International Journal of Research Publication and Reviews
Journal homepage: www.ijrpr.com ISSN 2582-7421

PHISHGUARD X-SMART ANALYSIS FOR PHISHING WEBSITE DETECTION USING URL

Mr. R.Makendran [a] , M.Pasumathi [b] ,P.Swathi [c] ,S.Swathi [d] ,S.Yasika [e]

[a] Assistant Professor, Department Of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, India.
[b, c, d, e] Student, Department Of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, India.

ABSTRACT:

Phishing websites continue to pose a significant threat to online security by tricking users into revealing confidential information such as usernames, passwords, and financial data. Traditional detection techniques like blacklists and rule-based heuristics often fall short, especially against new and sophisticated phishing attacks. To overcome these limitations, this project introduces a smart, machine learning–based approach that relies solely on URL analysis to detect phishing websites. The system is structured into three core modules: data collection, feature extraction, and model training. Publicly available datasets containing both legitimate and phishing URLs are used to extract a wide range of URL-based features, including URL length, number of special characters, use of IP addresses, presence of suspicious keywords, domain name similarity, and domain age. Lexical and host-based features are also considered, using natural language processing techniques to analyze the structure and semantics of URLs. These features are fed into machine learning models such as Random Forest and deep learning models like LSTM, which have demonstrated high accuracy, precision, and recall in classifying URLs as phishing or legitimate. Furthermore, the system includes a lightweight, user-friendly web interface developed using Flask, enabling real-time URL scanning and threat detection. One of the key strengths of this solution is its ability to identify zero-day phishing attacks by learning from URL patterns rather than relying on known blacklists. By offering a robust, scalable, and efficient tool for phishing detection, this project significantly enhances user safety and helps prevent identity theft and financial fraud on the internet.

Keywords: Phishing detection, Machine learning, URL analysis, Smart web security, Classification algorithms.

URL	— Universal Resource Locator
RF	— Random Forest
LSTM	— Long-Short Term Memory
SSL	—Security Socket Layer

INTRODUCTION

With the growing dependence on online services like banking, shopping, and social media, phishing attacks have become a serious cybersecurity concern. These attacks mimic legitimate websites using deceptive URLs to steal user information. Traditional defenses such as blacklisting and signature-based systems are increasingly ineffective, especially against zero-day phishing sites that haven't been reported yet. Alternative methods like heuristic and content-based detection offer some improvement but come with limitations. Heuristic techniques often misclassify legitimate sites due to rigid rules, while content-based and visual analysis approaches are resource-heavy and vulnerable to obfuscation tactics. These systems also lack adaptability, making them easy to bypass with evolving phishing strategies. To address these gaps, machine learning offers a smarter approach. By analyzing URL features such as domain age, SSL status, and symbol patterns, ML models can detect phishing attempts with high accuracy and minimal computational cost. PhishGuardX adopts this strategy, using lightweight and scalable URL-based classification to provide fast, adaptive, and effective phishing detection.

LITERATURE SURVEY

2.1 TITLE: URL-BASED PHISHING DETECTION USING RANDOM FOREST AND SVM

AUTHOR: **AHD AL-QASMI ET AL. (2021) DESCRIPTION**: This paper presents a comparative study of Random Forest and Support Vector Machine (SVM) classifiers for phishing detection, with an emphasis on real-time deployment scenarios. The dataset used is from 2020 and includes a wide range of phishing and legitimate URLs. Unlike traditional methods that use content-based or visual features, this research leverages URL lexical features and domain-based attributes, making it faster and less computationally expensive. Lexical features include URL length, number of dots, use of

symbols, and the presence of keywords like "login" or "verify". Domain-based features include domain age, use of HTTPS, and registration information. Random Forest achieved an impressive accuracy of 99.4%, outperforming SVM in both precision and recall.

