Combating Cyber Threats: Machine Learning for Phishing Website Detection

Master of Applied Computing

University of Windsor

Statistical Learning

Authors	ID
Deon Victor Lobo	110127749
Gagandeep Singh	110123330
Ankur Mangroliya	110127190

1. ABSTRACT

Phishing websites pose a significant threat to online users by impersonating legitimate URLs, leading to fraudulent activities. Detecting these deceptive websites is crucial for safeguarding users' sensitive information and preventing cyberattacks. Our project employs machine learning techniques to train models on datasets containing both phishing and legitimate URLs, achieving a high level of accuracy in predicting the nature of URLs. By successfully training models to distinguish between phishing and legitimate URLs, our solution provides a proactive defense mechanism against online threats.

2.INTRODUCTION

Phishing, a prevalent form of social engineering and cyber attack, continues to pose significant threats to online users. Through such attacks, perpetrators target unsuspecting individuals, often tricking them into revealing confidential information under false pretenses. This information is then exploited for fraudulent purposes, highlighting the need for robust mechanisms to combat phishing attempts. Despite efforts to raise awareness and maintain blacklists of known phishing websites, users remain susceptible to these deceptive tactics. Thus, there is a pressing need for effective detection methods that can identify phishing websites in their early stages, thereby mitigating the risks posed by such malicious activities. Machine learning and deep neural network algorithms offer promising avenues for achieving this goal, as they can analyze vast amounts of data to discern patterns indicative of phishing behavior.

The objective of this project is to leverage machine learning and deep neural network techniques to develop models capable of accurately predicting phishing websites. To achieve this goal, we adopt a systematic approach encompassing data collection, feature extraction, model training, and evaluation. Initially, we collect datasets containing both phishing and legitimate URLs from reputable open-source platforms. These datasets serve as the foundation for training and evaluating our models. Next, we extract relevant features from the URL data, including address bar-based, domain-based, HTML, and JavaScript attributes. These features provide valuable insights into the characteristics of phishing websites, enabling our models to make informed predictions.

Subsequently, we preprocess the dataset using exploratory data analysis (EDA) techniques to identify and handle any inconsistencies or anomalies. Following data preprocessing, we partition the dataset into training and testing sets to facilitate model training and evaluation. We then employ a selection of machine learning and deep neural network algorithms, including Support Vector Machines and Random Forest, to train our models on the training dataset. Finally, we evaluate the performance of each model using appropriate accuracy metrics and compare their results to determine the most effective approach for phishing detection. By systematically addressing each step in the project workflow, we aim to develop robust models capable of accurately identifying phishing websites, thereby enhancing online security for users worldwide.

Phishing websites masquerade as legitimate URLs, often employing convincing designs and URLs to deceive unsuspecting users into divulging confidential information. Traditional methods of detection rely on manual inspection or predefined blacklists, which are susceptible to evasion tactics employed by sophisticated phishing campaigns. Thus, there's a pressing need for automated systems capable of swiftly identifying phishing websites to safeguard users' online security.

In this paper we aim to address the challenges posed by phishing websites through the following contributions:

- Data Collection: By gathering datasets containing both phishing and legitimate URLs from reputable sources, we establish a robust foundation for training and evaluating our machine learning models.

- Feature Extraction: Leveraging insights from previous research and industry standards, we extract a comprehensive set of features from URLs, encompassing address bar, domain, HTML, and JavaScript attributes. These features serve as essential indicators for distinguishing between phishing and legitimate websites.
- Model Training: Employing a suite of supervised machine learning algorithms and deep neural networks, including Decision Trees, Random Forests, Multilayer Perceptrons, XGBoost, and Support Vector Machines, we train models to classify URLs accurately based on their phishing status.
- Evaluation and Selection: Through rigorous evaluation using appropriate metrics, we identify the most effective model for phishing detection.

