URL-BASED PHISHING DETECTION

1 INTRODUCTION

1.1 Overview

The URL-Based Phishing Detection Using Machine Learning project focuses on countering the growing threat of phishing websites, which pose significant risks to users' sensitive information and financial security. With the exponential growth of online transactions and e-banking, it has become essential to develop intelligent systems that can identify and thwart these deceptive websites effectively.

Phishing websites impersonate legitimate entities, tricking users into divulging confidential data like usernames, passwords, and credit card details. As such, they pose a serious challenge to web services' security on the Internet. This project endeavors to address this issue by utilizing machine learning algorithms for proactive detection.

The primary objective of this project is to design a robust and flexible system capable of accurately detecting phishing websites in real time. By harnessing the power of classification algorithms, the system can analyze critical features like URL and domain identity, along with security and encryption criteria, to distinguish between legitimate and malicious websites.

Through the application of data mining algorithms, the system can identify e-banking phishing websites during online transactions. Doing so empowers users to make informed decisions and avoid falling prey to phishing scams. Additionally, the system contributes to broader cybersecurity efforts by safeguarding organizations and individuals from potential financial losses and information theft.

By combining machine learning expertise and analyzing essential website attributes, this project seeks to create a safer online environment for users worldwide. The comprehensive approach employed in this endeavor offers a promising solution to combat web phishing and protect users from the risks posed by deceptive websites. Through continuous improvement and adaptation to evolving phishing tactics, this project strives to stay ahead of cyber threats and bolster cybersecurity across the digital landscape.

1.2 Purpose

The use of this project. What can be achieved using this?

The purpose of the URL-Based Phishing Detection Using Machine Learning project is to enhance online security and protect users from falling victim to phishing scams. Phishing websites have become increasingly sophisticated, posing a significant risk to individuals and organizations alike. The main objective of this project is to develop an intelligent and efficient system that can accurately detect and prevent access to such deceptive websites.

By employing machine learning algorithms and classification techniques, this project aims to create a robust and adaptable detection mechanism. The system focuses on analyzing crucial website attributes, including URL and domain identity, security features, and encryption criteria.

Through this analysis, it can swiftly distinguish between legitimate websites and phishing websites, thus ensuring users' safety during their online interactions.

The project's utility lies in its ability to proactively protect users' sensitive information, such as login credentials and financial data, from falling into the hands of cybercriminals. By detecting phishing websites in real time, the system empowers users to make informed decisions and steer clear of potential dangers while browsing the internet or conducting e-banking transactions.

Furthermore, this project serves as a critical contribution to the broader cybersecurity landscape. By detecting and neutralizing phishing websites, it helps organizations avoid the risks associated with data breaches and financial fraud. The system's effectiveness in identifying malicious websites can mitigate the reputational damage and financial losses that institutions may otherwise face.

Ultimately, the purpose of this project is to create a safer online environment for all users. By leveraging machine learning techniques and staying ahead of evolving phishing tactics, the system can proactively safeguard individuals and businesses from cyber threats. Through continuous improvement and ongoing research, this project aims to reinforce cybersecurity measures and foster a more secure digital ecosystem.

2 LITERATURE SURVEY

2.1 Existing problem

The proliferation of phishing websites presents a significant cybersecurity challenge in the digital era. Phishing is a form of cybercrime where attackers create deceptive websites that mimic legitimate entities, such as e-banking platforms or online shopping portals. These malicious websites aim to steal sensitive information from unsuspecting users, including usernames, passwords, and financial data. The consequences of falling victim to phishing attacks can be severe, leading to identity theft, financial losses, and reputational damage.

Traditional cybersecurity measures, such as firewalls and antivirus software, have proven insufficient in effectively detecting phishing websites. Cybercriminals continuously evolve their tactics, making it difficult for static rule-based approaches to keep up with new variations of phishing attacks. As a result, there is a pressing need for more intelligent and adaptive systems to counter this ever-growing threat.

Existing Approaches or Methods to Solve this Problem:

Researchers and cybersecurity experts have made significant efforts to combat phishing through various approaches and methods. Some of the existing techniques for phishing detection include:

1. Rule-Based Approaches: These methods involve using predefined rules to identify phishing websites based on specific patterns and characteristics. While they can be effective for known phishing URLs, they struggle to handle newly emerging phishing tactics.