2.2 TITLE: LIGHTWEIGHT PHISHING DETECTION USING URL FEATURES AND LSTM

AUTHOR: **M. KORKMAZ, O. K. SAHINGOZ, B. DIRI (2020) DESCRIPTION:** This study shifts focus from traditional machine learning models to deep learning, particularly Long Short-Term Memory (LSTM) networks, to identify phishing URLs. The innovation lies in treating the URL as a sequence of characters or tokens, enabling the model to learn sequential patterns and relationships that traditional models might miss. The dataset used includes thousands of phishing and legitimate URLs. The URLs are tokenized and padded to a uniform length to make them compatible 12 with LSTM input layers. The study compares the LSTM model's performance against classical algorithms like Random Forest and Decision Tree, and demonstrates that LSTM performs better in adaptability and accuracy, especially in dynamic phishing attacks. The approach is lightweight, making it suitable for systems with limited processing capacity. Moreover, the model eliminates the need for handcrafted feature extraction, relying instead on the LSTM's ability to learn complex patterns directly from raw input, thereby reducing human effort and increasing adaptability to unseen data.

SYSTEM STUDY

3.1. EXISTING SYSTEM

The existing phishing detection systems primarily rely on traditional approaches such as blacklists, heuristic rules, and user reports. Blacklist-based methods maintain databases of known phishing URLs and block access to them. While simple to implement, these systems are highly reactive and fail to detect zero-day phishing attacks—new phishing sites that have not yet been reported or listed. Heuristic-based systems, on the other hand, use manually defined rules to identify suspicious patterns in URLs or webpage content. However, these rules can be easily bypassed by attackers using slightly modified techniques. Additionally, many web browsers and email clients integrate basic phishing protection using these methods, but they offer limited coverage and struggle to keep up with the increasing sophistication of phishing tactics. Some systems also rely on analyzing the webpage content, which requires loading the entire page, making the process slower and less secure, especially for real-time detection. Overall, the existing systems are ineffective in detecting new or obfuscated phishing websites, lack adaptability, and cannot generalize well to evolving threats. These limitations highlight the need for a more intelligent and proactive solution, such as the machine learning–based system proposed in this project.

Disadvantages:

• While the proposed phishing detection system using machine learning offers significant improvements over traditional methods, it also comes with certain limitations. One major disadvantage is its dependence on the quality and size of the training dataset. If the dataset is not diverse or updated regularly, the model may fail to detect newly emerging phishing patterns or may produce false positives. Additionally, the system relies solely on URL features without analyzing the full website content, which might limit its ability to catch sophisticated phishing websites that appear legitimate at the URL level but are malicious at the content level.

• Another limitation is the computational cost associated with training complex models like LSTM, especially when dealing with large datasets. Although these models offer higher accuracy, they may not be suitable for deployment in low-resource environments. Also, despite the real-time detection capability, processing time can slightly increase when multiple models or deep learning techniques are used simultaneously.

• Lastly, adversarial attackers may attempt to craft URLs that bypass machine learning models by mimicking legitimate patterns. As phishing techniques evolve, the model must be retrained regularly to maintain its effectiveness, which adds to the system's maintenance overhead.

3.2. PROPOSED SYSTEM

To overcome the limitations of traditional phishing detection methods, the proposed system introduces an intelligent, machine learning–based approach that identifies phishing websites by analyzing the structural and lexical characteristics of URLs. Instead of relying on outdated blacklists or manually crafted rules, this system uses data-driven models trained on a wide variety of phishing and legitimate URLs to detect threats more accurately and proactively. The system involves three key stages: data collection, feature extraction, and model training and prediction. A large dataset of URLs is gathered from public repositories and labeled accordingly. From each URL, critical features are extracted—such as length, presence of symbols, number of subdomains, use of IP addresses, domain registration details, and suspicious keywords. These features are then used to train machine learning models such as Decision Trees, Random Forests, and deep learning models like LSTM to classify URLs as phishing or legitimate. The model is integrated into a lightweight Flask-based web application that allows users to input a URL and receive an instant prediction, making the system easy to use and deploy in real-world scenarios. It is designed to be fast, scalable, and capable of identifying zero-day phishing attacks, even those that do not appear in blacklists or reports. By learning from patterns within the URLs themselves, the proposed system delivers a proactive, efficient, and modern solution to phishing detection—enhancing cybersecurity protection for both individuals and organizations.

Advantages:

- A Phishing Detection URL Website project offers

numerous advantages, primarily in enhancing cybersecurity.

- By detecting phishing URLs, it helps prevent users

from accessing fraudulent websites designed to steal sensitive information like passwords and credit card details. The website provides real-time protection, alerting users to potential threats before they engage with harmful sites.

- This project also serves an educational purpose,

raising awareness about phishing scams and promoting safer browsing habits.

- Additionally, it collects valuable data on phishing

trends, which can aid in improving detection algorithms and cybersecurity research.