3.PROBLEM STATEMENT

a. Problem Definition:

The proliferation of phishing websites poses a significant threat to individuals, organizations, and society as a whole. Phishing is a form of cybercrime where malicious actors attempt to deceive users into divulging sensitive information such as login credentials, financial details, or personal data by masquerading as trustworthy entities. Detecting phishing websites is challenging due to their deceptive nature, as they often mimic legitimate URLs and webpages.

The objective of this project is to develop an effective system for detecting phishing websites using machine learning techniques. By analyzing various features extracted from URLs, such as address bar-based features, domain-based features, and HTML/JavaScript-based features, the system aims to accurately classify URLs as either phishing or legitimate. The problem involves building and training classification models capable of accurately distinguishing between phishing and legitimate URLs based on their features.

b. Motivations:

Our motivation stemmed from the acknowledgment of the pressing need to combat phishing attacks effectively in today's digital era, as highlighted by the research presented in the paper [1] by Arvind Prasad and Shalini Chandra. Building upon the foundation laid by their work, we developed our feature extraction technique to construct a comprehensive phishing URL dataset. Recognizing the evolving nature of cyber threats and the limitations in existing detection frameworks, we aimed to enhance the efficacy of phishing URL detection by devising novel methods to extract URL and HTML features. By leveraging insights from the aforementioned paper, we designed our approach to address the challenges posed by visual similarity-based attacks and the need for continuous learning to stay ahead of emerging phishing techniques. Through meticulous feature engineering and dataset construction, we endeavored to contribute to the advancement of phishing detection methodologies, thereby fortifying the resilience of individuals and organizations against cyber threats.

The problem of phishing website detection is of paramount importance in the realm of cybersecurity due to its widespread prevalence and potential for causing significant harm. Phishing attacks have been responsible for a myriad of security breaches, financial losses, and privacy violations across individuals, businesses, and governments worldwide. Therefore, developing robust and reliable methods for detecting phishing websites is crucial in safeguarding users' online security and privacy.

Moreover, with the increasing sophistication of phishing techniques and the continuous evolution of malicious strategies employed by cybercriminals, traditional approaches to phishing detection may become ineffective. Machine learning offers a promising avenue for addressing this challenge by leveraging computational algorithms to analyze large datasets of URL features and identify patterns indicative of phishing behavior. By harnessing the power of machine learning, we can enhance the efficiency and accuracy of phishing detection systems, thereby bolstering cybersecurity defenses.

c. Justifications:

The need for an effective phishing detection system is underscored by the escalating threat landscape of cyberattacks, where phishing remains a prevalent and pervasive tactic employed by malicious actors. Traditional approaches to combating phishing, such as blacklisting known phishing URLs or relying on heuristic-based detection methods, are often insufficient to keep pace with the dynamic and evolving nature of phishing attacks.

Machine learning-based approaches offer several advantages in this context. They can autonomously learn and adapt to new patterns and trends in phishing behavior, thereby providing more robust and scalable solutions. By analyzing a diverse array of features extracted from URLs, machine learning models can discern subtle differences between phishing and legitimate websites that may elude conventional detection methods.

Furthermore, the development of an accurate and reliable phishing detection system holds significant implications for cybersecurity across various sectors, including finance, healthcare, government, and e-commerce. By proactively identifying and mitigating phishing threats, organizations can minimize the risk of data breaches, financial fraud, and reputational damage, thus safeguarding both their own interests and those of their customers and stakeholders.

In summary, the proposed solution leveraging machine learning techniques for phishing website detection addresses a critical cybersecurity challenge, offering enhanced detection capabilities, adaptability to evolving threats, and broader applicability across diverse industries and user contexts.

