the way individuals and organizations interact, communicate, and conduct business. Online services such as banking, e-commerce, social networking, and cloud computing have become an integral part of daily life. However, this increased reliance on digital platforms has also led to a significant rise in cyber threats, among which phishing attacks are one of the most prevalent and dangerous. Phishing is a form of cybercrime in which attackers attempt to deceive users into revealing sensitive information by impersonating legitimate entities through fraudulent websites, emails, or messages.

Phishing attacks typically involve the creation of fake websites that closely resemble trusted platforms such as banking portals, payment gateways, and social media sites. These websites are designed with high precision, often replicating the visual appearance, logos, and structure of legitimate websites, making it difficult for users to distinguish between genuine and malicious sites. Attackers distribute phishing links through various channels, including emails, SMS messages, and social media platforms, increasing the likelihood of user interaction. Once users enter their credentials or sensitive data, the information is captured and exploited for malicious purposes such as financial fraud, identity theft, or unauthorized access.

The impact of phishing attacks has grown significantly over the years, both in terms of frequency and sophistication. Modern phishing techniques have evolved beyond simple deceptive URLs to include advanced strategies such as URL obfuscation, domain spoofing, and the use of HTTPS to create a false sense of security. The presence of HTTPS, which was traditionally considered a sign of a secure website, is now frequently used by attackers to gain user trust. As a result, users can no longer rely solely on visual cues or basic indicators to determine the authenticity of a website.

Traditional approaches to phishing detection include blacklist-based systems and heuristic-based methods. Blacklist-based systems maintain a database of known phishing URLs and block access to them. While these systems are effective against previously identified threats, they fail to detect newly created or zero-day phishing websites. Since phishing sites are often short-lived and continuously changing, maintaining an up-to-date blacklist is a challenging task. There is always a delay between the discovery of a phishing site and its inclusion in the blacklist, during which users remain vulnerable.

Heuristic-based methods attempt to identify phishing websites by analyzing specific characteristics or patterns, such as unusual URL structures, presence of suspicious keywords, or abnormal domain behavior. Although these methods provide some level of detection for unknown attacks, they are limited by predefined rules and lack adaptability. Attackers can easily bypass these rules by modifying their techniques, making heuristic approaches less reliable in the long term.

To overcome the limitations of traditional methods, Machine Learning (ML) has emerged as a promising solution for phishing detection. Machine learning algorithms are capable of learning patterns from large datasets and making predictions based on previously unseen data. By analyzing features extracted from URLs, ML models can identify subtle differences between legitimate and phishing websites that may not be easily detectable by human users or rule-based systems.

In this research, a Machine Learning-based approach is proposed for detecting phishing URLs using structural and lexical features. The system focuses on analyzing the URL itself rather than relying on external data sources such as webpage content or third-party services. This approach offers several advantages, including faster processing, reduced dependency on external factors, and improved scalability. The features considered in this study include URL length, number of dots, presence of special characters, use of HTTPS, number of subdomains, and occurrence of suspicious keywords. These features are selected based on common characteristics observed in phishing URLs.

Among various machine learning algorithms, the Random Forest classifier is chosen for this study due to its robustness and high performance in classification tasks. Random Forest is an ensemble learning technique that combines multiple decision trees to improve accuracy and reduce overfitting. It is particularly effective in handling structured data and provides insights into feature importance, making it suitable for phishing detection applications.

The proposed system is designed to operate in real-time, allowing users to input a URL and receive an instant prediction indicating whether the URL is phishing or legitimate. To achieve this, the machine learning model is integrated with a backend API developed using Fast API, and a simple frontend interface is provided for user interaction. This end-to-end system demonstrates the practical applicability of machine learning in enhancing cybersecurity measures.

The significance of this research lies in its ability to provide an efficient and scalable solution for phishing detection. By leveraging machine learning techniques, the system can adapt to evolving phishing strategies and improve detection accuracy over time. Additionally, the lightweight nature of the proposed approach makes it suitable for deployment in various environments, including web applications, browser extensions, and enterprise security systems.

Despite its advantages, the proposed system has certain limitations. It primarily relies on URL-based features and does not analyze the content of web pages or user behavior. As a result, highly sophisticated phishing attacks that closely mimic legitimate URLs may not always be detected. Furthermore, the effectiveness of the model depends on the quality and diversity of the training dataset. Continuous updates and retraining are necessary to maintain optimal performance.