- Automation is another key benefit, allowing for

efficient and scalable scanning of a large number of URLs, saving both time and effort compared to manual checks.

- The project can be used by individuals, businesses,

and organizations to safeguard against phishing attacks, enhancing both personal and corporate security. If open-sourced, it could contribute to the global cybersecurity community, fostering collaboration and innovation in phishing detection.

- Furthermore, by helping businesses meet compliance

standards such as GDPR or HIPAA, the website supports privacy protection and strengthens legal safeguards. Overall, this project plays a crucial role in protecting users and organizations from the growing threat of online fraud.

METHODOLOGY

In PhishGuardX, both Random Forest (RF) and Long Short-Term Memory (LSTM) models play a crucial role in accurately detecting phishing URLs. The Random Forest model is an ensemble learning technique that builds multiple decision trees, each trained on a random subset of features from the dataset. During prediction, each tree votes on whether a URL is phishing or legitimate, and the majority vote is used as the final decision. RF is effective in handling highdimensional data, capturing complex relationships between URL features, such as domain age, SSL certificate presence, and the number of subdomains. Its ability to reduce overfitting by averaging multiple trees makes it robust, and it is particularly useful for classifying URLs based on predefined features, ensuring high accuracy even with noisy data. On the other hand, the LSTM model, a type of Recurrent Neural Network (RNN), is used to capture sequential patterns in URLs that evolve over time. LSTM is designed to remember long-term dependencies, which makes it suitable for analyzing URL patterns where order matters, such as token sequences or the arrangement of certain keywords. In PhishGuardX, LSTM is employed to detect more subtle and dynamic phishing tactics, such as obfuscated or polymorphic URLs that change frequently. By processing the URL as a sequence of characters or substrings, the LSTM can detect irregular patterns that static models like RF might miss. Together, RF and LSTM provide a powerful hybrid approach, combining the strengths of both tree-based and deep learning methods, ensuring accurate detection across a broad spectrum of phishing URLs, including both known and novel attack strategies.

MODULES IMPLEMENTATION

5.1 LIST OF MODULES

- Data Collection Module
- Features Extracting Module
- Training Hybrid Module
- Deployment And Detection

5.2 MODULES DESCRIPTION

5.2.1 DATA COLLECTION MODULE

At the start of our project, our primary goal was t0

gather a reliable and well-structured dataset containing both phishing and legitimate URLs. For this, we turned to a publicly available repository on GitHub called "Phishing Website Detection" by MVS Chamanth. The dataset, which can be accessed here, was an ideal choice as it comes with pre-extracted features in CSV format, making it immediately usable for machine learning applications. The dataset includes two main files: legitimate-urls.csv and phishingurls.csv. Both files follow the standard CSV format, making them easy to read and manipulate using Python libraries such as pandas and NumPy. Each row represents a URL, accompanied by a range of extracted features that help determine whether the URL is safe or malicious. One of the key advantages of this dataset is its balanced nature, with an equal number of phishing and legitimate URLs. This balance is crucial in preventing our machine learning model from becoming biased toward either class. The dataset includes several important features designed to help the model 25 accurately differentiate between phishing and legitimate URLs. For example, it checks for the presence of special characters like '@', which are often used in phishing links, and whether the URL uses an IP address instead of a domain name. Other features include the structure of the URL, the presence of hyphens in domain names, the protocol used (HTTP vs. HTTPS), and how many redirections the URL goes through. These indicators collectively provide a solid basis for identifying potential phishing threats. By starting with this rich and balanced dataset, we've laid a strong groundwork for the next steps of our project—such as preprocessing the data, training the model, evaluating its performance, and eventually deploying a smart, URL-based phishing detection system.

5.2.2 FEATURE EXTRACTING MODULE

In this phase of the project, we turned our attention to the most critical part of building an intelligent detection system — extracting meaningful features from each URL. The goal was to transform raw, unstructured URLs into a structured format that machine learning models can effectively learn from. By examining the structure, content, and technical components of each URL, we aimed to uncover patterns and warning signs typically associated with phishing attacks.We started by looking at the basic structure of each URL. Features like the length of the domain name, total URL length, and the number of dots or hyphens were captured. These might seem simple, but phishing URLs often rely on overly long or oddly structured domains to trick users. The presence of subdomains was also noted, as phishing links frequently use multiple subdomains to imitate trusted websites.Some red flags are easier to spot — for instance, URLs containing the '@' symbol, which is often used to deceive users by hiding the actual domain. Similarly, IP addresses used in place of proper domain names were marked as suspicious, since legitimate websites rarely use raw IPs in their links.

