Enhanced E-Banking Security System for Phishing Detection Using Machine Learning and Privacy Preservation

Orokor Favour Wenebunwo
Department of Computer Science, Rivers State University, Rivers State
orokorjagz@gmail.com

DOI: 10.56201/ijcsmt.v11.no2.2025.pg73.84

Abstract

E-banking systems have revolutionized financial transactions, but they are increasingly vulnerable to phishing attacks that compromise sensitive user data. Phishing detection in e-banking presents several challenges, including the ability to accurately identify deceptive URLs, the presence of imbalanced datasets where phishing attempts are significantly fewer than legitimate activities, and the need for privacy-preserving mechanisms that protect user data during the detection process. Addressing these issues requires the development of advanced machine learning algorithms that can detect phishing attacks with high accuracy while preserving user privacy. This dissertation presents the development of a phishing detection system for e-banking, addressing key challenges such as detecting sophisticated phishing attempts while ensuring data privacy. The system was built using Object-Oriented Design (OOD) methodology, with Python as the programming language. Utilizing the Random Forest Classifier (RFC) and privacy preservation techniques, the model enhances e-banking security. Exploratory Data Analysis (EDA) revealed an imbalanced dataset, which was mitigated through Random OverSampling. Key features were identified using feature importance ranking and correlation matrices, leading to a highly accurate RFC model. To protect sensitive data, differential privacy was incorporated by adding Laplace noise during both training and deployment. The model achieved an accuracy of 98.98%, outperforming other systems, and was deployed via a Flask web application with a user-friendly interface for phishing URL detection. Comparisons with existing models showed the system's superior performance across key metrics.

Keywords: E-banking Security; Phishing Detection; Machine Learning; Privacy Preservation.

1. INTRODUCTION

Enhancing e-banking security is crucial in today's digital age to protect users' sensitive information and ensure the integrity of online transactions. Various studies emphasize the importance of robust security frameworks in e-banking systems to address authentication, confidentiality, integrity, and non-repudiation issues (Sarjiyus et al., 2019). By improving security measures, such as encryption, banks can enhance customers' secure access to e-banking services, leading to increased usage and satisfaction.

In the context of e-banking, federated learning offers significant advantages in terms of security and privacy. It allows for the training of AI models over distributed clients while providing secure privacy information to the data owners ("Centralized Machine Learning Versus Federated Averaging: A Comparison using MNIST Dataset", 2022). Moreover, federated learning enables users to leverage machine learning models without compromising their data, thus ensuring the confidentiality and integrity of sensitive banking information (Xie et al., 2020). By adopting federated learning, e-banking institutions can enhance their security measures by preventing unauthorized access to customer data and improving overall data protection protocols.

The primary statement of this problem in E-banking security remains a persistent and evolving challenge in the digital era. Cybercriminals continuously exploit vulnerabilities in the online banking ecosystem, deploying sophisticated tactics to deceive users and gain unauthorized access to sensitive financial information. Despite the implementation of various security measures, the prevalence and impact of e-banking phishing attacks persist, demanding a deeper understanding of the problem to develop effective countermeasures. Key challenges are as follows: User vulnerability, Technological Gaps, and the ever-evolving Tactics.

Furthermore, Machine Learning can play a crucial role in enhancing e-banking security by enabling model training without the need for data sharing (Xu, 2021). Additionally, Machine learning can help e-banking institutions address the challenges of data privacy and security by providing secure aggregation solutions that protect sensitive information during the model training process (Mansouri et al., 2023). One of the foundational approaches to phishing URL detection is the use of blacklist-based techniques. These methods involve maintaining a list of known phishing URLs and blocking access to them. While effective, this approach suffers from limitations, particularly in terms of its reliance on the quality and timeliness of the blacklist, which can lead to false negatives if new phishing URLs are not promptly added (Chang-Hoon et al., 2015; Ali, 2017).

Alhamad et al. (2020) reduced and prevented cyber-attacks by differentiating between genuine and deceptive websites using the C4.5 data mining method. The selection of C4.5, renowned for its classification prowess, allows for the detection and classification of authentic and fraudulent websites. The work used the data analysis tool WEKA to implement the algorithm and tackle phishing concerns in electronic banking systems. The experiment utilized a dataset consisting of 32 variables, leading to a remarkable accuracy rate of 98.11%. The algorithm's high accuracy demonstrates its efficacy in categorising websites. Furthermore, the algorithm's low prevented rate of 1.89% emphasizes its minimum number of wrong classifications, showcasing its potential as a strong solution for improving cybersecurity in electronic banking environments.