In conclusion, phishing remains a critical challenge in the field of cybersecurity, requiring advanced and adaptive solutions. Machine learning offers a powerful approach to address this challenge by enabling automated and intelligent detection of malicious URLs. The proposed system demonstrates the potential of ML-based techniques in improving online security and protecting users from phishing attacks. Future research can explore the integration of additional features, such as webpage content analysis and deep learning models, to further enhance detection capabilities.

II. LITERATURE REVIEW

The rapid increase in phishing attacks over the past decade has led to extensive research in the field of phishing detection. Various approaches have been proposed by researchers, ranging from traditional blacklist-based systems to advanced machine learning and deep learning techniques. This section reviews the existing work related to phishing detection, highlighting their methodologies, strengths, and limitations.

One of the earliest approaches to phishing detection is the use of blacklist-based systems. These systems maintain a database of known phishing URLs and block access when a match is found. Popular implementations such as Google Safe Browsing and Phish Tank rely on continuously updated blacklists to identify malicious websites. While this approach is simple and efficient, it suffers from a major drawback: it cannot detect newly generated or zero-day phishing attacks. As phishing websites are often short-lived and dynamically created, there is always a delay in updating the blacklist, making users vulnerable during that time window. Researchers have consistently pointed out that blacklist-based detection alone is insufficient for comprehensive protection against phishing attacks.

To address the limitations of blacklist methods, heuristic-based approaches were introduced. These techniques analyze specific characteristics of URLs and web pages to identify suspicious patterns. Features such as URL length, presence of special characters (e.g., '@', '-'), number of subdomains, and use of misleading domain names are commonly used in heuristic detection. Some studies proposed frameworks that analyze lexical features of URLs to detect phishing attempts. Although heuristic methods can detect some unknown phishing websites, they rely heavily on predefined rules, which makes them less adaptable to evolving attack strategies. Attackers can easily modify their techniques to bypass these rules, reducing the effectiveness of heuristic systems over time.

With advancements in data-driven techniques, machine learning-based approaches have gained significant attention in phishing detection. These methods involve training models on

labeled datasets containing both legitimate and phishing URLs. The models learn patterns and relationships between extracted features and the classification labels. Research has shown that machine learning algorithms can effectively identify malicious websites by analyzing URL-based features. These approaches outperform traditional blacklist and heuristic methods in detecting previously unseen phishing attacks.

Several machine learning algorithms have been explored in this domain, including Logistic Regression, Decision Trees, Support Vector Machines, and Random Forest. Each algorithm has its own advantages and limitations. Logistic Regression provides a simple baseline model but may not capture complex patterns in data. Decision Trees offer interpretability but are prone to overfitting. Support Vector Machines are effective in high-dimensional spaces but require careful parameter tuning. Among these, Random Forest has emerged as one of the most reliable algorithms due to its ensemble nature, which combines multiple decision trees to improve accuracy and reduce overfitting.

Research studies have also highlighted the importance of URL structure in phishing detection. Attackers often manipulate URLs using techniques such as encoding, excessive subdomains, and misleading domain names to deceive users. This emphasizes the need for robust feature extraction methods that can capture these patterns effectively. Another approach focuses specifically on URL-based detection systems. These systems analyze only the URL without accessing webpage content or external data sources. Studies have shown that such approaches can achieve high accuracy while maintaining low computational cost. This is particularly beneficial for real-time applications, as it reduces processing time and improves system efficiency.

In addition to traditional machine learning techniques, deep learning approaches have also been explored for phishing detection. Neural networks, including Convolutional Neural Networks and Recurrent Neural Networks, are capable of learning complex patterns from data. These models can achieve very high accuracy; however, they require large datasets and significant computational resources. This makes them less suitable for lightweight systems and real-time applications. Hybrid approaches that combine multiple techniques have also been proposed to improve detection performance. For example, some systems integrate blacklist databases with machine learning models to enhance accuracy and reduce false positives. Others combine heuristic rules with machine learning algorithms to improve feature representation. While these hybrid systems show improved performance, they often increase system complexity and require additional maintenance.

Despite the progress made in phishing detection research, several challenges remain. One of the key challenges is the detection of zero-day phishing attacks, which are newly created and not present in existing datasets. Another challenge is adapting detection systems to evolving attack techniques. Attackers continuously develop new methods to bypass detection mechanisms, such as using URL shortening services, homograph attacks, and HTTPS-based phishing websites.