5.2.3 TRAININGD HYBRID MODULE

To achieve this, we used a hybrid approach involving two powerful algorithms: the Random Forest Classifier and LSTM (Long Short-Term Memory) neural networks. The Random Forest, known for its high accuracy and resistance to overfitting, helps provide strong baseline predictions using structured features. Meanwhile, LSTM models — typically used for sequence prediction tasks — bring in the ability to understand patterns within the URL strings themselves. This combination allowed us to capture both high-level structural indicators and deeper sequential relationships. We trained the model on a balanced dataset containing an equal number of phishing and legitimate URLs, ensuring fair learning without bias. The dataset was split into 80% for training and 20% for testing, allowing the model to learn from a broad sample while also being evaluated on unseen data to test its performance. Before training, we performed essential data preprocessing. This included cleaning and normalizing URLs, encoding labels, and extracting structured features such as the length of the URL, number of special characters, domain age, and more. These steps ensured the data was consistent, informative, and ready for machine learning. For the Random Forest, we built an ensemble of 100 decision trees. Using Grid Search, we fine-tuned the model's parameters, especially tree depth, to boost accuracy. On the other hand, for the LSTM model, URLs were first tokenized and then transformed into padded sequencesso that the network could process them effectively. The model was trained over multiple epochs, helping it capture subtle and complex phishing patterns hidden within the sequences. By the end of this module, we had a hybrid model capable of making informed predictions.

5.2.4 DEPLOYMENT AND DETECTION MODULE

In this final phase of our project, we bring everything together by deploying our phishing detection model in a real-world setting using a Flask-based web application. The main goal here is to offer real-time URL detection that empowers users to verify the safety of links before they click, reducing the risk of falling prey to phishing attacks. The system is designed with user convenience and protection in mind. Through a simple web interface, users can input any URL they wish to check. Once the URL is submitted, the backend — powered by our trained machine learning model — instantly analyzes it and provides a prediction: phishing or legitimate. This ensures that users get quick, accurate feedback on the safety of the URL in question. To further strengthen user security, the interface includes protective features. If a URL is classified as phishing, access is immediately blocked, and the system prevents users from proceeding. Even if a phishing link is re-entered, the model intervenes again, maintaining a secure browsing environment. This proactive blocking mechanism builds user trust and offers an added layer of defense against malicious sites. The entire deployment is supported by a well-integrated tech stack. The backend model runs on Flask, which handles URL requests and serves model predictions efficiently. The frontend is developed using HTML, CSS, and JavaScript, ensuring a responsive and clean design for smooth user interaction. We also implemented fallback mechanisms to handle cases where the model might fail or temporarily become unavailable, ensuring that the system continues to function reliably. Overall, this module successfully bridges the gap between research and real-world application. By integrating the model into a functional and secure web platform, we demonstrate how machine learning can be transformed into a practical tool for 28 everyday cybersecurity, protecting users from threats in real-time.