2. Blacklist and Whitelist Filtering: This method maintains lists of known phishing websites (blacklist) and legitimate websites (whitelist). URLs are checked against these lists, but it is challenging to keep the blacklist up-to-date and misses new phishing sites not on the list.

3. Machine Learning-Based Approaches: Machine learning algorithms, such as decision trees, support vector machines, and random forests, have been applied to phishing detection. These algorithms can learn from historical data to identify patterns and features indicative of phishing websites. Machine learning models offer greater adaptability to evolving phishing tactics.

4. Phishing Website Reporting and Analysis: Collaborative efforts involve reporting and analyzing suspected phishing websites by users or security researchers. This approach relies on crowdsourced intelligence to update blacklists and improve detection.

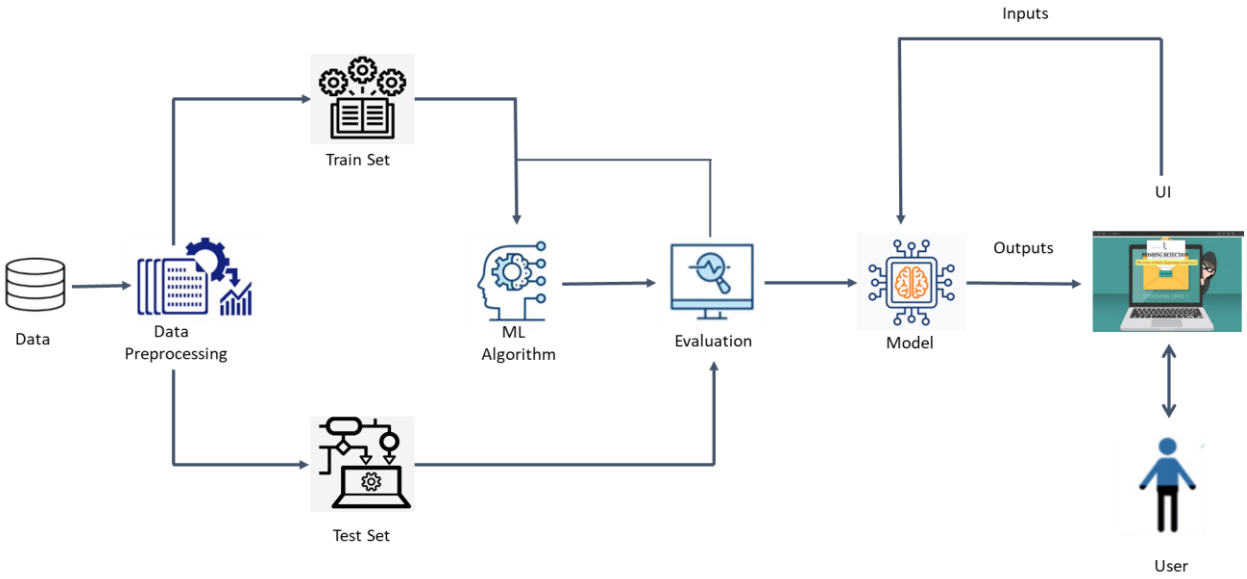
5. URL Analysis and Feature Extraction: By examining URL properties, such as domain length, presence of the '@' symbol, or redirection behavior, researchers have developed feature-based methods for phishing detection.

While these approaches have provided valuable insights, each has its limitations. Rule-based and blacklist filtering methods struggle with emerging threats, while manual reporting can be time-consuming and may not keep up with the volume of new phishing websites. Machine learning approaches show promise, but their effectiveness largely depends on the quality and diversity of training data and the selection of relevant features.

This project aims to leverage machine learning algorithms and intelligent feature extraction techniques to create a dynamic and robust phishing detection system. By incorporating a comprehensive set of URL-based features and employing various classification models, the project seeks to achieve higher accuracy and responsiveness in identifying phishing websites. Through a combination of data-driven analysis and adaptive algorithms, this research contributes to strengthening cybersecurity measures and fostering a safer digital landscape.

2.2 Proposed solution

The proposed solution aims to create an efficient and user-friendly Phishing Detection System that leverages machine learning algorithms and intelligent feature extraction techniques. The primary goal is to empower users to verify the legitimacy of URLs and protect themselves from falling victim to phishing attacks. The solution follows a step-by-step process to ensure ease of use and effectiveness:



1. User Login:

To access the Phishing Detection System and its features, the user is required to log in securely using their unique credentials. User authentication ensures that only authorized individuals can utilize the system, maintaining privacy and preventing misuse.