Furthermore, many existing systems rely on external data sources, such as webpage content, DNS records, or third-party APIs. While these features can improve accuracy, they also introduce latency and dependency on external services, which may not be suitable for real-time applications. Therefore, there is a need for lightweight and efficient systems that can operate using only URL-based features. Based on the literature review, it is evident that machine learning provides a promising solution for phishing detection. Among various approaches, URL-based feature analysis combined with ensemble learning techniques offers a good balance between accuracy, efficiency, and scalability. This forms the foundation for the proposed system.In conclusion, traditional methods such as blacklist and heuristic approaches are limited in their ability to detect modern phishing attacks. Machine learning techniques, particularly ensemble models, provide a more effective and adaptable solution. However, continuous improvement and updates are necessary to keep up with evolving cybersecurity threats.

III EXISTING SYSTEM

Phishing detection has been an active area of research and development for many years, leading to the creation of several existing systems aimed at identifying and preventing phishing attacks. These systems primarily rely on traditional techniques such as blacklist-based detection, heuristic analysis, and basic rule-based mechanisms. While these approaches have contributed significantly to improving cybersecurity, they still face several limitations in handling modern and sophisticated phishing attacks.

One of the most commonly used existing systems is the blacklist-based detection approach. In this method, a database of known phishing URLs is maintained and continuously updated. When a user attempts to access a website, the URL is compared against this database. If a match is found, the system blocks access and warns the user. This approach is widely implemented in browsers and security tools because of its simplicity and speed. However, it is only effective against previously identified phishing websites. Since new phishing URLs are constantly being generated, blacklist systems fail to detect zero-day attacks. There is always a delay between the creation of a phishing site and its addition to the blacklist, leaving users vulnerable during that period.

Another existing approach is heuristic-based detection. This method analyzes the characteristics of a URL or webpage to identify suspicious patterns. For example, features such as long URLs, excessive use of special characters, multiple subdomains, and the presence of misleading keywords like “login” or “verify” are considered indicators of phishing. While heuristic methods can detect some unknown phishing websites, they depend on predefined rules and patterns. As attackers continuously evolve their techniques, these static rules become less effective. Moreover, heuristic systems may generate false positives, incorrectly classifying legitimate websites as phishing. Some systems also use content-based detection, where the actual webpage content is analyzed to determine its authenticity. These systems examine elements such as HTML structure, images, scripts, and forms to identify similarities with known phishing websites. Although content-based approaches can improve detection accuracy, they require more processing time and computational resources. Additionally, they depend on network access to retrieve webpage data, which may introduce latency and reduce system efficiency in real-time applications.

In addition, browser-based security mechanisms are widely used as existing solutions. Modern web browsers include built-in phishing protection features that warn users when they attempt to visit suspicious websites. These systems typically combine blacklist databases with heuristic checks to provide basic protection. While useful, they are not foolproof and can sometimes fail to detect newly created or well-designed phishing websites.

Another limitation of existing systems is their lack of adaptability. Most traditional approaches do not learn from new data or improve over time. They rely on fixed rules or manually updated databases, making them less effective against evolving phishing techniques. Attackers often exploit this limitation by creating URLs that bypass detection rules, such as using URL shortening services, homograph attacks, or slight variations of legitimate domain names.

Furthermore, many existing systems require user awareness and manual judgment. Users are often expected to identify suspicious URLs or heed warning messages. However, not all users have sufficient knowledge of cybersecurity practices, and many tend to ignore warnings, increasing the risk of falling victim to phishing attacks.

Overall, while existing systems provide a basic level of protection against phishing attacks, they are not sufficient to address the growing complexity and frequency of modern cyber threats. Their reliance on static rules, inability to detect zero-day attacks, and lack of adaptability highlight the need for more intelligent and automated solutions. These limitations have motivated the development of advanced approaches, particularly those based on machine learning, which can learn patterns from data and detect both known and unknown phishing URLs more effectively.

IV OBJECTIVES

The primary objective of this research is to design and develop an efficient and reliable system for detecting phishing URLs using machine learning techniques. With the increasing number of cyber threats and the growing dependence on online platforms, there is a strong need for intelligent systems that can automatically identify malicious websites and protect users from potential attacks.