Nuha et al. (2021) seeks to determine the primary elements that contribute to the increase in fraudulent activities, specifically examining the relationship between a lack of awareness and vulnerability to fraud. The study utilizes a blend of primary and secondary data analytics to accomplish its objectives. The findings demonstrate a strong correlation between a lack of knowledge and the susceptibility to fraud, since a substantial proportion (76%) of individuals have no comprehension of e-banking and mobile banking fraud. Moreover, the results indicate that a significant majority (86.3%) of individuals who fell victim to fraud in these areas were not cognizant of the particular form of fraud they encountered. In contrast, a mere 13.7% of victims possessed prior awareness of the fraudulent activity. The research highlights that a lack of information and awareness are significant variables that contribute to this form of fraud.

2. RESEARCH METHOD

Research methodology uses various approaches, techniques, schemes, styles, procedures, and algorithms applied in research. It is a systematic approach applied to get the problem solved. It is the process that spells out the method, by which a researcher describes, elaborates, and forecasts phenomena. It is the study of the procedure and process by which desired information and understanding is acquired. It targets to offer research the needed working map It is the proficiency or procedure applied to identify, select, process, and analyze information about a subject in a research document. It deals with how data are generated and analyzed. It is the systematic, theoretical analysis of methods applied to a topic of study. Quite several research methodologies are employed in carrying out research, in order to accomplish the objectives of this research and come out with a software product that will adequately resolve the essence of this research to end users. An adaptation of constructive methodology was made and deployed in various areas of research.

2.1 Exploratory Data Analysis (EDA)

The EDA on the dataset involved the utilization of the pandas, seaborn, and matplotlib libraries. The EDA was performed in order to gain a comprehensive understanding of the dataset prior to training the ML algorithm. The analysis phases involve visualizing the dataset if there exist missing values in the dataset. Figure 1 shows a visualized image showing that the dataset contains no missing values. The next plot was check the number of instances in the target column of dataset. This can be depicted, the number of instances in the dataset is not of equal size. This signifies that the dataset is imbalanced. Which if not resolved, it will cause the performance of the model to be biased. In order to address the issue of data imbalance, it is necessary to implement random oversampling**.**