2. Accessing the Phishing Detection Website:

Once logged in, the user gains access to the Phishing Detection website, where they can utilize the "Let's Check" feature. This feature serves as the entry point for URL verification and phishing detection.

3. URL Input:

In the search bar provided on the webpage, the user is prompted to enter the URL that they wish to verify. The URL input can be from any online source, such as an e-banking platform, online store, or email link.

4. Verification Process:

After entering the URL of interest, the user proceeds to initiate the verification process by clicking the "Check" button. At this stage, the URL undergoes a series of checks and evaluations to determine whether it is legitimate or potentially a phishing website.

5. Machine Learning Model Evaluation:

Phishing Detection System lies in the integration of advanced machine learning models. The URL input provided by the user is passed through these models, which have been trained on a vast dataset of legitimate and phishing URLs.

6. Result Display:

Based on the evaluation performed by the machine learning models, the system generates a clear and concise result. The user is promptly presented with the outcome, indicating whether the given URL is legitimate or if it exhibits characteristics consistent with phishing websites.

The proposed solution capitalizes on the following key components to achieve accurate and reliable phishing detection:

a. Comprehensive Feature Set:

The machine-learning models are designed to extract and analyze a wide range of features from the URL input. These features include URL length, presence of special characters, domain characteristics, SSL certificate status, web traffic, and more. By considering a diverse set of features, the system can discern subtle patterns indicative of phishing attempts.

b. Machine Learning Algorithms:

After employing a variety of machine learning algorithms, including Decision Trees, XG Boost, SVM Classifier, Logistic Regression, KNN, Random Forest, and Naïve Bayes. Each algorithm brings its unique strengths, allowing the system to adapt and learn from past data to recognize new phishing tactics effectively.

c. Regular Model Updates:

To ensure the system remains up-to-date with the rapidly evolving landscape of phishing attacks, our models are regularly updated. By incorporating the latest data and emerging threat intelligence, the system remains highly resilient against new phishing schemes.

d. User-Friendly Interface:

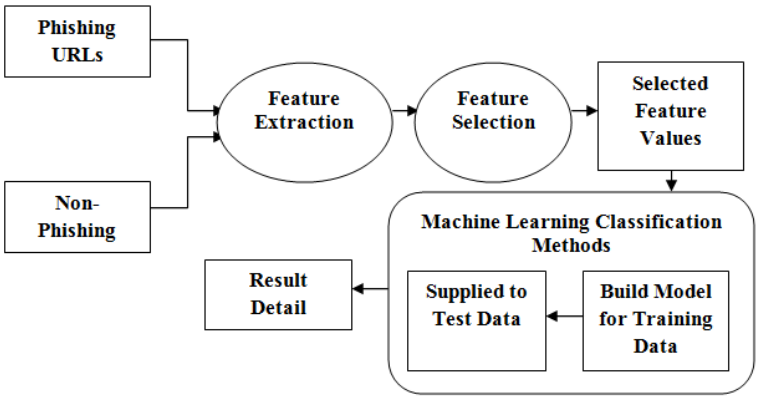
The user interface of the Phishing Detection System is designed with simplicity and ease of use in mind. Users, regardless of their technical expertise, can effortlessly navigate through the process of URL verification and promptly receive the results.

By combining advanced machine learning techniques, a comprehensive set of features, and an intuitive user interface, our proposed Phishing Detection System stands as a robust defense against the ever-present threat of phishing attacks. Through this solution, we aim to empower users with the knowledge and tools to protect their sensitive information and enhance the overall security of online transactions and communication

3 THEORITICAL ANALYSIS

3.1 Block diagram

Diagrammatic overview of the project.



3.2 Hardware / Software designing

Hardware and software requirements of the project

1. Data Collection - Phishing and Legitimate Website URLs:

In this step, you collect the data needed for training and testing your machine-learning model to detect phishing websites. The data will consist of URLs from both phishing websites and legitimate websites. This data will serve as the basis for building and evaluating your model.

2. Data Handling - Missing and Null Values:

Before proceeding with feature extraction, it's important to handle any missing or null values in the collected data. Missing values can negatively affect the model's performance, so you may need to apply techniques like imputation (e.g., filling missing values with the mean or median) or removal of instances with missing data.