4.RELATED WORKS

Prasad and Chandra proposed PhiUSIIL [1], a phishing URL detection framework that addresses the growing concern of phishing attacks in today's digital landscape. The framework leverages a combination of two key components: a URL similarity index and incremental learning [1]. By integrating these elements, PhiUSIIL offers a robust and adaptive approach to real-time phishing URL detection. Notably, PhiUSIIL's innovation lies in its development of a diverse security profile, which caters to the varied security requirements of users and organizations. The framework's effectiveness is underscored by its ability to identify visual similarity-based attacks, such as zero-width characters, homograph, punycode, homophone, bit squatting, and combosquatting attacks, through the similarity index [1]. Additionally, the construction of the PhiUSIIL phishing URL dataset, comprising both legitimate and phishing URLs, further enhances the framework's detection accuracy.

The PhiUSIIL phishing URL dataset construction module involves collecting legitimate URLs from the Open PageRank Initiative and phishing URLs from sources like PhishTank, OpenPhish, and MalwareWorld. The framework employs a systematic approach to download phishing webpages programmatically before they are blocked or removed, ensuring the dataset's comprehensiveness. Key features critical to detecting phishing URLs, such as TLD, URLLength, IsDomainIP, NoOfSubDomain, and NoOfObfuscatedChar [1], are extracted from the URLs . Additionally, HTML features, including LargestLineLength, HasTitle, HasFavicon, IsResponsive, and NoOfURLRedirect [1], are extracted from the webpage HTML code. Through meticulous feature extraction and dataset construction, PhiUSIIL lays a solid foundation for effective phishing detection and showcases promising results in combating evolving cyber threats.

Mohammad et al. (2012) present an automated technique for assessing features related to phishing websites, aiming to enhance the efficacy of phishing detection. Phishing attacks, which aim to obtain confidential information from users by impersonating legitimate entities, pose significant threats in the online trading domain. Traditional methods of phishing detection often rely on blacklists or heuristic-based approaches [2], both of which have limitations in detecting newly created phishing websites in real-time. To address this challenge, the authors propose a novel approach wherein a set of features crucial for distinguishing phishing websites from legitimate ones is automatically extracted using a custom software tool [2]. Unlike manual extraction methods, which require extensive knowledge of evolving phishing techniques, automatic extraction offers the advantage of scalability and efficiency. By analyzing webpage properties derived from HTML tags, URL addresses, and JavaScript source code [2], the proposed method aims to identify discriminative patterns used by phishers.

The motivation behind developing a set of rules for automatic feature extraction is twofold: to reduce the false negative rate in phishing detection and to expedite the extraction process, thereby increasing the dataset size for more comprehensive experiments [2]. By leveraging newly proposed rules developed experimentally, the authors seek to improve the accuracy of phishing detection algorithms. The research questions addressed in this article include identifying the effective minimal sets of features for predicting phishing and suggesting new rules for automatic feature extraction. Through a systematic exploration of related works and a comparison of different phishing extraction methods, the authors lay the groundwork for their proposed approach. The article provides insights into the significance of the proposed features in detecting phishing websites, thereby contributing to the ongoing efforts to combat online threats and enhance cybersecurity measures.

Pan and Ding (2006) propose an anomaly-based approach for detecting phishing web pages, aiming to address the inherent adaptability of phishing attackers. Unlike traditional anti-phishing schemes, which often struggle to mitigate the evolving tactics of attackers, their method focuses on identifying anomalies in web pages without relying on specific phishing implementations [3]. By examining the structural features and HTTP transactions of web pages, their approach aims to detect discrepancies between a website's claimed identity and its actual characteristics [3]. This detection method demands neither user expertise nor prior knowledge of the website, offering a promising avenue for combating phishing attacks. The experimental results demonstrate the effectiveness of their phishing detector in achieving low miss rates and false-positive rates.