The main goal of the proposed system is to accurately classify a given URL as either phishing or legitimate by analyzing its structural and lexical features. Unlike traditional methods that rely on static rules or predefined databases, this system aims to use a data-driven approach that can learn patterns from past data and adapt to new and evolving phishing techniques.

A key objective of this work is to extract meaningful features from URLs that can effectively represent the characteristics of phishing websites. These features include URL length, number of dots, presence of special characters, use of HTTPS, number of subdomains, and occurrence of suspicious keywords. By transforming raw URLs into structured feature vectors, the system enables machine learning models to perform accurate classification. Another important objective is to implement and evaluate a suitable machine learning algorithm for phishing detection. In this research, the Random Forest classifier is selected due to its robustness, high accuracy, and ability to handle complex relationships between features. The system aims to achieve high performance in terms of accuracy, precision, recall, and F1 score, ensuring reliable detection of phishing URLs while minimizing false positives and false negatives.

The proposed system also focuses on real-time detection capability. It is designed to process user input quickly and provide instant predictions, making it practical for real-world applications. This is achieved by using lightweight feature extraction techniques and an efficient machine learning model that does not require extensive computational resources In addition to accuracy and performance, usability is another key objective of this system. A simple and user-friendly interface is developed to allow users to easily input URLs and receive clear results. The system is designed to be accessible to both technical and non-technical users, ensuring wider adoption and usability.

Scalability is also considered as an important objective. The system is designed in a modular manner, allowing it to be extended with additional features or integrated into larger cybersecurity frameworks. It can be deployed as a web application, browser extension, or integrated into enterprise-level security systems.

Finally, the system aims to provide a foundation for future research and development in phishing detection. It can be further enhanced by incorporating additional features such as webpage content analysis, deep learning models, and real-time threat intelligence data. This ensures that the system remains adaptable to emerging threats and evolving attack strategies.

In summary, the objectives of this research are focused on developing

V. METHODOLOGY

The methodology adopted in this research focuses on developing an efficient and reliable phishing URL detection system using machine learning techniques. The process follows a structured pipeline that includes data collection, preprocessing, feature extraction, model development, evaluation, and deployment. Each stage is carefully designed to ensure accuracy, scalability, and real-time performance of the system.

The first step in the methodology is data collection. A dataset containing both legitimate and phishing URLs is required to train the machine learning model. The dataset is gathered from publicly available sources such as phishing repositories and open datasets. These datasets include labeled URLs, where each entry is classified as either phishing or legitimate. The quality and diversity of the dataset play a crucial role in determining the performance of the model..

Feature Name	Description	Type
URL Length	Total number of characters in the URL	Numerical
Number of Dots	Count of "." in the URL	Numerical
Presence of @	Indicates whether "@" symbol exists in URL	Binary
Presence of Hyphen (-)	Checks if "-" is present in URL	Binary
HTTPS Usage	Indicates if URL uses secure protocol (HTTPS)	Binary
Number of Subdomains	Count of subdomains in the URL	Numerical

Feature Name	Description	Type
Suspicious Keywords	Presence of keywords like login, verify, bank	Binary

Table 1: Extracted URL Features and Description

The core component of the methodology is features extraction. In this stage, meaningful attributes are derived from the URLs to represent their characteristics in a structured format. Instead of analyzing webpage content, the system focuses on URL-based features, which makes the process faster and more efficient. Several important features are extracted, including URL length, number of dots, presence of special characters such as '@' and '-', usage of HTTPS protocol, number of subdomains, and occurrence of suspicious keywords like “login,” “secure,” “verify,” and “bank.” These features are selected based on common patterns observed in phishing URLs. The extracted features are then converted into numerical values, forming a feature vector that can be used for machine learning.
After feature extraction, the dataset is divided into training and testing sets. Typically, a large portion of the data is used for training the model, while a smaller portion is reserved for testing. This split allows the model to learn patterns from the training data and then evaluate its performance on unseen data. Proper data splitting is essential to avoid overfitting and to ensure that the model generalizes well to new inputs.

The next stage involves model selection and training. Various machine learning algorithms can be used for classification tasks, but in this research, the Random Forest classifier is chosen due to its robustness and high accuracy. Random Forest is an ensemble learning technique that builds multiple decision trees and combines their predictions to produce a final result. This approach reduces overfitting and improves overall performance. During training, the model learns the relationship between the extracted features and the corresponding labels, enabling it to classify new URLs effectively.