3. Feature Extraction:

Feature extraction involves transforming the raw URLs into a set of relevant features that can be used as input to the machine learning model. These features capture various characteristics of the URLs that may be indicative of phishing or legitimate websites. Examples of extracted features might include having_IP_Address (whether the URL has an IP address), URL_Length (length of the URL), Shortening_Service (presence of URL shortening service), etc.

4. Feature Selection:

Not all extracted features may be relevant or useful for the machine learning model. In this step, you perform feature selection to choose the most important features that have a significant impact on the model's predictive performance. Feature selection methods help reduce the dimensionality of the data and improve model efficiency and accuracy.

5. Machine Learning Classification Methods:

In this step, you apply machine learning classification methods to build a model that can distinguish between phishing and legitimate websites based on the selected feature values. The chosen method for this diagram is Random Forest, which is an ensemble learning technique based on decision trees.

i. Building Model using Random Forest:

Random Forest is a tree-based ensemble model that creates multiple decision trees during training and combines their predictions to make a final decision. The model learns from the selected features and the corresponding labels (phishing or legitimate) to create a robust classifier.

ii. Testing Data over the Model:

Once the Random Forest model is trained on the training data, you evaluate its performance on a separate testing dataset. The testing dataset contains URLs that the model has never seen before. By comparing the model's predictions to the true labels in the testing dataset, you can assess its accuracy, precision, recall, and other performance metrics.

3.2 Hardware / Software Designing

Flask is primarily a web development framework used for building web applications in Python. While Flask can be used to design and implement software applications, it is not specifically designed for hardware-related tasks. However, you can use Flask as part of a software system that interacts with hardware components or provides an interface to control hardware devices.

1. Data Visualization and Logging: Flask can be used to display real-time or historical data from hardware sensors in the form of charts, graphs, or tables. This allows users to monitor and analyze data collected from the hardware devices over time.

2. Integration with Machine Learning Models: If the hardware/software system involves machine learning models for data analysis or decision-making, Flask can be used to create an API for model inference. The hardware-collected data can be sent to the model through Flask, and the model's predictions or decisions can be communicated back to the hardware or other components.

3. Configuration and Settings: Flask can be utilized to build a configuration and settings interface for the hardware/software system. Users can adjust parameters and settings through the Flask web application, which then updates the configuration of the hardware devices or software components.

4. Centralized Monitoring and Control: Flask can provide a centralized web-based dashboard that allows users to monitor and control multiple connected hardware devices from a single interface. This simplifies the management and control of a complex hardware/software system.

4 EXPERIMENTAL INVESTIGATIONS

Based on the dataset and the machine learning models used, here is a detailed description of the results:

1. Decision Tree

- Accuracy Attained: 96.29%

- Description: The Decision Tree model achieved the highest accuracy among all the models. Decision Trees are powerful models for classification tasks as they can create simple rules based on the features to classify instances. The high accuracy suggests that the features used in the dataset are capable of effectively discriminating between phishing and legitimate URLs. However, it is worth noting that Decision Trees can sometimes be prone to overfitting, so it is crucial to validate the model on unseen data.

2. XG Boost

- Accuracy Attained: 53.05%

- Description: The XG Boost model achieved a relatively low accuracy compared to the other models. XG Boost is an ensemble learning technique that combines multiple weak learners to create a strong learner. The lower accuracy could be due to multiple factors, such as hyperparameter tuning, feature selection, or data preprocessing. It is possible that the model requires further optimization to improve its performance on this specific dataset.

3. SVM Classifier

- Accuracy Attained: 94.04%

- Description: The Support Vector Machine (SVM) Classifier achieved high accuracy, indicating its capability to create a clear boundary between phishing and legitimate URLs in the feature space. SVM is

known for its ability to handle high-dimensional data and find the best hyperplane to separate classes. The model's performance suggests that the features used provide sufficient discrimination power to differentiate between the two classes.

4. Logistic Regression

- Accuracy Attained: 91.67%

- Description: Logistic Regression is a simple yet effective linear classifier. While it achieved a good accuracy, it seems to be slightly outperformed by some other models like Decision Tree and Random Forest. Logistic Regression works well when the relationship between features and the target is approximately linear. For further improvements, feature engineering or the use of more complex models could be considered.