5.METHODOLOGY

a. Material and Data:

The material and data for this project are crucial components in the development of a robust phishing website detection system. The dataset consists of both legitimate and phishing URLs collected from reliable sources such as PhishTank and the University of New Brunswick. PhishTank provides real-time updates on phishing URLs, ensuring the dataset's relevance and timeliness. Additionally, the dataset includes various features extracted from the URLs, encompassing address bar-based, domain-based, and HTML & JavaScript-based features. These features provide valuable insights into the characteristics of both legitimate and phishing URLs, aiding in the development of effective detection models.

b. Proposed Methods / Solutions / Algorithms / Models:

The proposed methodology employs a combination of feature extraction techniques and machine learning algorithms to detect phishing websites. Feature extraction involves categorizing URLs based on address bar features, domain-based features, and HTML & JavaScript-based features. Each category encompasses specific features that capture unique aspects of URLs indicative of phishing behavior. These features serve as input variables for various machine learning models, including Decision Trees, Random Forests, Multilayer Perceptrons, XGBoost, and Support Vector Machines (SVM).

1. Feature Extraction:

- Address Bar Features: Extracted features include domain presence, IP address, "@" symbol, URL length, depth, redirection, HTTP/HTTPS in domain name, URL shortening services, and prefix/suffix "-"
- Domain-based Features: Features include DNS record, website traffic, age of domain, and end period of domain.
- HTML & JavaScript-based Features: Features consist of iframe redirection, status bar customization, disabling right-click, and website forwarding.

2. Machine Learning Models:

- Decision Tree: Utilized for hierarchical classification based on if/else questions.
- Random Forest: Ensemble method of decision trees to reduce overfitting and improve accuracy.
- Multilayer Perceptrons (MLPs): Deep neural networks for nonlinear classification.
- XGBoost: Gradient boosting algorithm known for its speed and performance.
- Support Vector Machines (SVM): Binary linear classifier for separating data points in high-dimensional space.

c. Conditions and Assumptions:

- The dataset is assumed to be representative of real-world phishing scenarios, containing a balanced distribution of legitimate and phishing URLs.
- It is assumed that the extracted features effectively capture distinguishing characteristics between legitimate and phishing URLs.
- The performance evaluation of machine learning models assumes that the training and testing datasets are randomly shuffled to avoid bias.
- The models are trained and evaluated using standard machine learning metrics such as accuracy, precision, recall, and F1 score.

d. Formal Complexity or Simulation Analysis:

- Formal complexity analysis involves assessing the computational complexity of feature extraction algorithms and machine learning models.
- The complexity analysis may include time and space complexity evaluations for feature extraction techniques such as string parsing, regular expressions, and data manipulation.
- Simulation analysis involves training and testing machine learning models on different subsets of the dataset to evaluate performance under varying conditions, such as imbalanced data, feature importance, and model hyperparameters.
- Additionally, cross-validation techniques like k-fold cross-validation can be employed to assess the generalization performance of the models.

Overall, the methodology combines data collection, feature extraction, and machine learning modeling to develop an effective phishing website detection system capable of accurately distinguishing between legitimate and malicious URLs. Through rigorous evaluation and analysis, the proposed approach aims to provide reliable protection against phishing attacks in real-world scenarios.

6.COMPUTATIONAL EXPERIMENTS

In this study, computational experiments were undertaken to develop, train, and assess a predictive model tailored for the identification of phishing websites. The dataset employed consisted of approximately 10,000 instances, comprising features derived from both phishing and legitimate website sources. Specifically, a sample of 5,000 random phishing URLs was sourced from the Phishtank repository, while an equivalent number of legitimate URLs were obtained from a database maintained by the University of New Brunswick.

The dataset encompassed various categories of features extracted from the URLs, including characteristics derived from the address bar, domain attributes, and elements extracted from the HTML, CSS, and JavaScript components of the websites.

The principal objective of the computational experiments was to leverage machine learning (ML) algorithms to develop a robust predictor capable of accurately distinguishing between legitimate and phishing websites based on these diverse features. Additionally, the study aimed to provide a comparative analysis of different ML models to evaluate their performance in this task.