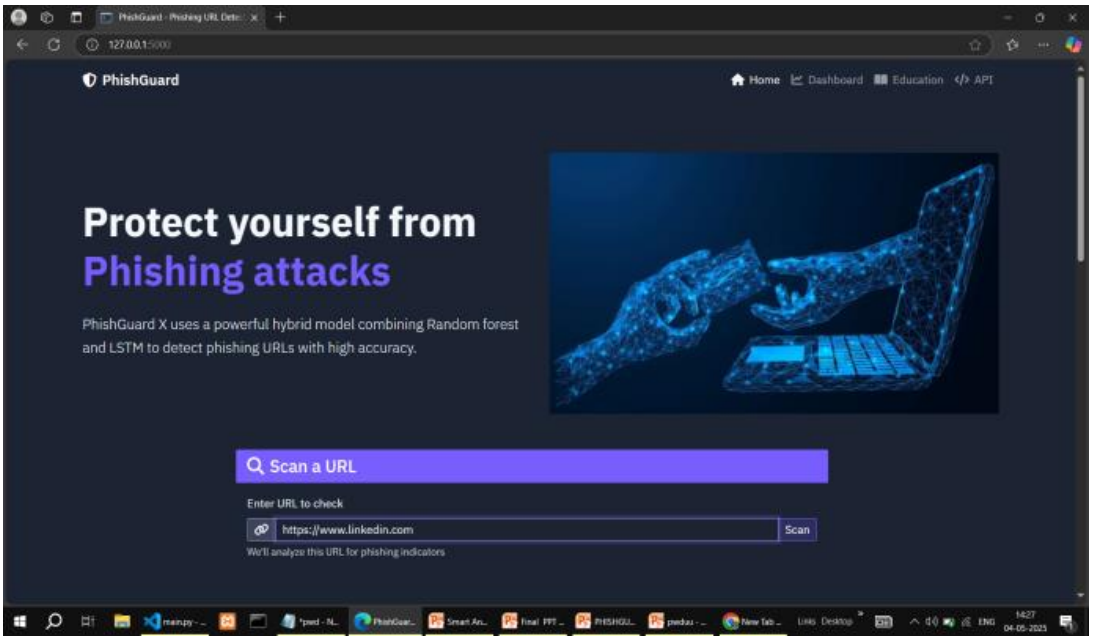

Figure 5.1.1: Home page & URL Enter page

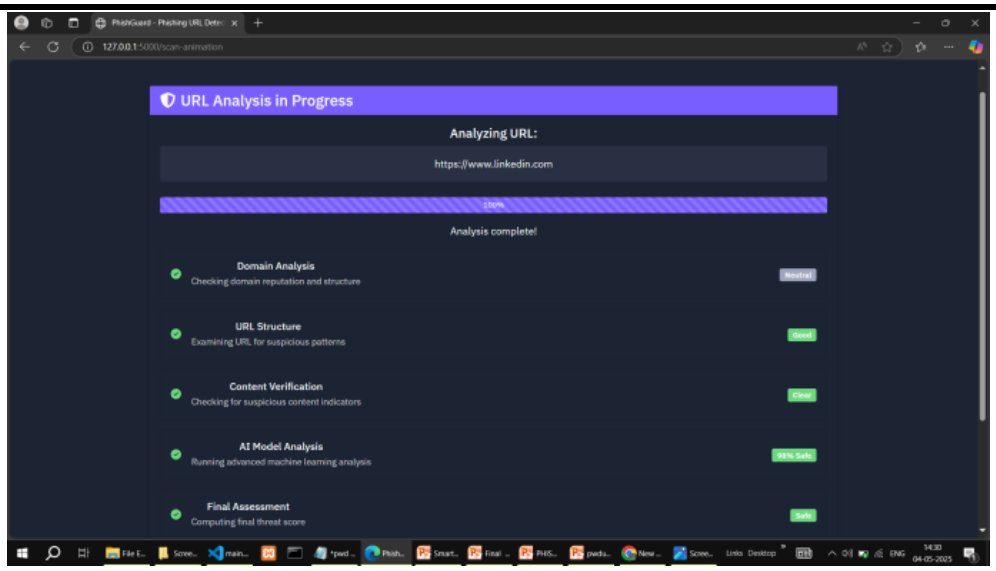

Figure 5.1.2: URL Analysing page

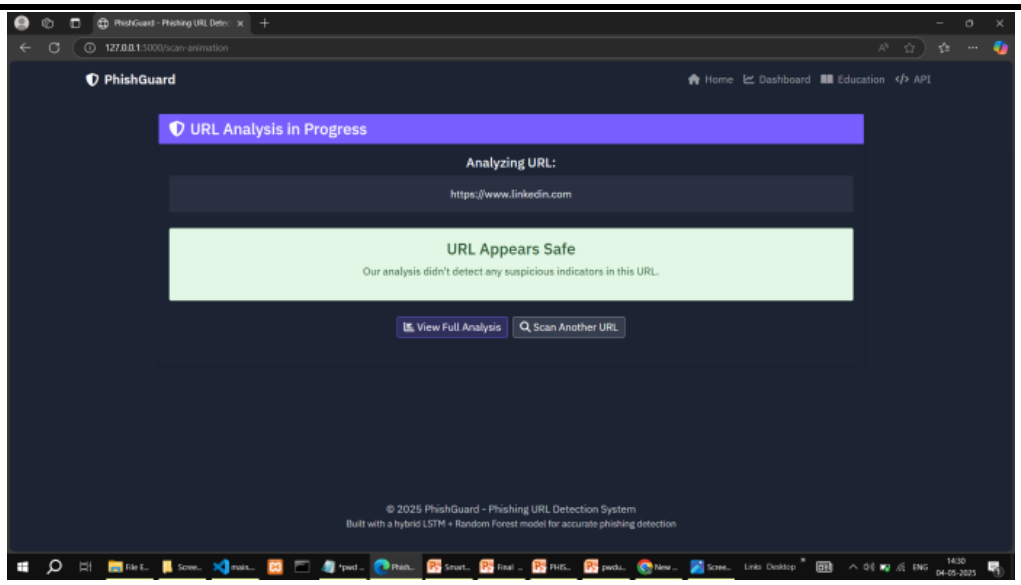

Figure 5.1.3: Legal URL

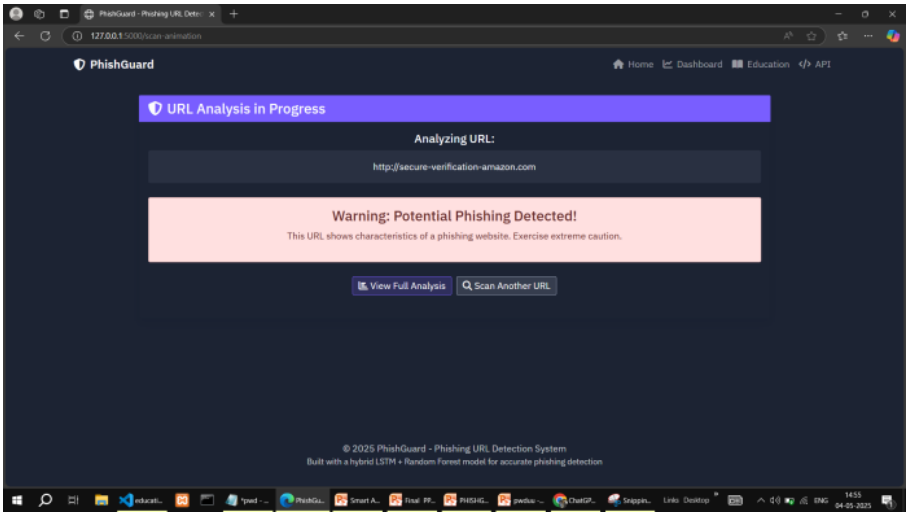

Figure 5.1.4: Fake URL

Figure 5.1.5:Overall Dashboard

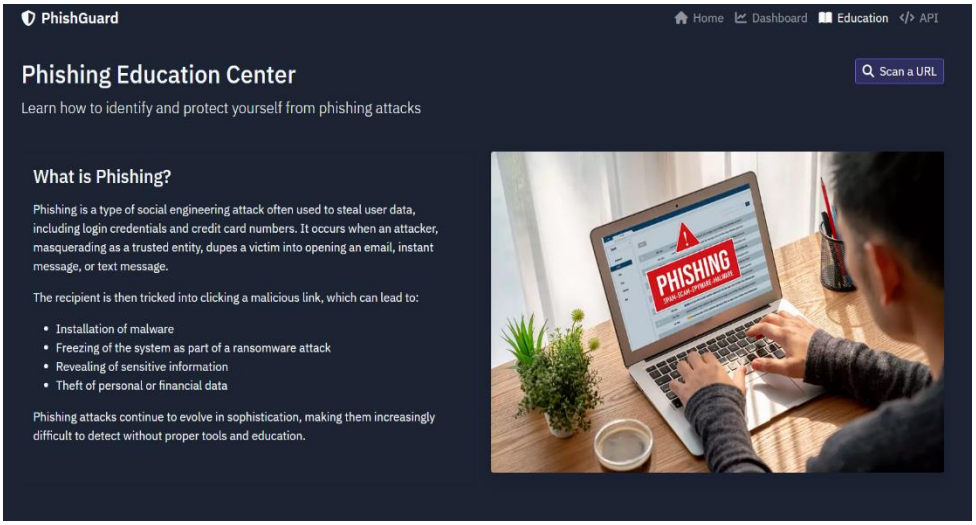

Figure 5.1.6:Education Page

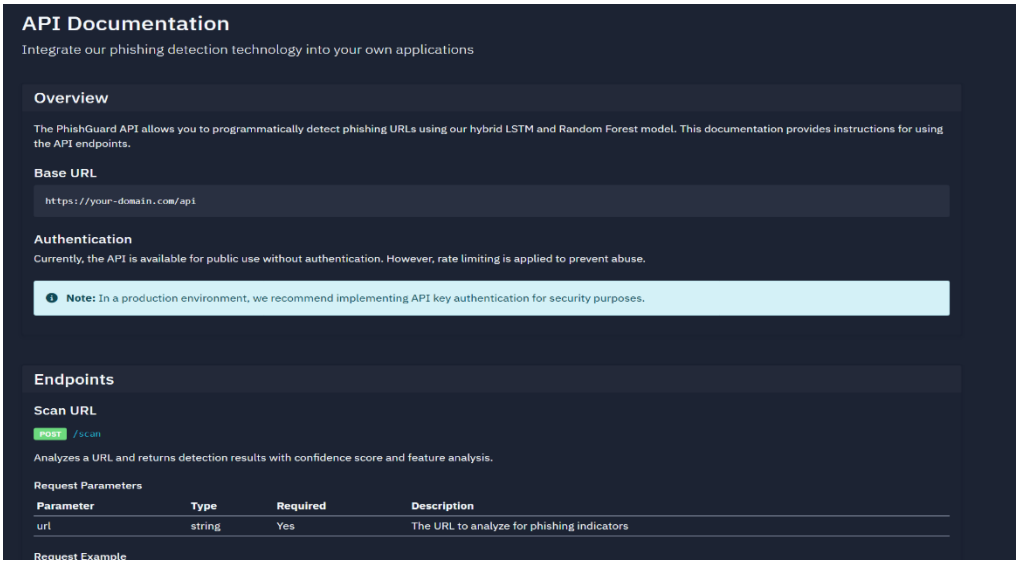

Figure 5.1.7: API Documentation page

SYSTEM ARCHITECTURE

The following diagram represents the architecture of PhishGuard X, an intelligent phishing detection system that analyzes URLs to determine their legitimacy. It starts with an Input URL, which can be entered by either a legitimate User or a malicious Attacker. This URL is passed into the Feature Extraction Engine, where critical characteristics such as URL length, special characters, and the use of IP addresses are analyzed. These extracted features are sent to a Random Forest Classifier, which evaluates the risk based on predefined phishing indicators. Simultaneously, the raw URL is fed into an LSTM Sequence Analyzer, a deep learning model that examines patterns and sequences typical of phishing attempts. The predictions from both the Random Forest and LSTM models are passed into a Fusion Layer, which intelligently merges their outputs for more accurate results. This hybrid approach strengthens the detection mechanism by combining the advantages of both traditional machine learning and deep learning. The final Output from the system clearly classifies the URL as either Phishing or Legitimate. The flow is streamlined, ensuring real-time evaluation while maintaining a high level of accuracy. This architecture is robust, scalable, and suitable for browser extensions or network-level monitoring. The use of both statistical features and sequence analysis allows the system to detect even sophisticated phishing schemes. By integrating multiple detection strategies, PhishGuard X reduces false positives and enhances user safety. The modular design also supports future upgrades and integration with reporting tools. Overall, the diagram captures a comprehensive and efficient pipeline for phishing detection using AI.