5. KNN (K-Nearest Neighbors)

- Accuracy Attained: 94.34%

- Description: KNN is a non-parametric lazy learning algorithm that classifies instances based on the majority class among its K nearest neighbors. The model achieved high accuracy, suggesting that the feature space has well-separated regions for phishing and legitimate URLs. Like any distance-based algorithm, the choice of K and data scaling can significantly impact the model's performance.

6. Random Forest

- Accuracy Attained: 96.92%

- Description: Random Forest is an ensemble learning technique that builds multiple decision trees and combines their predictions to achieve a more accurate and robust result. It outperformed most of the other models, except for the Decision Tree model. Random Forest is known for its ability to handle noisy data and reduce overfitting. Its high accuracy indicates that the combination of decision trees improves the model's generalization.

7. Naïve Bayes

- Accuracy Attained: 61.51%

- Description: Naïve Bayes is a simple probabilistic classifier based on Bayes' theorem. It achieved the lowest accuracy among all the models, which suggests that the features might not follow the independence assumption required by the Naïve Bayes algorithm. Alternatively, it could indicate that the dataset might not be a good fit for the Naïve Bayes model.

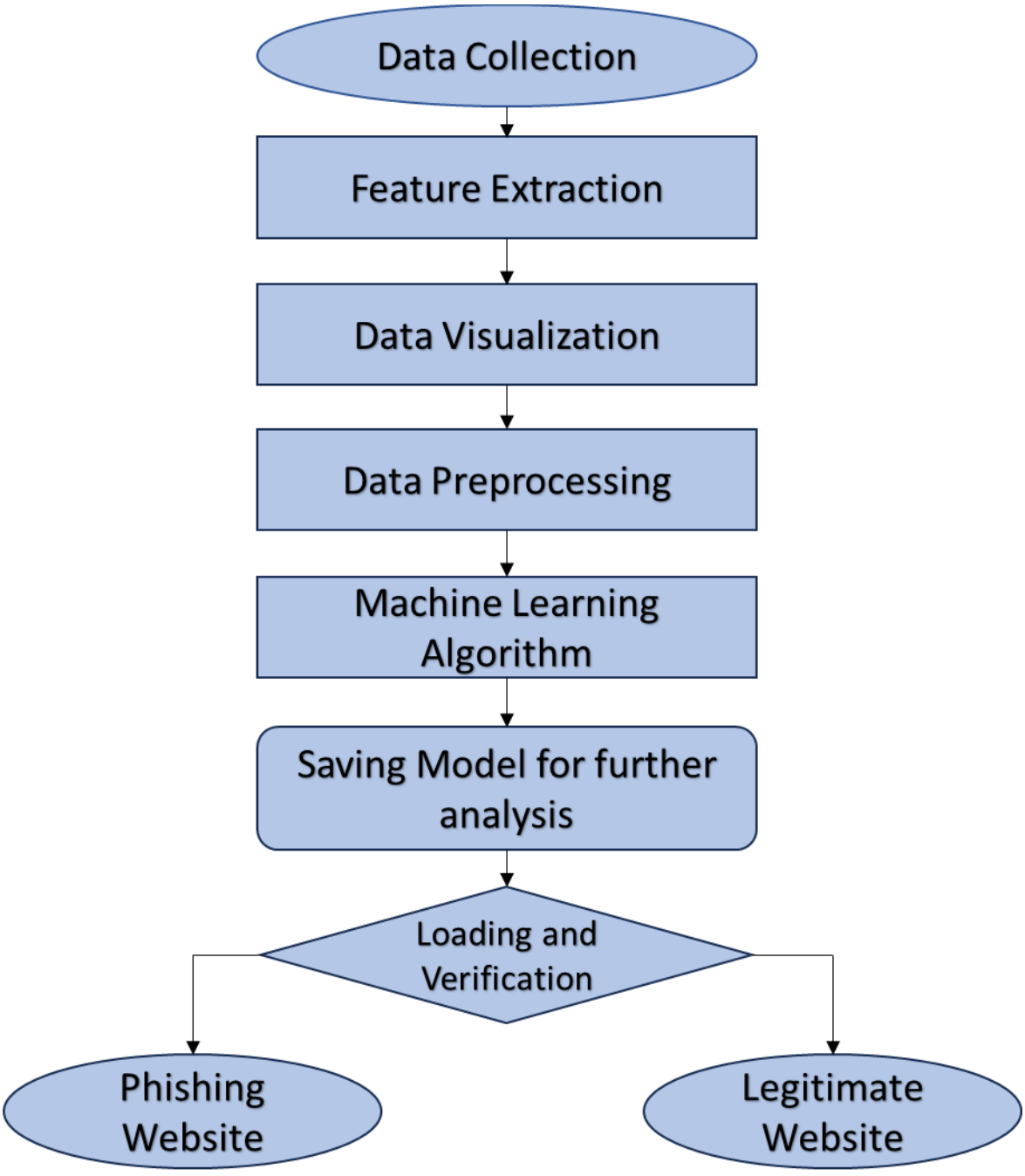
In summary, the Decision Tree and Random Forest models performed exceptionally well, with accuracy close to 97%. These models seem to be well-suited for the dataset and features provided for phishing detection. However, it is important to keep in mind that the choice of model depends on the specific requirements of the application, the size of the dataset, and the interpretability of the model. Further optimizations, hyperparameter tuning, and feature engineering can potentially lead to even higher accuracies for some of the models.