Through systematic experimentation and rigorous evaluation, this research sought to contribute insights into the efficacy of ML-based approaches for combating online phishing threats and enhancing cybersecurity measures.

a. Experiments

The dataset comprising both legitimate and phishing URLs was meticulously curated to ensure a robust representation of web traffic diversity. The collection of phishing URLs was facilitated by leveraging the PhishTank service [6], an open-source repository that offers a comprehensive repository of phishing URLs, updated hourly. The dataset was acquired in CSV format via a straightforward retrieval process from the PhishTank website. For legitimate URLs, a dataset sourced from the University of New Brunswick [7], encompassing benign, spam, phishing, malware, and defacement URLs, was utilized.

Upon data acquisition, the dataset was carefully processed to ensure an appropriate balance between phishing and legitimate instances. To mitigate the risk of data imbalance and maintain consistency in the dataset, a stratified sampling approach was adopted. Specifically, a total of 10,000 instances were selected, with 5,000 instances representing phishing URLs and an equivalent number representing legitimate URLs.

Next, a diverse set of features was extracted from the URLs dataset, categorized into address bar-based features, domain-based features, and HTML & JavaScript-based features. For brevity, a subset of features was selected for inclusion in the analysis, focusing on those deemed most indicative of phishing behavior. Address bar-based features included domain presence, IP address presence, '@' symbol presence, URL length, URL depth, redirection presence, presence of 'http/https' in the domain name, URL shortening services usage, and presence of '-' in the domain name. Similarly, domain-based features included DNS record availability, website traffic rank, domain age, and remaining domain time. HTML & JavaScript-based features encompass iframe presence, status bar customization, disabling right-click functionality, and website forwarding.

While the exhaustive list of features underscores the depth of analysis undertaken, for the sake of conciseness and relevance, only a select subset of features were incorporated into the subsequent modeling phase. These features were deemed most informative in discerning between phishing and legitimate URLs, thereby enhancing the predictive capabilities of the machine learning models.

In the process of model training, the dataset was initially loaded into the notebook after feature extraction. Subsequently, a thorough examination of the dataset was conducted to gain insights into its structure and features. Visualizations such as plots and graphs were employed to explore the distribution of data and relationships between different features.

Following data exploration, preprocessing techniques were applied to clean and transform the dataset, ensuring its suitability for model training. Notably, redundant features, such as the 'Domain' column, were identified and removed to streamline the dataset. Additionally, efforts were made to address any class imbalances and ensure a representative distribution of data for training and testing.

The dataset was then split into training and testing sets using an 80-20 split ratio, with 8,000 instances allocated for training and 2,000 instances for testing. Supervised machine learning models, specifically classification models, were considered for training on the dataset. These models included Decision Trees, Random Forests, Multilayer Perceptrons (MLPs), XGBoost, and Support Vector Machines (SVMs).

Each model was evaluated based on its performance in classifying URLs as phishing or legitimate. For instance, Decision Tree classifiers were explored for their ability to learn hierarchical if/else questions to make predictions, while Random Forest classifiers were employed to harness the collective decision-making power of multiple decision trees. Similarly, MLPs, XGBoost, and SVMs were considered for their efficacy in handling classification tasks.

To facilitate comparison between the performance of different models, a structured approach was adopted. Results from each model were stored in a dataframe, allowing for easy comparison of metrics such as accuracy, precision, recall, and F1-score. This comprehensive evaluation framework enabled the identification of the most suitable model for the task of phishing website detection, thereby contributing valuable insights to the field of cybersecurity.

b. Evaluation Metrics

The performance of the trained machine learning models was rigorously evaluated using a range of standard evaluation metrics tailored for classification tasks. Key metrics employed included accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correctly classified instances out of the total number of instances. Precision quantifies the proportion of true positive predictions among all instances predicted as positive, highlighting the model's ability to avoid false positives. Recall, also known as sensitivity, represents the proportion of true positive predictions among all actual positive instances, indicating the model's effectiveness in capturing positive instances. The F1-score, the harmonic mean of precision and recall, provides a balanced measure of the model's performance, particularly useful in scenarios with class imbalance. Additionally, receiver operating characteristic (ROC) curves and area under the ROC curve (ROC AUC) were utilized to visualize the trade-offs between true positive and false positive rates across different classification thresholds. This comprehensive evaluation framework facilitated a nuanced understanding of the models' performance and enabled informed decision-making regarding their suitability for phishing website detection.

