

Phishing Website Detection Using GUI Implementation

Namavarapu Roja¹, N Naveen Kumar²

¹Post-Graduate Student, Department of Information Technology, Software Engineering, Jawaharlal Nehru Technological University, Hyderabad, India.

²Professor, Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad, India

ABSTRACT:

This work proposes a system that combines machine learning for phishing website detection with a user-friendly graphical user interface (GUI). The machine learning model is trained on a comprehensive dataset, analyzing URL characteristics, website content analysis, and security certificates. The system's user-friendliness is achieved through a well-designed GUI, which allows users to enter a website URL and the model analyzes the data to classify the website as legitimate or a potential phishing attempt. This system offers a robust, automated method for detecting phishing websites and eliminates the need for technical expertise, empowering users to make informed decisions about interacting with the website.

Keywords—phishing website detection, Legitimate websites, Digital landscape, Machine learning, GUI, URL, Model, Features.

1. INTRODUCTION:

The internet has revolutionized communication, commerce, and information access, but it has also opened doors for malicious actors. Phishing attacks, a prominent form of cybercrime, utilize social engineering techniques to trick consumers and steal private data. Deceptive phishing websites masquerade as legitimate ones, tricking unsuspecting users into divulging credentials like usernames, passwords, or credit card details. This project focuses on using machine learning to detect phishing websites through a curated dataset.

Several machine learning algorithms were explored for their suitability in phishing website detection, including Decision Trees, Random Forest, Multilayer Perceptrons (MLPs), XGBoost, Autoencoder Neural Networks, and Support Vector Machines (SVMs). Phase 1 successfully trained a chosen model on the prepared dataset, enabling it to effectively classify a website as legitimate or potentially phish.

Phase 2 focuses on bridging the gap between the trained model in phase 1 and the user by implementing a user-friendly Graphical User Interface (GUI). A well-designed GUI acts as a crucial intermediary, empowering users to leverage the model's capabilities without requiring in-depth technical knowledge. The primary goal of Phase 2 is to bridge the gap between the model and its users by implementing a GUI that fosters user control and security.

The objectives of the GUI include ease of use, increased security awareness, proactive website evaluation, real-time detection, widespread adoption, and scalability for future development. The GUI should create a smooth workflow where users enter a URL, present clear results, enable real-time detection, encourage widespread adoption, and allow for potential integration with web browsers or other applications. By addressing these challenges, the project aims to protect against phishing attacks and improve overall cybersecurity.

2. TECHNOLOGIES:

Python is the backbone of our project, providing versatility and libraries for essential tasks. It allows seamless integration of various functionalities, enabling data analysis using libraries like NumPy and Pandas. Flask Framework is used for frontend technologies, including HTML, CSS, and JS. BeautifulSoup, a Python package, simplifies website navigation and extracts text content, links, and meta tags for analysis. CheckPageRank APIs provide a metric for website importance, but are unreliable for identifying phishing attempts. Overall, Python's readable syntax promotes clear and maintainable code.

3.METHODOLOGY:

The project developed a machine learning model for detecting phishing websites using a structured methodology. The dataset consisted of legitimate and phishing URLs, preprocessed for consistency. Relevant features were extracted, and a gradient boosting algorithm was used to train the model. The model learned patterns and relationships, and its performance was evaluated using unseen data, highlighting its advantages and disadvantages.

3.1 SYSTEM ARCHITECTURE:

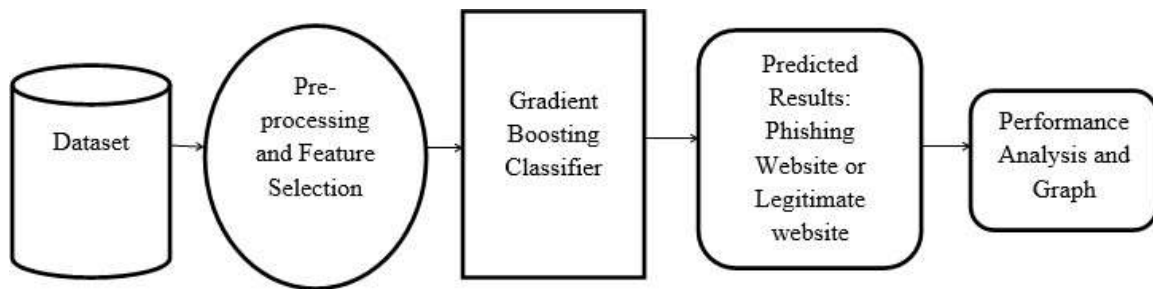


Fig 3.1: Project Diagram

3.2 USE CASE DIAGRAM:

In the Unified Modeling Language (UML), a use case diagram is a kind of behavioral