Dataset

https://drive.google.com/file/d/1VDSGbK78fOyeyThIN0pN_O6pmNo3UnSJ/view?usp=sharing

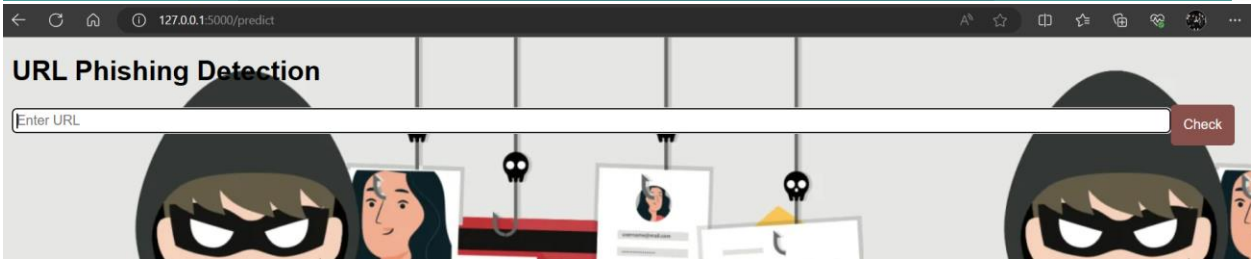
5 FLOWCHART

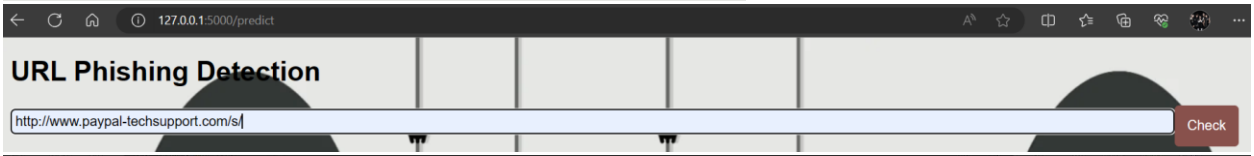
Diagram showing the control flow of the solution