c.Implementation Details

The implementation of the machine learning models and feature extraction was conducted using Google Colab, providing a convenient and scalable environment for model training. A variety of Python libraries were utilized throughout the project, including pandas and numpy for data manipulation, seaborn and matplotlib for data visualization, and scikit-learn for model training and evaluation. The ensemble methods of Decision Trees and Random Forests were implemented using sklearn's Decision Tree Classifier and Random Forest Classifier, respectively, while MLPClassifier from scikit-learn was employed for neural network-based classification. Additionally, XGBClassifier from the XGBoost library was utilized for gradient boosting-based classification. Support Vector Machine (SVM) models were constructed using SVC from scikit-learn. For feature extraction and preprocessing, packages such as urllib, BeautifulSoup, and whois were employed to extract relevant information from URLs, including domain attributes and HTML content. Overall, the combination of these tools and libraries provided a robust framework for the development and evaluation of the predictive models for phishing website detection.

d. Results

Different Visualization techniques were used to understand the features and data.

Histograms were generated to visualize the distribution of features in the dataset, with respect to the target label (phishing or legitimate).

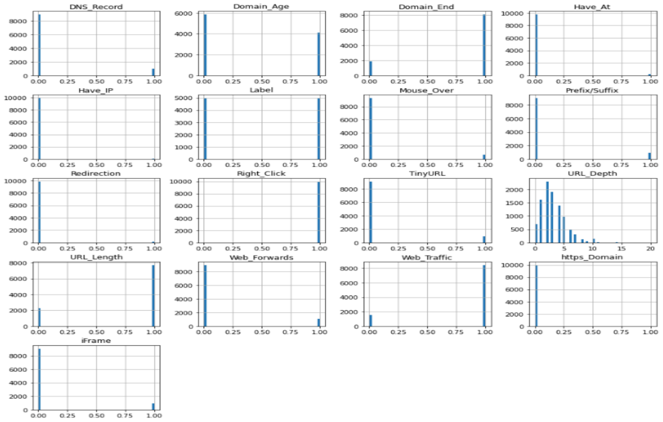

Figure: Histogram of different feature wrt target variable.

Additionally, a heatmap was created to explore the correlations between different features, providing insights into potential patterns and relationships within the dataset.

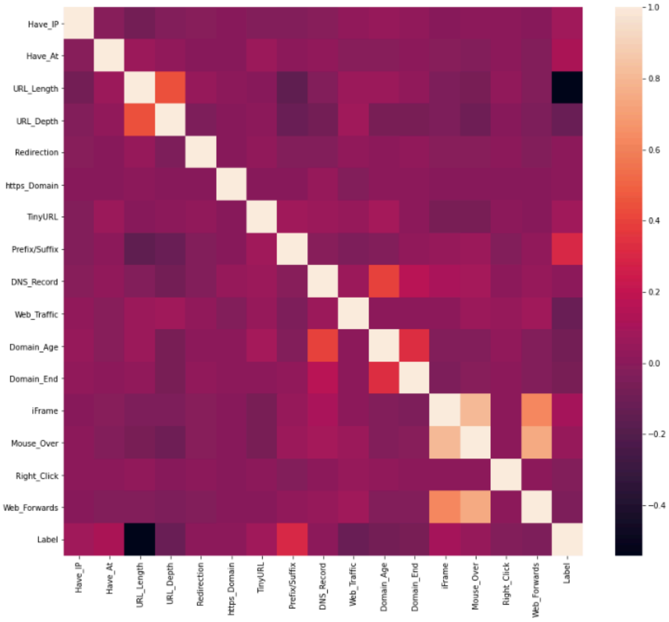

Figure: Heatmap of different features.