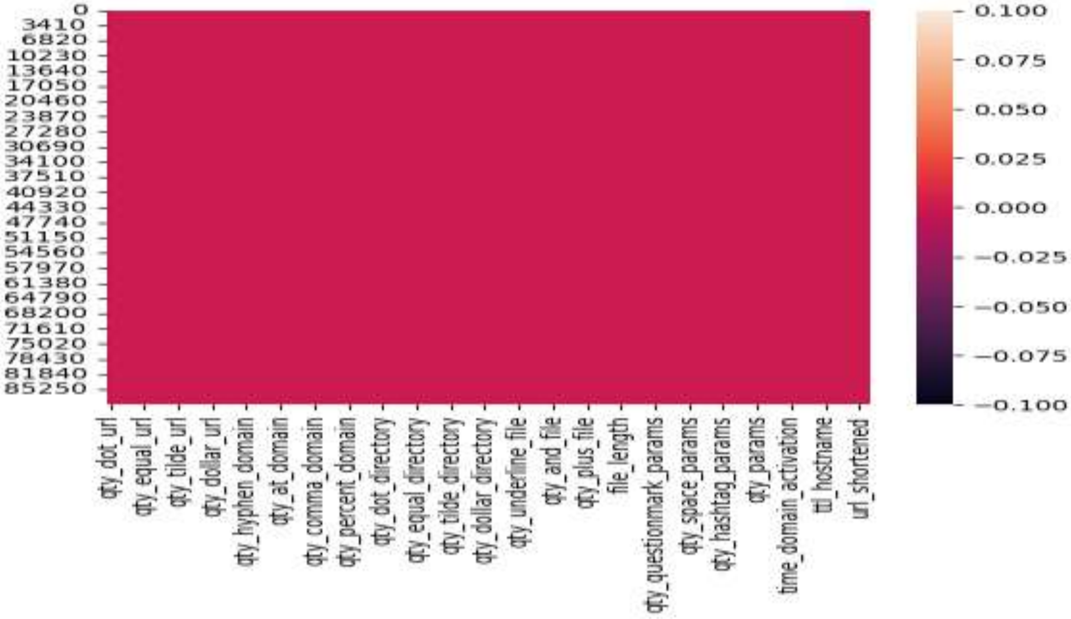


Figure 1: Visualized Image of Cleaned Data

2.2 Implementation of Random Forest Algorithm With Privacy Preservation For Enhancing E-banking Security

The implementation of the Random Forest classifier for detecting phishing URLs involves several key steps. First, the dataset comprising URLs and their labels (phishing or legitimate) is vectorized using the TF-IDF (Term Frequency-Inverse Document Frequency) method to transform the text data into numerical features suitable for machine learning. The vectorized data is then processed using a privacy-preserving technique to ensure sensitive customer information is protected during both training and deployment phases. Specifically, Laplace noise is added to the features to provide differential privacy, which involves perturbing the data in a manner that maintains overall data utility while masking individual data points. This privacy-preserved data is then subjected to Random Oversampling to balance the class distribution, followed by splitting the data into training and testing sets.The core of the classification process is the Random Forest algorithm, an ensemble learning method that builds multiple decision trees and merges their predictions to achieve higher accuracy and robustness. Each decision tree in the forest is trained on a random subset of the training data, with features randomly selected at each split point, reducing overfitting and improving generalization. The trained model is then evaluated on the test set, with performance measured using metrics such as accuracy. Finally, the trained Random Forest model and the TF-IDF vectorizer are saved for deployment, where they can be used to predict the legitimacy of new URLs. The system's results, including privacy-preserved feature values and visualizations, are displayed in a user-friendly web interface developed with Flask and styled with Bootstrap to enhance interactivity and aesthetics.

3. PROPOSED PHISHING DETECTION MODEL

The proposed system presents a novel and promising approach to combat the ever-increasing threat of phishing attacks. The study focuses on leveraging machine learning techniques to build an AI system capable of continuously improving its knowledge base to identify and block phishing websites effectively. By incorporating self-improvement mechanisms, the AI model adapts to emerging phishing tactics and evolves its detection capabilities over time. The proposed solution holds great promise in enhancing internet security, safeguarding users from falling victim to deceptive online practices, and ultimately contributing to a safer and more trustworthy online environment.

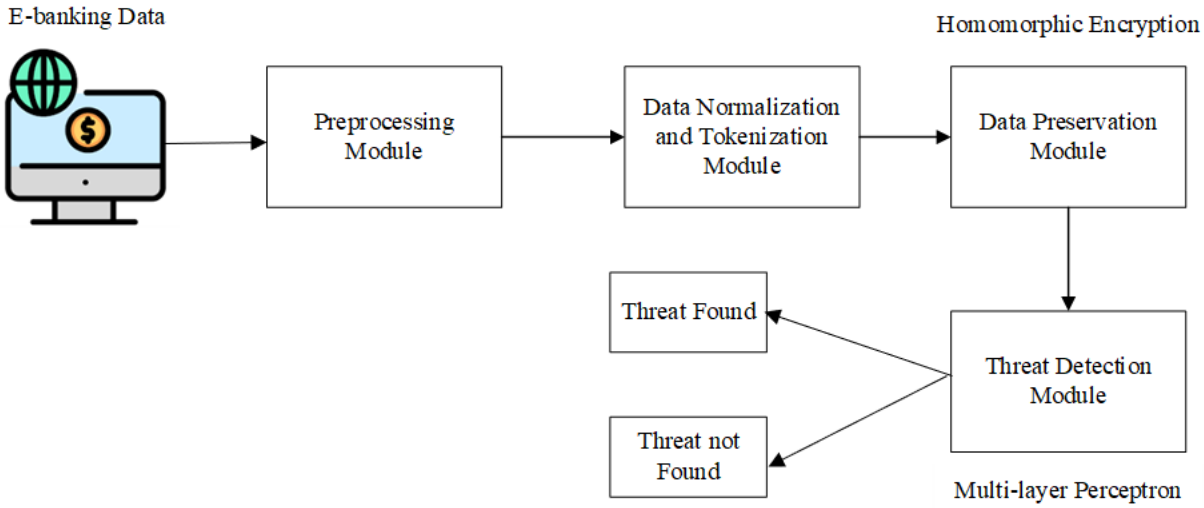
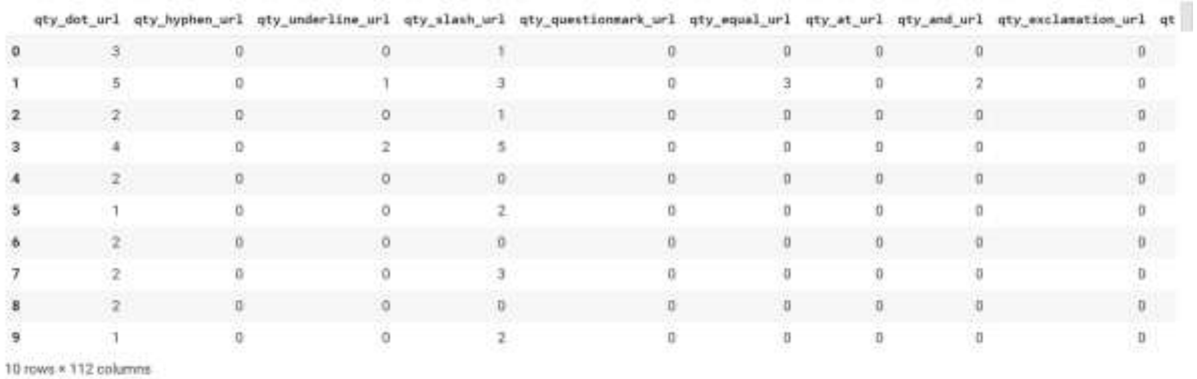