diagram that is produced from and defined by a use-case study. Presenting a graphical summary of a system's functionality in terms of actors, their objectives (shown as use cases), and any dependencies among those use cases is its main objective. A use case diagram's primary objective

is to illustrate which actors use the system and what functions are carried out. It is possible to illustrate the roles of the system's actors.

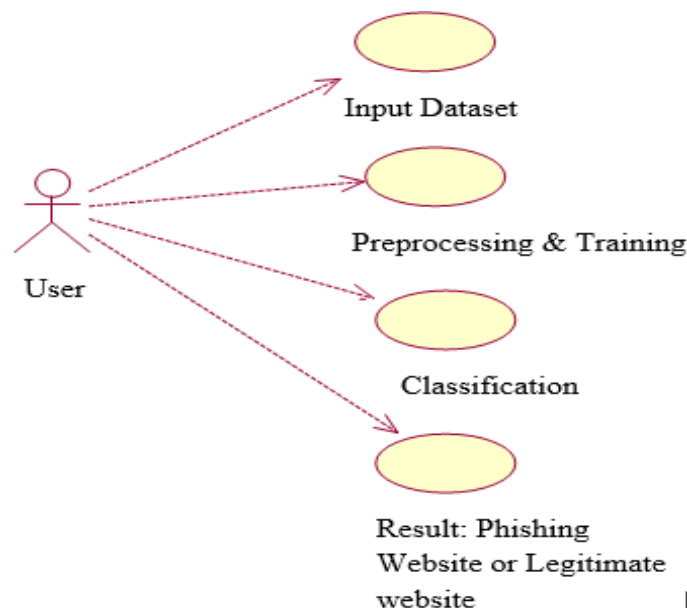


Fig 3.2 Use Case Diagram

3.3 FEATURE EXTRACTION:

The Python class 'FeatureExtraction' is used to extract features from URLs, using techniques such as regular expressions, string manipulation, and external libraries. The extracted features are then grouped into a list within the 'FeatureExtraction' class instance. Error handling mechanisms are implemented throughout the code to ensure resilience against potential issues. Exception handling is employed to manage scenarios like invalid URLs, failed HTTP requests, or parsing errors encountered while processing HTML content.

The Gradient Boosting classification Algorithm is used to predict links from the extracted features. XGBoost is an effective and scalable machine learning technique for supervised learning applications, known for its exceptional performance, speed, and versatility. It sequentially constructs an ensemble of weak learners, usually decision trees, with each tree fixing the mistakes of the previous one. XGBoost also has a unique mechanism for handling missing data, allowing it to automatically learn the best imputation strategy during the training process.