Following the training phase, the model is evaluated using standard performance metrics. These metrics include accuracy, precision, recall, and F1 score. Accuracy measures the overall correctness of the model, while precision and recall evaluate its performance in identifying phishing URLs. The F1 score provides a balance between precision and recall. These metrics help in assessing the effectiveness of the model and identifying areas for improvement. A well-performing model should achieve high accuracy while maintaining a balance between false positives and false negatives.

To further enhance performance, model optimization techniques may be applied. These include tuning hyperparameters, improving feature selection, and increasing dataset size. Optimization helps in refining the model and achieving better results. It also ensures that the model performs consistently across different datasets and scenarios. Once the model is trained and evaluated, it is deployed for real-time usage. The trained model is saved and integrated into a backend system de veloped using Fast API. This backend serves as an interface between the user and the machine learning model. When a user inputs a URL, the system processes the request, extracts features, and uses the model to generate a prediction. The result is then returned to the user in a simple and understandable format.

To make the system user-friendly, a frontend interface is developed. The frontend allows users to input URLs and view results instantly. This integration of frontend, backend, and machine learning components creates a complete end-to-end system capable of real-time phishing detection.

The overall workflow of the methodology can be summarized as follows: a user enters a URL, the system extracts relevant features, the trained model analyzes these features, and the final prediction is displayed. This streamlined process ensures fast and accurate detection of phishing URLs. One of the key advantages of this methodology is its efficiency. By focusing only on URL-based features, the system avoids the need for complex webpage analysis, reducing processing time and computational cost. Additionally, the use of machine learning allows the system to adapt to new patterns and improve over time.
However, the methodology also has certain limitations. Since it relies only on URL features, it may not detect highly sophisticated phishing attacks that closely mimic legitimate URLs. Furthermore, the performance of the system depends on the quality of the dataset and the selected features. Regular updates and retraining are necessary to maintain effectiveness.

In conclusion, the methodology provides a systematic approach to developing a phishing detection system using machine learning. It combines data processing, feature engineering, model training, and system integration to create an efficient and scalable solution. This approach demonstrates the practical application of machine learning in cybersecurity and provides a strong foundation for further enhancements.

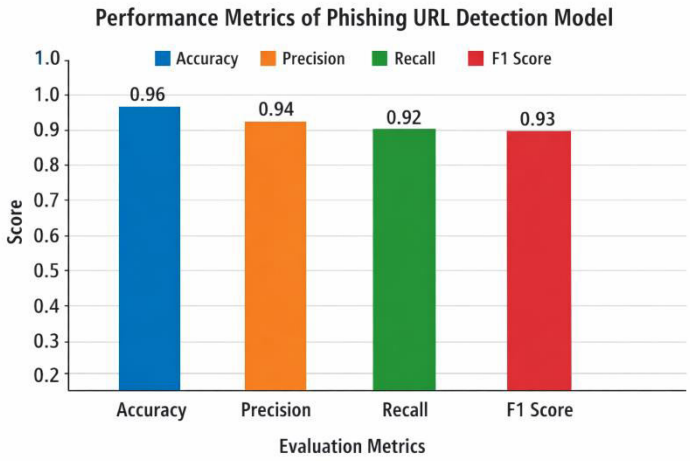

Figure 1: Performance Metrics Of Phishing URL Detector Model

VI. SYSTEM ARCHITECTURE

The system architecture of the proposed phishing URL detection system is designed to provide an efficient, scalable, and real-time solution for identifying malicious URLs. It follows a modular and layered approach, ensuring that each component performs a specific function while maintaining smooth communication with other components. The architecture integrates frontend interaction, backend processing, and machine learning prediction into a unified system.
At a high level, the system consists of three main layers: the user interface layer, the application layer, and the machine learning layer. These layers work together to process user

input, extract relevant features, and generate predictions. The separation of responsibilities across these layers improves system maintainability, scalability, and performance.

The first layer is the user interface layer, also known as the presentation layer. This layer is responsible for interacting with the user. It provides a simple and intuitive interface where users can input a URL and request analysis. The interface is designed to be user-friendly so that even non-technical users can easily use the system. Once the user enters a URL and submits the request, the frontend sends this data to the backend through an API call. The response received from the backend is then displayed clearly, indicating whether the URL is phishing or legitimate along with a confidence level.