Figure 2: **Architectural Design of the Proposed System**

3.1 Dataset Description

This dataset contains 112 features extracted from 55,000 webpages, evenly split between 58,000 phishing webpages and 5,000 legitimate webpages, collected over two time periods: January to May 2015 and May to June 2022. The dataset leverages an advanced feature extraction technique using Selenium WebDriver, a browser automation tool, which ensures precise and robust data collection compared to traditional regular expression-based parsing methods. The features captured in this dataset span various categories, including URL-based features (such as URL length and the presence of special characters), domain-related features (like domain age, SSL certificate status, and DNS records), content-based features (such as suspicious keywords and embedded HTML tags), and JavaScript behavior (including the use of redirection and suspicious functions). Additionally, form-handling features (e.g., form submission methods and hidden inputs), link-based features (including internal and external links), page structure characteristics (such as the ratio of text to HTML code), visual elements (like favicon analysis and misleading logos), and user interaction behavior (including pop-up manipulation) are included. This wide range of features makes the dataset highly valuable for phishing detection, enabling researchers to analyze the effectiveness of these features, conduct rapid proof-of-concept experiments, and benchmark machine learning models for phishing classification. The dataset provides a comprehensive foundation for advancing research in phishing detection and can be used to develop, validate, and optimize models that help combat phishing attacks.



	qty_dot_url	qty_hyphen_url	qty_underline_url	qty_slash_url	qty_questionmark_url	qty_equal_url	qty_at_url	qty_and_url	qty_exclamation_url	qt
0	3	0	0	1	0	0	0	0	0	
1	5	0	1	3	0	3	0	2	0	
2	2	0	0	1	0	0	0	0	0	
3	4	0	2	5	0	0	0	0	0	
4	2	0	0	0	0	0	0	0	0	
5	1	0	0	2	0	0	0	0	0	
6	2	0	0	0	0	0	0	0	0	
7	2	0	0	3	0	0	0	0	0	
8	2	0	0	0	0	0	0	0	0	
9	1	0	0	2	0	0	0	0	0	

10 rows × 112 columns

Table 1: Dataset Sample of First Ten Rows

Data Filtering Module: The data filter module involves the following process:

1. **Remove Duplicates:** Check for and remove any duplicate entries in the dataset to avoid biasing the analysis or model training.

2. **Data Pre-processing:** a. Remove Irrelevant Information: Remove any data columns or features that do not contribute to the analysis or detection of phishing websites. For example, timestamps or user IDs might not be relevant. b. Handle Missing Values: Deal with missing data points appropriately. You can choose to impute missing values with mean, median, or mode, or simply remove rows with missing values, depending on the data.

3. **Handling Imbalanced Data:** If your dataset contains an imbalanced representation of phishing and legitimate websites, consider techniques Zsuch as oversampling, undersampling, or using synthetic data generation methods to address the imbalance.

Normalization and Tokenization Module: The data normalization and tokenization module can be seen as follows:

i. **URL Normalization:** Normalize the URLs to a standard format to remove inconsistencies and make the comparison process more accurate. For instance: a. Convert URLs to lowercase to eliminate case-sensitivity issues. b. Remove unnecessary query parameters and fragments. c. Standardize the format of encoded characters.

ii. **Feature Engineering:** Create additional features that could potentially enhance the detection of phishing websites. For example, you might extract the domain age, use of HTTPS, or the presence of suspicious keywords.

iii. **Labeling:** Ensure that each website in the dataset is properly labeled as either phishing or legitimate to enable supervised learning if applicable.

iv. **Feature Scaling**: Normalize numerical features to bring them to a common scale, such as [0, 1], or use standardization (mean = 0, standard deviation = 1). This step is crucial if you plan to use algorithms that rely on distance metrics or gradients.