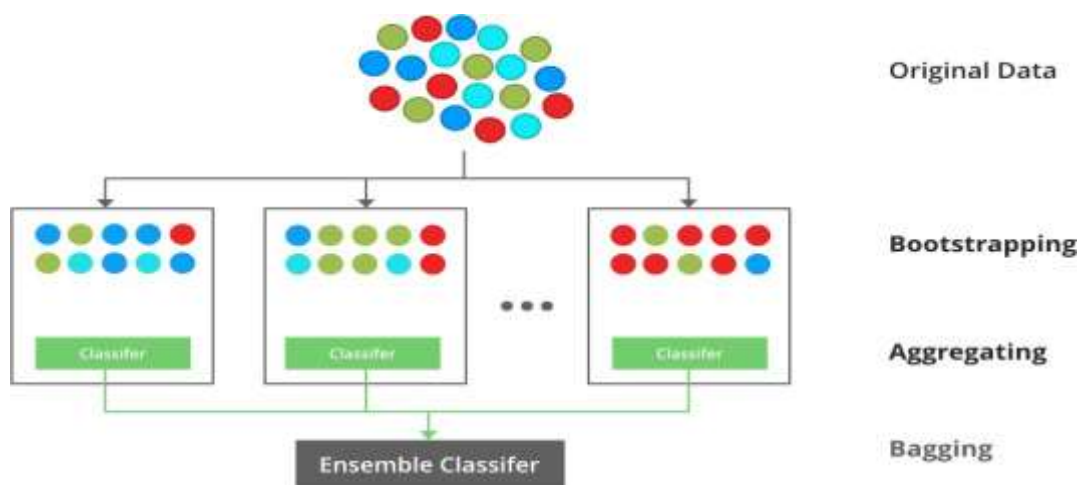


Fig 3.3: XGBoost algorithm

4. IMPLEMENTATION:

The phishing website detection system's GUI aims to bridge the gap between trained machine learning models and everyday users, empowering them to proactively assess website legitimacy. Key features include a home page, login page, dataset upload page, prediction page, performance analysis page, and confusion matrix. The home page provides a brief overview of the system's purpose and functionalities, while the login page allows authorized users to log in with appropriate credentials. The prediction page uses the trained model to classify websites as legitimate or phishing attempts, with results displayed on the prediction page. The performance analysis page displays metrics like accuracy, precision, and recall.



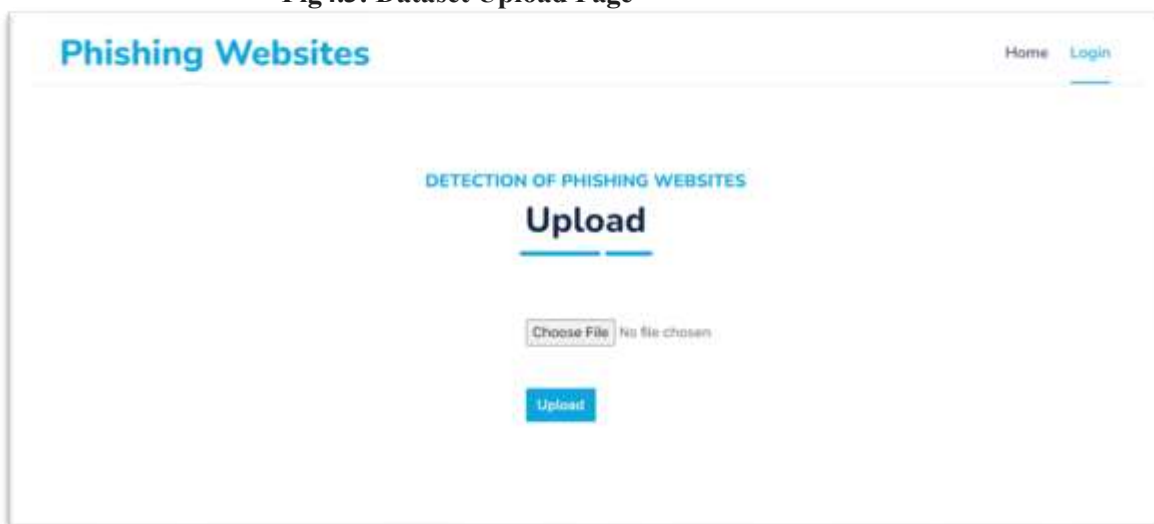
Fig4.1:Home Page



The login page features a header with the title "Phishing Websites" and navigation links for "Home" and "Login". The main content area is titled "DETECTION OF PHISHING WEBSITES" and contains a "Login" section. This section includes input fields for "Username" and "Password", followed by a blue "Login" button.

Fig 4.2: Login page

Fig4.3: Dataset Upload Page



The dataset upload page has a header with "Phishing Websites" and "Home" and "Login" links. The main section is titled "DETECTION OF PHISHING WEBSITES" and features an "Upload" section. It includes a "Choose File" button, a text indicator "No file chosen", and a blue "Upload" button.