The second layer is the application layer, which acts as the core processing unit of the system. This layer is implemented using a backend framework and is responsible for handling all incoming requests from the frontend. When a request is received, the backend performs several operations, including input validation, feature extraction, and communication with the machine learning model. The backend ensures that the input URL is properly formatted and valid before proceeding with further processing. This helps in preventing errors and improving system reliability. Within the application layer, the feature extraction module plays a crucial role. It converts the raw URL into a structured format that can be understood by the machine learning model. Various features are extracted from the URL, such as its length, number of dots, presence of special characters, usage of HTTPS, and occurrence of suspicious keywords. These features are transformed into numerical values and combined into a feature vector. This transformation is essential because machine learning models require numerical input for processing.

The third layer is the machine learning layer, which is responsible for classification. This layer contains the trained Random Forest model that has learned patterns from labeled data. When the feature vector is passed to the model, it analyzes the input based on previously learned patterns and generates a prediction. The output is typically a classification label indicating whether the URL is phishing or legitimate, along with a confidence score that reflects the certainty of the prediction.

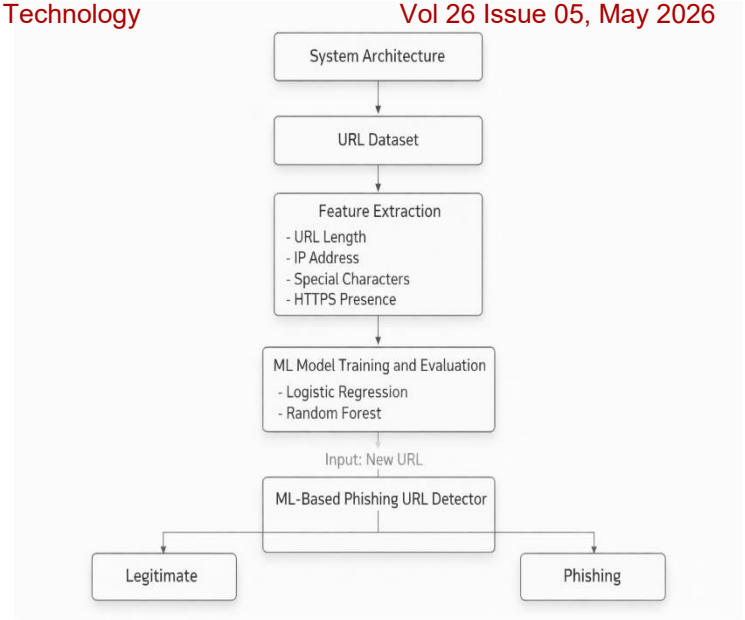

Fig. 2: System Architecture

The interaction between these layers follows a well-defined workflow. When a user submits a URL, the frontend sends a request to the backend API. The backend processes the request, extracts features, and forwards them to the machine learning model. The model then evaluates the features and returns a prediction. The backend formats this result and sends it back to the frontend, where it is displayed to the user. This entire process is designed to be fast and efficient, enabling real-time detection

.

One of the key design considerations in the system architecture is performance. Since the system is intended for real-time use, it must respond quickly to user requests. By focusing on URL-based feature extraction instead of analyzing full webpage content, the system significantly reduces processing time. Additionally, the use of an efficient machine learning model ensures that predictions are generated with minimal delay. Another important aspect of the architecture is scalability. The modular design allows the system to handle multiple requests simultaneously and can be extended to support additional features or functionalities. For example, the system can be integrated with browser extensions, email filtering systems, or enterprise security tools. The backend can also be deployed on cloud platforms to support a larger number of users.

Security is also considered in the architectural design. Input validation mechanisms are implemented to ensure that only valid URLs are processed. This prevents potential misuse of the system and enhances its reliability. Additionally, the system avoids storing sensitive user data, ensuring privacy and security.

The architecture also supports maintainability and future enhancements. Since each module is independent, updates can be made to individual components without affecting the entire system. For instance, the machine learning model can be retrained with new data and replaced without modifying the frontend or backend logic. Similarly, additional features can be incorporated into the feature extraction module to improve detection accuracy.

Despite its advantages, the architecture has certain limitations. The system primarily relies on URL-based features and does not analyze webpage content or user behavior. As a result, it may not detect highly sophisticated phishing attacks that closely resemble legitimate URLs. However, this limitation can be addressed in future work by integrating additional analysis techniques. In summary, the system architecture provides a well-structured and efficient framework for phishing URL detection. It integrates user interaction, backend processing, and machine learning prediction into a cohesive system. The modular design ensures scalability, maintainability, and real-time performance, making it suitable for practical deployment in various cybersecurity applications.