Data Preservation Module: Homomorphic encryption is a cryptographic technique that allows computations to be performed on encrypted data without decrypting it first. The data preservation module utilizes homomorphic encryption to perform operations on sensitive e-banking data while keeping it encrypted, thus preserving the confidentiality and privacy of the data. This module ensures that operations like data analysis or querying can be conducted securely without exposing the raw data to unauthorized parties.

Detection Module: This module is responsible for detecting potential threats or anomalies within the e-banking data. It employs machine learning techniques, specifically a Random Forest classifier, to analyze patterns in the data and identify suspicious activities or deviations from normal behaviour. The Random Forest model is trained on historical data to recognize patterns associated with fraudulent transactions, security breaches, or other types of threats to the e-banking system. Once deployed, the threat detection module continuously monitors incoming data in real-time and triggers alerts or takes preventive actions when it detects suspicious behavior.

3.2 Component Design of the Detection Module

The component design in Figure 3 demonstrates how the system protects users from clicking on a malicious website links that looks very similar to a legitimate website. On the one hand, attackers used spelling mistakes, similar alphabetic characters, and other methods to forge the URL of the legitimate website, especially the domain name and network resource directory. For example, the link “https://aimazon.amz-z7acyuup9z0y16.xyz/v” imitates https://www.amazon. com. Although the browser on the computer can see the URL address by moving the mouse to the clickable link, it is difficult for the average user to identify these URLs with the naked eye and memory as imitating legitimate URLs. On the other hand, imitation of web content is also a key point. Typically, attackers use scripts to obtain logos, web layouts, and text from genuine web pages. Form submission pages that require user input of sensitive information are most often faked by cybercriminals, such as the login page, payment page, and find password page. With the use of the improved self-learning internet knowledge model, the system would be able to prevent the user for clicking on the malicious link, with its knowledge of phishing link.

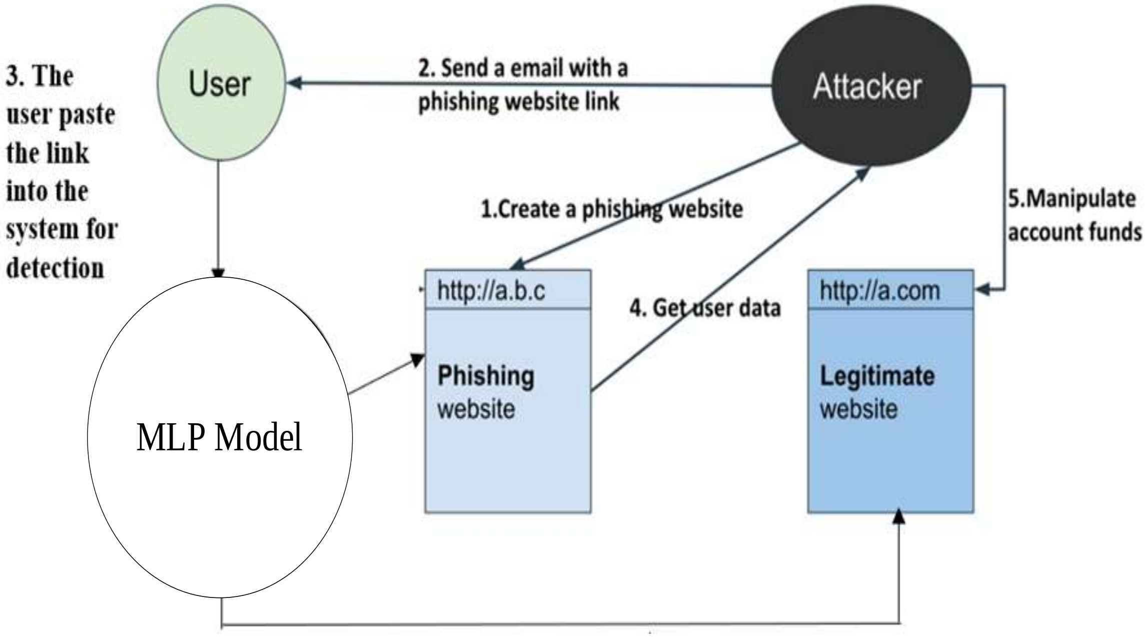

3. The user paste the link into the system for detection
User
2. Send a email with a phishing website link
Attacker
1.Create a phishing website
5.Manipulate account funds
http://a.b.c
Phishing website
4. Get user data
http://a.com
Legitimate website
MLP Model