Fig 4.4: Confusion Matrix

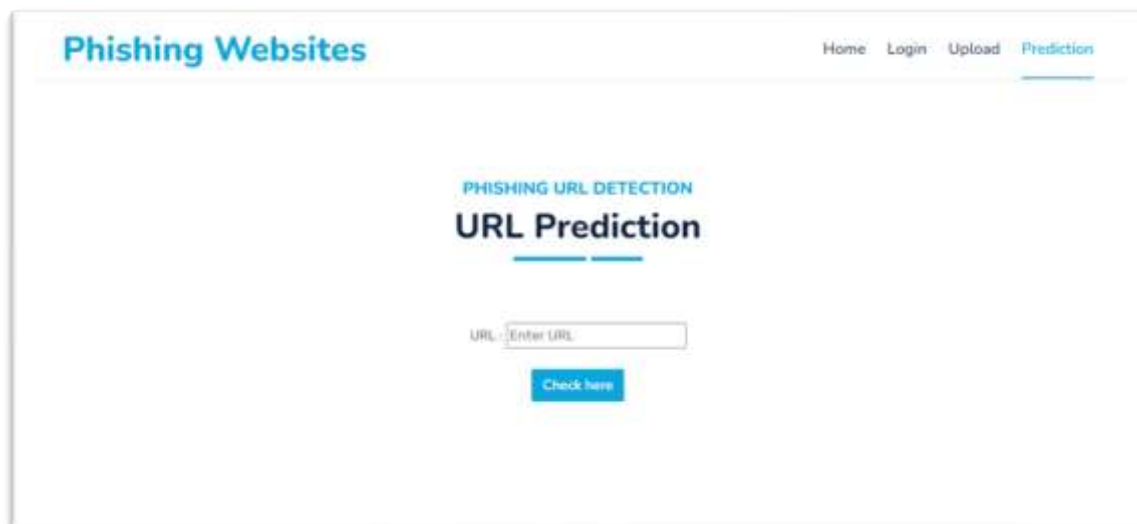


Fig4.5: Prediction Page

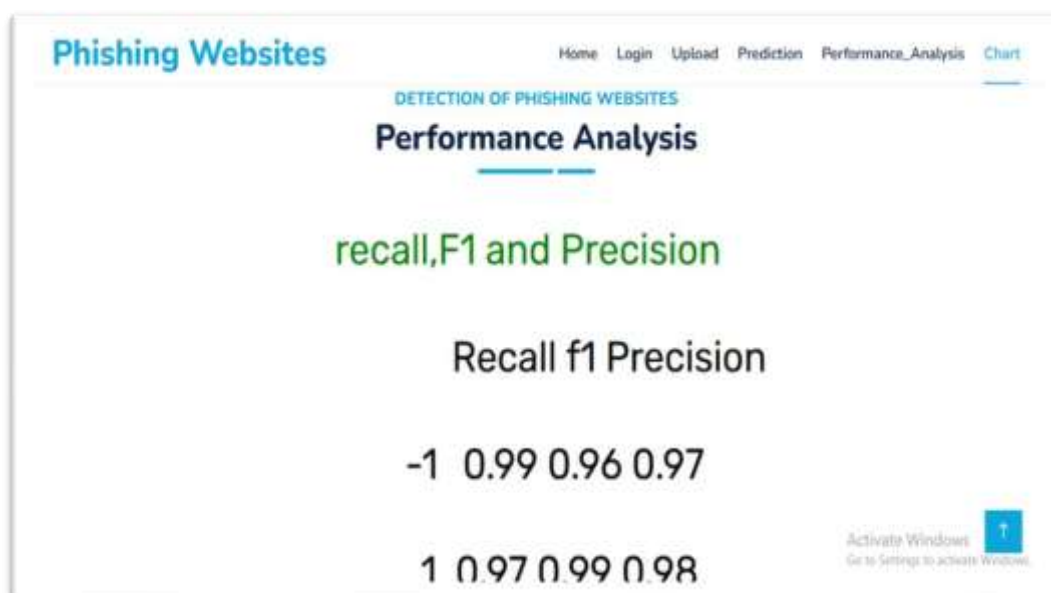


Fig4.6: Recall, F1 and Precision



Fig 4.7: Pie Chart showing the distribution of Legitimate and Phishing Links in the dataset.