VII. RESULTS AND DISCUSSIONS

The results obtained from the implementation of the proposed phishing URL detection system demonstrate the effectiveness of using machine learning techniques for identifying malicious URLs. This section presents a detailed analysis of the model's performance, evaluation metrics, and observations derived from experimental testing. The discussion also highlights the strengths and limitations of the system based on the obtained results.

The model was trained and tested using a dataset containing both legitimate and phishing URLs. The dataset was divided into training and testing sets to ensure proper evaluation. The training data was used to build the model, while the testing data was used to evaluate its performance on unseen inputs. This approach helps in assessing how well the model generalizes to new data.

To evaluate the performance of the system, several standard metrics were used, including accuracy, precision, recall, and F1 score. These metrics provide a comprehensive understanding of the model's effectiveness in detecting phishing URLs. Accuracy measures the overall correctness of the model, indicating the percentage of correctly classified URLs. Precision reflects the proportion of correctly identified phishing URLs out of all URLs predicted as phishing. Recall measures the ability of the model to identify actual phishing URLs, while the F1 score provides a balance between precision and recall. The experimental results indicate that the model achieves a high level of accuracy, demonstrating its ability to correctly classify most URLs. The precision value is also high, which means that the system produces fewer false positives, reducing the chances of legitimate websites being incorrectly flagged as phishing. Similarly, the recall value is strong, indicating that the model successfully identifies a large proportion of phishing URLs. The balanced F1 score further confirms the reliability of the model in handling both phishing and legitimate classifications effectively.

A key tool used in evaluating the model is the confusion matrix. The confusion matrix provides a detailed breakdown of the model's predictions by categorizing them into true positives, true negatives, false positives, and false negatives. True positives represent phishing URLs correctly identified by the system, while true negatives represent legitimate URLs correctly classified. False positives occur when legitimate URLs are incorrectly marked as phishing, and false negatives occur when phishing URLs are misclassified as legitimate. An ideal model aims to maximize true positives and true negatives while minimizing false positives and false negatives. The confusion matrix analysis shows that the system performs well in distinguishing between phishing and legitimate URLs. The number of correctly classified instances is significantly higher than the number of misclassifications. This indicates that the selected features and the Random Forest model are effective in capturing patterns associated with phishing URLs.

Another important aspect of the results is the response time of the system. Since the model relies only on URL-based features, the processing time is relatively low. The system is able to generate predictions almost instantly after receiving user input. This makes it suitable for real-time applications where quick decision-making is essential. The lightweight nature of the model ensures that it can be deployed in environments with limited computational resources.

The graphical representation of performance metrics further illustrates the effectiveness of the model. The comparison of accuracy, precision, recall, and F1 score shows that all metrics are consistently high, indicating stable and reliable performance. The close values of precision and recall suggest that the model maintains a good balance between detecting phishing URLs and avoiding false alarms.

In addition to quantitative evaluation, qualitative analysis was also performed using sample URLs. The system was tested with various types of URLs, including simple legitimate websites, clearly suspicious URLs, and more complex cases that closely resemble real websites. The model successfully identified most phishing URLs, even when they included subtle variations designed to deceive users. This demonstrates the robustness of the feature extraction process and the effectiveness of the machine learning model.

When compared to traditional methods such as blacklist-based and heuristic approaches, the proposed system shows significant improvement in detection capability. Unlike blacklist systems, which can only detect known phishing URLs, the machine learning model is capable of identifying new and previously unseen phishing attempts. Similarly, unlike heuristic methods that rely on fixed rules, the model adapts to patterns learned from data, making it more flexible and accurate.

However, the results also reveal certain limitations. In some cases, highly sophisticated phishing URLs that closely mimic legitimate domains may not be detected accurately. This is primarily due to the reliance on URL-based features alone. Additionally, the performance of the model depends on the quality and diversity of the training dataset. If the dataset does not include a wide range of phishing patterns, the model may struggle to generalize effectively.

Another limitation is the possibility of false positives, where legitimate URLs are incorrectly classified as phishing. Although the number of such cases is relatively low, it can still affect user trust in the system. Continuous improvement through dataset expansion and feature enhancement can help reduce these errors.