Figure 3: Component Design of the Detection Module

4. Encryption-Based Phishing Detection in E-Banking

In the context of phishing detection in e-banking, the encryption tests simulated in the code focus on key security aspects that help distinguish between legitimate and phishing websites. The first test, **SSL/TLS Certificate Validation**, checks whether a website has a valid SSL/TLS certificate, which ensures that the communication between the user and the website is encrypted. Phishing websites often use self-signed or expired certificates, making this an important indicator of potential fraud. The second test, **HTTPS Usage**, verifies if a website is using HTTPS, which encrypts the data transmitted between the user's browser and the server. Phishing sites may try to mimic legitimate e-banking websites but fail to implement HTTPS, leaving communication unencrypted and more vulnerable to interception. The third test, **Cipher Strength**, assesses the strength of the SSL/TLS encryption ciphers used by the website. Legitimate e-banking websites use strong, up-to-date encryption ciphers to protect sensitive financial data, while phishing sites may employ weak or outdated encryption, increasing the risk of data breaches. Together, these tests offer a framework for detecting phishing sites by focusing on encryption standards, which are critical for ensuring the confidentiality, integrity, and security of online banking transactions. These encryption-related features can be valuable in machine learning models for phishing detection, helping to flag websites that fail to meet security standards.

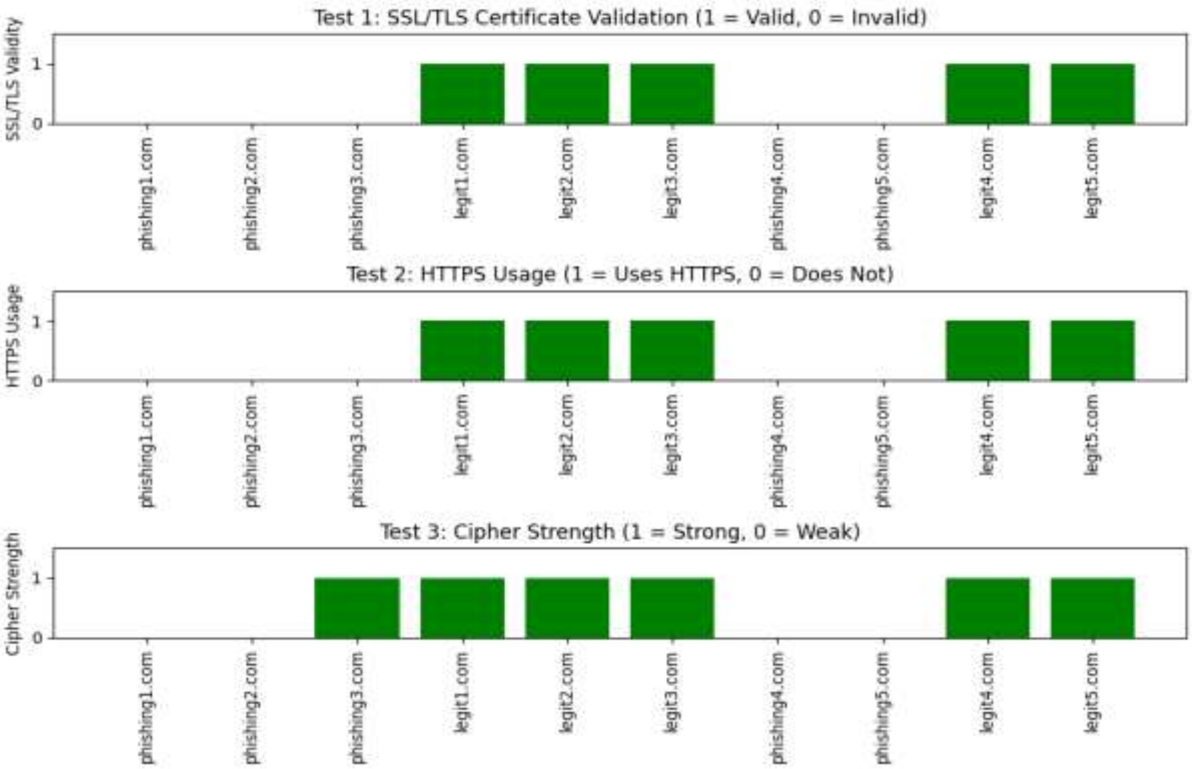
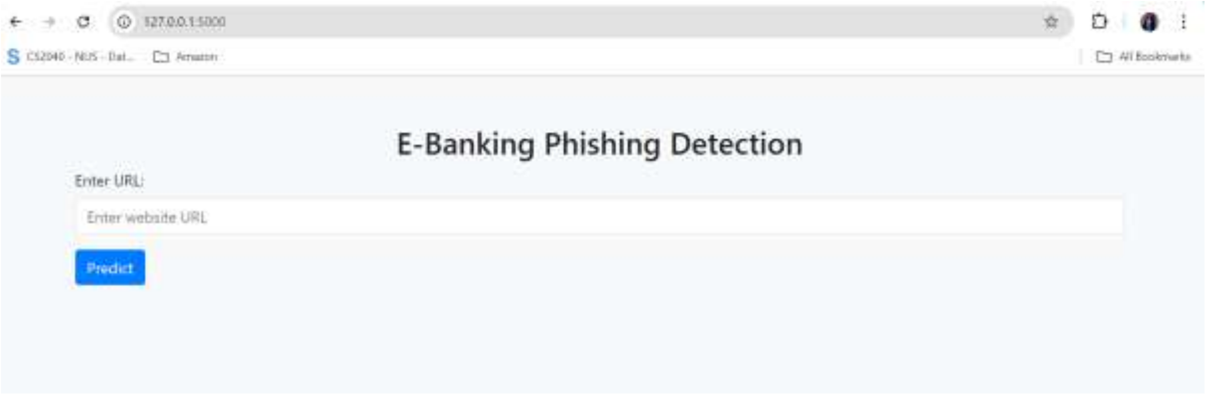