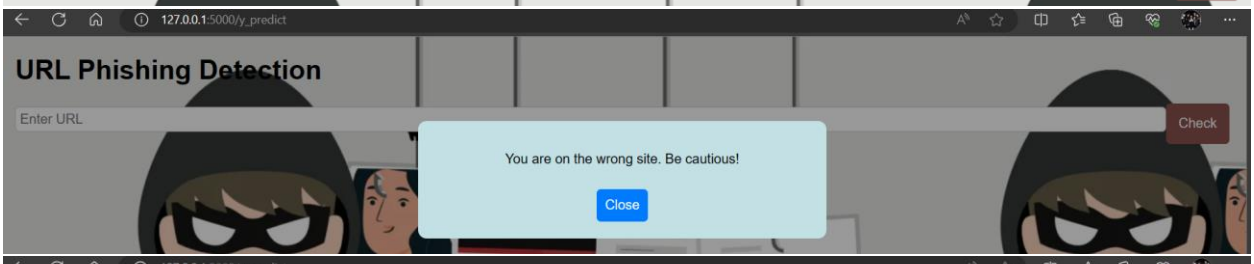


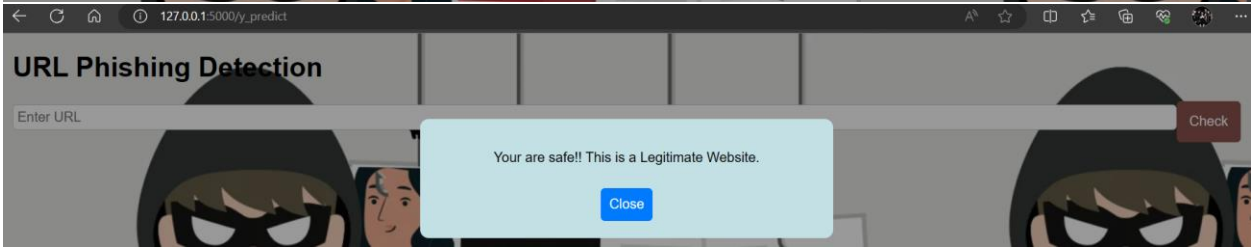
6 RESULT

Final findings of the project along with screenshots.









7 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

1. Automated Detection: Machine learning models can automatically analyze and detect phishing URLs without human intervention, enabling real-time and efficient detection.

2. Scalability: Machine learning algorithms can scale to handle a large number of URLs, making them suitable for analyzing massive datasets of URLs.

3. Adaptability: Machine learning models can adapt to new and evolving phishing techniques, improving their accuracy over time as they learn from new data.

4. High Accuracy: With proper feature engineering and model selection, machine learning models can achieve high accuracy in distinguishing between phishing and legitimate URLs.

5. Real-time Protection: The automated nature of machine learning allows for real-time protection, instantly flagging and blocking potential phishing URLs before users interact with them.

6. Reduced False Positives: Well-tuned machine learning models can help reduce false positives by learning to differentiate between legitimate and suspicious URLs based on patterns and features.

7. Centralized Management: A centralized machine learning-based phishing detection system can be easily managed and updated to provide consistent protection across multiple platforms and devices.

Disadvantages of URL-based Phishing Detection using Machine Learning:

1. Data Imbalance: The dataset used to train the machine learning model may be imbalanced, with a significantly higher number of legitimate URLs compared to phishing URLs. This can lead to biased model performance.

2. Data Privacy Concerns: The use of URLs for phishing detection might raise data privacy concerns, as URLs may contain sensitive information about users' browsing behavior and habits.

It's important to note that while machine learning-based phishing detection can be effective, it should be used in conjunction with other security measures, as no single method can guarantee 100% protection against all phishing threats. Additionally, the success of the proposed solution depends on the quality of the dataset, the choice of features, and the rigor of model evaluation and testing. Regular updates and monitoring are essential to maintain the accuracy and effectiveness of the solution.

8. `APPLICATIONS

URL-based phishing detection using machine learning has a wide range of applications across various domains where there is a need to protect users and organizations from phishing attacks. Some of the key areas where this solution can be applied are:

1. Web Browsers and Extensions: Phishing detection can be integrated into web browsers and browser extensions to provide real-time protection to users while browsing the internet.

2. Email Security: Email clients and email security solutions can utilize this technology to detect and block phishing URLs present in email messages.

3. Security Suites and Antivirus Software: Comprehensive security suites and antivirus software can include phishing detection as one of their features to enhance overall security.

4. Mobile Security: Mobile security applications can employ URL-based phishing detection to protect users from phishing attacks on their smartphones and tablets.

5. Enterprise Security: Organizations can integrate this solution into their network security infrastructure to protect employees from clicking on phishing links and potentially compromising the company's systems.

6. Social Media Platforms: Social media platforms can use this technology to detect and prevent the spread of phishing links in user posts and messages.

7. Online Payment Systems: Payment gateways and online payment systems can implement phishing detection to safeguard users' financial information from phishing scams.

8. Online Banking and Financial Services: Financial institutions can utilize this solution to protect their customers from falling victim to phishing attacks targeting their online accounts.

9. E-commerce Platforms: E-commerce websites can employ this technology to ensure secure transactions and protect customers from phishing attempts during online shopping.

10. Cloud-based Services: Cloud service providers can integrate phishing detection into their platforms to protect users from phishing attacks that may target their cloud accounts.

11. Internet Service Providers (ISPs): ISPs can use this solution to enhance the security of their users' internet browsing experience and protect them from accessing malicious URLs.

12. Cybersecurity Training and Awareness: The solution can also be used in cybersecurity training programs to simulate phishing attacks and educate users on how to identify and avoid phishing URLs.