5.RESULT:

This study focuses on the development and evaluation of a phishing website detection system using a Graphical User Interface (GUI) and machine learning. The project involved selecting and training six algorithms, including Random Forest, XGBoost, Support Vector Machine (SVM), Decision Tree, Autoencoder Neural Network, and Multilayer Perceptron. The results showed that XGBoost, a gradient boosting classification algorithm, achieved the best accuracy in differentiating between reputable and fraudulent websites. A user-friendly GUI was implemented to bridge the gap between the trained model and its practical application. The system allows users to enter a URL for legitimacy assessment, initiate the prediction process, and view the model's predictions. Future work could include performance analysis metrics, real-world testing, and false positive/negative analysis. Overall, this project demonstrates the successful development of a phishing website detection system.



Fig 5.1:Phishing URL Detection



Fig5.2: Phishing URL Detection

6. CONCLUTION:

To sum up, the project to detect phishing websites has provided valuable insights into the effectiveness of various machine learning models for safeguarding users against malicious online activities. Through a comprehensive evaluation of Decision Trees, Support Vector Machines (SVM), Random Forest, XGBoost, Multi-Layer Perceptrons (MLP), and an Autoencoder Neural Network, we have identified nuanced strengths and trade-offs within each approach. Notably, XGBoost emerged as a top-performing model, demonstrating a high accuracy of 94%, showcasing its robustness in capturing complex patterns indicative of phishing activities. Ensemble methods, such as Random Forest, and the

integration of feature learning through the Autoencoder Neural Network with SVM, also exhibited strong performance, emphasizing the significance of diverse model architectures in phishing detection.

This project not only advances our understanding of the capabilities of machine learning in cybersecurity but also underscores the importance of a tailored approach to model selection based on specific project requirements. As the threat landscape continues to evolve, the findings from this study will inform ongoing efforts to refine and enhance phishing detection systems, emphasizing the need for adaptability, continuous improvement, and a holistic understanding of the advantages and disadvantages of various machine learning methodologies in cybersecurity applications.

REFERENCES:

1. S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "Detection of phishing emails using data mining algorithms, " in 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2015: IEEE, pp. 1-8.
2. M. Aydin and N. Baykal, "Feature extraction and classification phishing websites based on URL, " in 2015 IEEE Conference on Communications and Network Security (CNS), 2015: IEEE, pp. 769-770. paragraph text.
3. W. Chu, B. B. Zhu, F. Xue, X. Guan, and Z. Cai, "Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs, " in 2013 IEEE international conference on communications (ICC), 2013: IEEE, pp. 1990-1994.
4. L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, "A novel approach for phishing detection using URL-based heuristic, " in 2014 International conference on Computing, management, and telecommunications (ComManTel), 2014: IEEE, pp. 298-303.
5. S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, Dec. 2014. doi: 10.1109/TNSM.2014.2369055
6. N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based on hybrid feature selection and random forest algorithm," *2014 International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia, 2014, pp. 1–6. doi: 10.1109/ICCOINS.2014.6868463
7. Y. Zhang, J. Hong, and L. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites," *Proceedings of the 16th International Conference on World Wide Web*, Banff, Canada, 2007, pp. 639–648. doi: 10.1145/1242572.1242659
8. M. Aburrous, M. A. Hossain, F. K. Almazroi, and K. S. Almuhtadi, "Intelligent Phishing Detection System for E-banking Using Fuzzy Data Mining," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7913–7921, Dec. 2010. doi: 10.1016/j.eswa.2010.04.033
9. □ D. H. K. Tsai, Y. T. Li, and K. C. Wu, "Detecting Phishing Websites Based on URL Features Using Deep Neural Network," *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Taoyuan, Taiwan, 2020, pp. 1–2. doi: 10.1109/ICCE-TW49152.2020.9226003
10. □ S. M. Aljawarneh, M. Aldwairi, and M. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, Jan. 2018. doi: 10.1016/j.jocs.2017.03.006
11. □ A. Verma and K. K. Bhatia, "A framework for phishing detection and classification using machine learning algorithms," *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Noida, India, 2016, pp. 1–6. doi: 10.1109/CCAA.2016.7813772