Figure 4. Encryption Test and Sensitive Information

4. Deployment to Web

The deployment of the phishing detection system involves setting up a Flask web application that provides an interactive and user-friendly interface for users to input URLs and receive predictions about their legitimacy. The system loads the pre-trained Random Forest model and TF-IDF vectorizer, both of which were saved during the training phase. When a user submits a URL, the system vectorizes it using the TF-IDF vectorizer, applies the privacy-preserving preprocessing to protect sensitive data, and then feeds it into the Random Forest model to predict whether the URL is phishing or legitimate. This can be seen in Figure 5 and Figure 6.

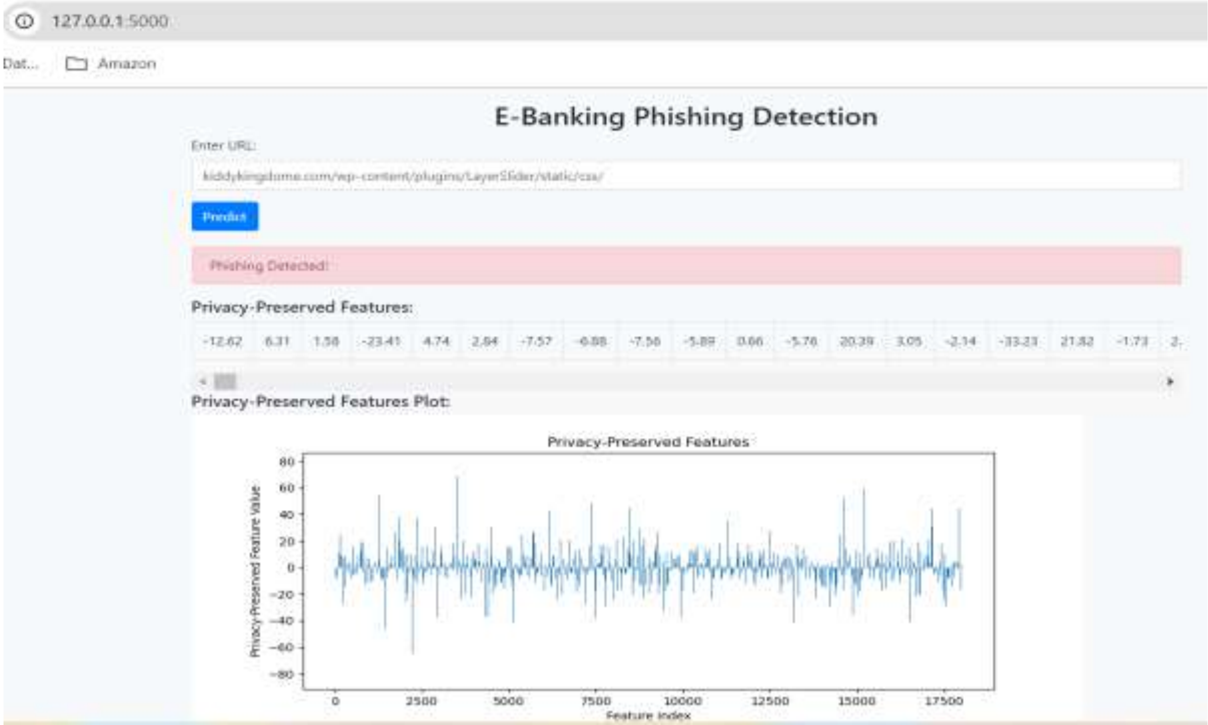


E-Banking Phishing Detection

Enter URL:

Enter website URL

Figure 5: Homepage of the System



E-Banking Phishing Detection

Enter URL:

kiddykingdome.com/wp-content/plugins/LayerSlider/static/css/

Phishing Detected!