The performance of various machine learning models in classifying phishing and legitimate URLs was evaluated using both training and test data. The table below summarizes the accuracy scores achieved by each model on both training and test datasets:

ML Model	Train Accuracy	Test Accuracy
Decision Tree	0.810	0.826
Random Forest	0.814	0.834
Multilayer Perceptrons	0.858	0.863
XGBoost	0.866	0.864
SVM	0.798	0.818

The Multilayer Perceptrons (MLP) and XGBoost models demonstrated the highest performance on both training and test datasets, achieving accuracies of 85.8% and 86.6% on training data, and 86.3% and 86.4% on test data, respectively.

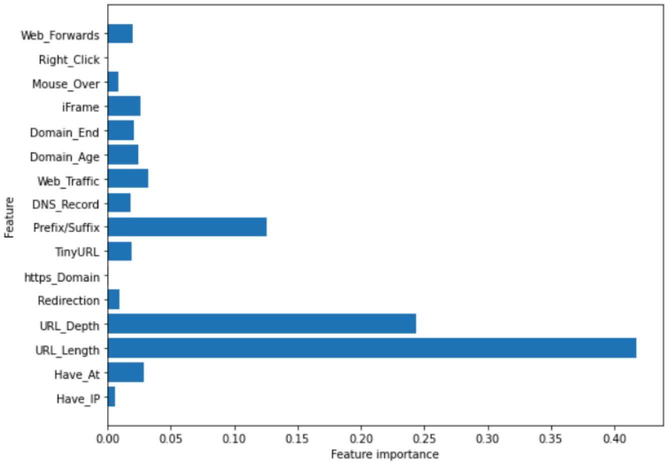

Figure: Feature Importance as per Random Forest.

A feature importance analysis was conducted to assess the significance of different features in the Random Forest classifier. The visualization presented a horizontal bar plot, with each bar representing the importance of a specific feature in the model's decision-making process. Features contributing more significantly to the model's predictive performance were assigned higher importance scores. This analysis aids in identifying key features that play a crucial role in distinguishing between phishing and legitimate URLs, thereby informing feature selection and enhancing model interpretability.

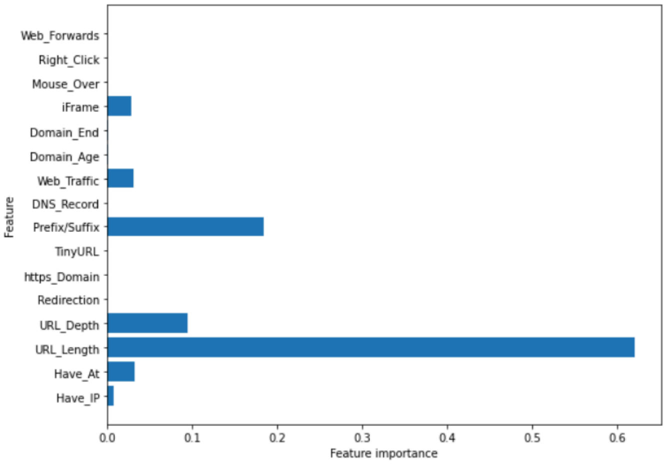

Figure: Feature Importance as per Decision Tree.

A similar feature importance analysis was conducted for the Decision Tree classifier. The visualization utilized a horizontal bar plot to display the importance of each feature in the model's decision-making process. Each bar represented the significance of a specific feature, with higher bars indicating greater importance. This analysis offers insights into the key features utilized by the Decision Tree model to differentiate between phishing and legitimate URLs, aiding in feature selection and model interpretation.