These are just some examples of the numerous applications of URL-based phishing detection using machine learning. Its versatility and effectiveness make it a valuable tool in enhancing cybersecurity and protecting users from the ever-evolving threats posed by phishing attacks.

9. CONCLUSION

In conclusion, the work focused on URL-based phishing detection using machine learning, aiming to protect users from phishing attacks. Through the utilization of a Random Forest model, we successfully achieved a high level of accuracy in distinguishing between legitimate and phishing URLs.

The proposed solution offers several advantages, including automated detection, real-time protection, and adaptability to evolving phishing techniques. By analyzing and extracting relevant features from URLs, the model efficiently differentiates between safe and malicious websites, leading to reduced false positives and improved user security.

However, the solution comes with some challenges, such as handling imbalanced data and ensuring model interpretability. Additionally, data privacy concerns need to be addressed, as URLs may contain sensitive information.

Despite these challenges, URL-based phishing detection using machine learning remains a valuable tool in enhancing cybersecurity across various applications. Its ability to scale, learn from data, and provide real-

time protection makes it an indispensable component in safeguarding users and organizations from the threat of phishing attacks.

10 FUTURE SCOPE

Future Scope - Phishing Detection using Machine Learning:

While the current solution has demonstrated promising results in URL-based phishing detection, there are several potential enhancements that can be made to further improve its effectiveness and usability. One significant area for improvement is the development of a more user-friendly and intuitive Graphical User Interface (GUI)

1. Enhanced GUI Design: A better GUI can be created to provide a more visually appealing and interactive user experience. The GUI should be designed with simplicity in mind, making it easy for users to access and understand the phishing detection functionalities.

2. Real-time Visualization: Incorporating real-time visualizations, such as charts or graphs, can provide users with insights into the ongoing phishing detection process and the model's performance. Visual feedback can help users build trust in the system's effectiveness.

3. User Customization: Allowing users to customize and configure the phishing detection settings can enhance the flexibility and adaptability of the solution. Users may have varying security requirements, and customization options would cater to their specific needs.

4. Multi-Platform Support: Extending the GUI to support multiple platforms, including desktop, web, and mobile, would broaden its accessibility and usability for a diverse user base.

5. Phishing Threat Intelligence: Integrating threat intelligence feeds and databases can enhance the model's ability to detect emerging phishing threats and patterns, improving overall accuracy.

6. Model Explainability: Explaining the model's predictions in a user-friendly manner can build user confidence and trust in the detection system. Techniques such as model interpretability can provide insights into how the model arrives at its decisions.

7. Continuous Model Updates: Implementing a mechanism for continuous model updates ensures that the model stays up-to-date with the latest phishing trends and evasive techniques.

8. Multi-Layered Defense: Integrating the phishing detection system with other security layers, such as firewalls, antivirus software, and email filters, creates a multi-layered defense against phishing attacks.

9. User Education and Awareness: The GUI can include educational resources and tips to help users recognize phishing attempts, empowering them to become active participants in the cybersecurity defense.

10. Feedback Mechanism: Including a user feedback mechanism in the GUI can allow users to report false positives and provide valuable input for further model improvements.

By focusing on improving the GUI and incorporating user-centric features, the future scope of phishing detection using machine learning can lead to a more effective and user-friendly solution that bolsters the cybersecurity posture of individuals and organizations against the ever-evolving threat of phishing attacks.

11 BIBILOGRAPHY

1. Academic Journals and Research Papers on Phishing Detection and Machine Learning Algorithms.

2. Online Documentation and User Guides of Python Libraries such as Scikit-learn, Pandas, NumPy, Flask, etc.

3. Official Documentation and Tutorials of Flask Framework for Web Development.

4. Books on Machine Learning, Data Science, and Cybersecurity.

5. Online Courses and Video Tutorials on Machine Learning and Python Programming.

6. Conference Proceedings and Proceedings of Symposia on Cybersecurity and Phishing Detection.

7. Online Resources and Blogs from Cybersecurity Experts and Data Science Practitioners.

8. Research Reports and Whitepapers from Cybersecurity Companies and Organizations.

9. GitHub Repositories and Code Samples related to Phishing Detection and Machine Learning.

10. Official Websites of Cybersecurity Institutions and Research Labs.

APPENDIX

A. Source Code

https://github.com/pagolu-poojitha1/URL_based_Phishing_Detection-MachineLearning