Privacy-Preserved Features:

-12.62 6.31 1.58 -23.41 4.74 2.84 -7.57 -6.88 -7.56 -5.89 0.86 -5.76 20.39 3.05 -2.14 -33.23 21.82 -1.73 2.

Privacy-Preserved Features Plot:

Figure 6: Phishing Detected with Preserved Privacy

6. CONCLUSION

In addressing the issue of phishing attacks on e-banking platforms, the project began with a thorough analysis of vulnerabilities stemming from both user behavior and technological aspects. This included identifying how deceptive tactics exploit users' lack of awareness and the weaknesses in existing security measures, which together contribute to the high success rate of phishing attacks. By understanding these vulnerabilities, the project laid a solid foundation for developing a robust phishing detection model that can effectively mitigate such risks.

The core of the solution involved developing a machine learning model to detect phishing attacks, specifically using a Random Forest classifier. This model was trained on a dataset of URLs labeled as phishing or legitimate, leveraging features extracted through TF-IDF vectorization. The model's effectiveness in distinguishing between legitimate and phishing URLs was enhanced by incorporating a privacy-preserving technique, which involved adding Laplace noise to the feature data.

This approach ensures that sensitive customer information is protected, both during the training phase and when the model is deployed in a real-world application. The deployment was achieved through a Flask web application, which presents the model’s predictions in a user-friendly interface, providing users with clear and actionable insights while maintaining data privacy. This comprehensive approach effectively addresses phishing threats while safeguarding sensitive information, contributing to a more secure e-banking environment.

REFERENCES

AlHamad, A. Q. M. (2020). Acceptance of E-learning among university students in UAE: A practical study. *International Journal of Electrical and Computer Engineering*, *10*(4), 3660.

Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., ... & Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The lancet*, *395*(10223), 497-506.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and trends® in machine learning*, *14*(1–2), 1-210.

Karabatak, M., & Mustafa, T. (2018, March). Performance comparison of classifiers on reduced phishing website dataset. In *2018 6th international symposium on digital forensic and security (ISDFS)* (pp. 1-5). IEEE.

Mansouri, Z., Asgari, H. M., Battaleb-Looie, S., Heidari, M., & Azhdari, A. (2024). Source Identification of Urban dust Heavy metals using in situ and Satellite Data in three mega cities of Iran. *Bulletin of Environmental Contamination and Toxicology*, *113*(1), 11.

Parikh, S. B., & Atrey, P. K. (2018, April). Media-rich fake news detection: A survey. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)* (pp. 436-441). IEEE.

Peng, P., Barnes, M., Wang, C., Wang, W., Li, S., Swanson, H. L., ... & Tao, S. (2018). A meta-analysis on the relation between reading and working memory. *Psychological bulletin*, *144*(1), 48.

Sarjiyus, O., Oye, N. D., & Baha, B. Y. (2019). Improved online security framework for e-banking services in Nigeria: A real world perspective. *management*, *6*, 7.

Shima, K., Tasker, E. J., Federrath, C., & Habe, A. (2018). The effect of photoionizing feedback on star formation in isolated and colliding clouds. *Publications of the Astronomical Society of Japan*, *70*(SP2), S54.

Sonmez, A. I., Camsari, D. D., Nandakumar, A. L., Voort, J. L. V., Kung, S., Lewis, C. P., & Croarkin, P. E. (2019). Accelerated TMS for depression: a systematic review and meta-analysis. *Psychiatry research*, *273*, 770-781.

Tyagi, I., Shad, J., Sharma, S., Gaur, S., & Kaur, G. (2018, February). A novel machine learning approach to detect phishing websites. In *2018 5th International conference on signal processing and integrated networks (SPIN)* (pp. 425-430). IEEE.

Vazhayil, A., Vinayakumar, R., & Soman, K. P. (2018, July). Comparative study of the detection of malicious URLs using shallow and deep networks. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.