Despite these limitations, the overall performance of the system is highly satisfactory. The model achieves a strong balance between accuracy and efficiency, making it suitable for practical deployment. The ability to detect phishing URLs in real-time without relying on external data sources is a significant advantage.

In conclusion, the results demonstrate that the proposed machine learning-based approach is effective in detecting phishing URLs. The use of structured feature extraction and a robust classification algorithm enable the system to achieve high accuracy and reliable performance. The discussion highlights both the strengths and limitations of the system, providing insights for future improvements and enhancements.

VIII CONCLUSION

This paper presented a machine learning-based approach for detecting phishing URLs using structural and lexical features. The proposed system leverages the Random Forest algorithm to effectively classify URLs as phishing or legitimate, achieving high accuracy and reliable performance. By focusing solely on URL-based features, the system ensures fast processing and reduced computational overhead, making it suitable for real-time applications.

The experimental results demonstrate that the model performs well across key evaluation metrics such as accuracy, precision, recall, and F1-score, indicating its robustness in identifying both known and previously unseen phishing attacks. Compared to traditional methods like blacklist and heuristic-based approaches, the proposed system offers improved adaptability and scalability.

Although the system shows strong performance, it has certain limitations, particularly in detecting highly sophisticated phishing URLs that closely mimic legitimate domains. Future work can focus on enhancing the model by incorporating additional features such as webpage content analysis, deep learning techniques, and real-time threat intelligence.

In conclusion, the proposed machine learning-based phishing detection system provides an effective, efficient, and scalable solution for improving cybersecurity and protecting users from evolving phishing threats.

VIII. REFERENCES

[1] R. Verma and N. Das, “What you see is not what you get: The impact of URL obfuscation on phishing detection,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2017, pp. 1–12.

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

[3] A. Le, A. Markopoulou, and M. Faloutsos, “PhishDef: URL names say it all,” in *Proceedings of IEEE INFOCOM*, 2011, pp. 191–195.

[4] S. Garera, N. Provos, M. Chew, and A. D. Rubin, “A framework for detection and measurement of phishing attacks,” in *Proceedings of the ACM Workshop on Recurring Malcode*, 2007, pp. 1–8.

[5] 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 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 1245–1254.

[6] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[7] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. Sebastopol, CA, USA: O’Reilly Media, 2019.

[8] UCI Machine Learning Repository, “Phishing websites dataset.” [Online]. Available: https://archive.ics.uci.edu

[9] PhishTank, “Phishing data and information.” [Online]. Available: https://www.phishtank.com

[10] Kaggle, “Phishing URL dataset.” [Online]. Available: https://www.kaggle.com

[11] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” in Proceedings of the 6th Conference on Email and Anti-Spam (CEAS), 2009, pp. 1–10.

[12] Y. Zhang, J. Hong, and L. Cranor, “CANTINA: A content-based approach to detecting phishing web sites,” in Proceedings of the 16th International World Wide Web Conference (WWW), 2007, pp. 639–648.

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

[14] G. Xiang, J. Hong, C. Rose, and L. Cranor, “CANTINA+: A feature-rich machine learning framework for detecting phishing web sites,” ACM Transactions on Information and System Security, vol. 14, no. 2, pp. 1–28, 2011.

[15] A. Blum, B. Wardman, T. Solorio, and G. Warner, “Lexical feature based phishing URL detection using online learning,” in Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, 2010, pp. 54–60.

[16] R. Mohammad, F. Thabtah, and L. McCluskey, “Predicting phishing websites based on self-structuring neural network,” Neural Computing and Applications, vol. 25, no. 2, pp. 443–458, 2014.

[17] J. 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.

[18] M. Aburrous, M. Hossain, F. Thabtah, and K. Dahal, “Intelligent phishing detection system using fuzzy logic,” Expert Systems with Applications, vol. 37, no. 12, pp. 7913–7921, 2010.

[19] N. Abdelhamid, A. Ayesh, and F. Thabtah, “Phishing detection based associative classification data mining,” Expert Systems with Applications, vol. 41, no. 13, pp. 5948–5959, 2014.

[20] K. L. Chiew, K. S. C. Yong, and C. L. Tan, “A survey of phishing attacks: Their types, vectors and technical approaches,” Expert Systems with Applications, vol. 106, pp. 1–20, 2018.