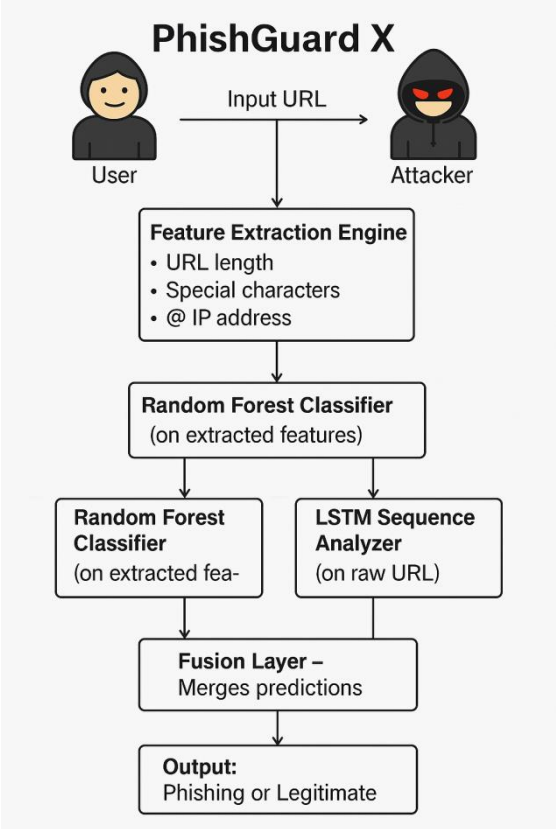

Figure 6.1: System architecture

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

Phishing remains a major cybersecurity threat, growing more sophisticated and dynamic over time. Traditional methods like blacklists and heuristics are no longer sufficient, highlighting the urgent need for intelligent and adaptable solutions. PhishGuardX addresses this challenge with a machine learning-based system that analyzes URL characteristics to effectively distinguish between phishing and legitimate sites. By utilizing lexical, domain-based, and structural features, and employing classifiers such as Decision Trees, Random Forests, and SVMs, PhishGuardX delivers high detection accuracy without relying on resourceintensive content analysis. The system's ability to classify URLs in real-time without fetching full web content enhances its speed and practicality. Experimental results show accuracy rates above 96%, particularly with Random Forest models, making it a reliable defense tool. Its modular architecture supports future enhancements like deep learning integration and real-time deployment in browsers or security gateways. PhishGuardX not only overcomes current limitations but also sets a solid foundation for next-generation anti-phishing technology. With continued evolution through real-time processing, adaptive learning, and collaborative data sharing, it holds strong potential as a critical cybersecurity asset.

FUTURE ENHANCEMENTS

- PhishGuardX currently achieves strong results using

classical machine learning and handcrafted URL features, but phishing threats are rapidly evolving. Attackers now use obfuscation, short-lived domains, and social engineering, requiring detection systems to adapt quickly.

- A key future direction is integrating deep learning

models like CNNs, RNNs, and LSTMs, which can learn complex patterns from raw data without manual feature engineering. Real-time detection is also essential, enabling integration into browsers and email gateways by using lightweight, fast models. Enhancing accuracy with contextual data—such as domain registration, server location, SSL status, and user behavior—can offer a more complete threat picture.

- For example, traffic spikes to new domains could

signal active phishing campaigns. Continuous learning through semi-supervised or online learning will keep models up-to-date without full retraining. Feedback from user reports and browser telemetry can refine performance and reduce false positives.

- To boost adoption, PhishGuardX should evolve into

an API-based service or browser plugin. Open-sourcing the tool would encourage community contributions and drive innovation. In essence, the future of PhishGuardX lies in deeper intelligence, real-time adaptability, contextual awareness, and seamless integration to stay ahead of phishing threats.

REFERENCES

[1] Sonam Malviya (2023) An Artificial Neural Network (ANN)- based model using 14 URL-based features like URL length, IP usage, and hyphens.

[2] Ahammad et al. (2023) Tested classifiers like LightGBM, Random Forest, SVM, and Decision Tree using lexical and domain features.

[3].Marwa A.H. Qasim and Nahla A. Flayh (2024) Reviewed supervised learning models like SVM, Decision Trees, and Random Forest, emphasizing features like SSL presence and anchor tag behavior.

[4] S. Arvind Anwekar and V. Agrawal (2023) Developed an ensemble model combining Random Forest, Decision Tree, and SVM, 97% accuracy using domain age and website rank.

[5] Choudhary et al. (2023) Used a PCA-based hybrid model integrated with SVM and Random Forest to reduce feature space and maintain accuracy.

[6] L. Tang and Q. Mahmoud (2023) Implemented an RNN-GRU deep learning model in a browser plug-in, surpassing traditional models.

[7] Zhang et al. (2023) Incorporated semantic features and language-specific analysis, particularly for Chinese content, using a fusion of Bagging, AdaBoost, and SMO classifiers.

[8] Md Sultanul Islam Ovi et al. (2024) PhishGuard: A Multi-Layered Ensemble Model for Optimal Phishing Website Detection.

[9] Ammar Odeh et al. (2023) Comparative Study of CatBoost, XGBoost, and LightGBM for Enhanced URL Phishing Detection: A Performance Assessment.

[10] Rekha Pal et al. (2023) Phishing Detection: A Hybrid Model with Feature Selection and Machine Learning Techniques.