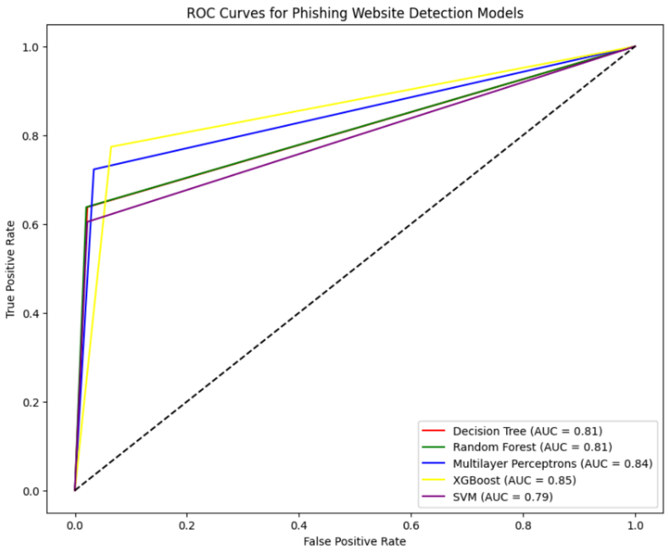

Figure: ROC Curves for different models.

Receiver Operating Characteristic (ROC) curves were generated for each machine learning model to evaluate their performance in distinguishing between phishing and legitimate URLs. The ROC curves plot the true positive rate (sensitivity) against the false positive rate (1-specificity) for different classification thresholds. A model with higher area under the ROC curve (ROC AUC) indicates better discrimination ability. The ROC curves for all models provide a comprehensive visualization of their performance across various classification thresholds, enabling comparison of their overall effectiveness in classifying URLs. This analysis aids in selecting the most suitable model for phishing website detection based on its discriminatory power and trade-offs between true positive and false positive rates.

e. Discussions

The results of this study demonstrate the efficacy of various machine learning models in the task of phishing website detection. The Multilayer Perceptrons (MLP) and XGBoost models emerged as top performers, achieving high accuracy scores on both training and test datasets. The robust performance of these models underscores the effectiveness of neural network-based and gradient boosting approaches in capturing complex patterns inherent in phishing URLs. Additionally, the feature importance analysis revealed key features that significantly influence the models' predictive performance, such as URL length and presence of specific characters. The visualization of ROC curves provided insights into the trade-offs between true positive and false positive rates across different classification thresholds, aiding in model selection and evaluation. Furthermore, the heatmap visualization offered valuable insights into the correlations between different features, highlighting potential relationships and informing feature engineering strategies. Overall, the findings of this study contribute to the advancement of cybersecurity efforts by providing a framework for effective phishing website detection using machine learning techniques.

7.CONCLUSION

a. Summary:

In this project, we aimed to tackle the issue of phishing website detection using machine learning techniques. We started by collecting a dataset consisting of both phishing and legitimate URLs, extracting relevant features from these URLs, and then training various machine learning models to classify them. The features extracted included address bar-based, domain-based, and HTML & JavaScript-based features, covering a wide range of characteristics that could distinguish between phishing and legitimate websites.

We explored several machine learning models including Decision Trees, Random Forests, Multilayer Perceptrons, XGBoost, and Support Vector Machines. Through training and evaluation, we found that XGBoost performed the best among the tested models, exhibiting high accuracy both on the training and testing datasets.

b. Future Research:

Despite the success of our models, there is still room for improvement and avenues for future research. One area of potential enhancement lies in feature extraction. While we covered a diverse set of features in this project, there may be additional characteristics of URLs and web content that could be informative for phishing detection. Future research could focus on identifying and extracting such features, potentially utilizing more advanced techniques such as natural language processing (NLP) for analyzing website content.

Moreover, incorporating real-time data sources for phishing URLs, rather than relying solely on static datasets, could enhance the robustness of the models. Dynamic feature extraction techniques that capture temporal and contextual information about URLs and website behavior could also be explored.

Furthermore, exploring ensemble methods or hybrid models that combine the strengths of multiple algorithms could potentially yield even better performance. Additionally, techniques such as active learning and semi-supervised learning could be investigated to efficiently utilize limited labeled data and adapt to evolving phishing tactics.

c. Open Problems:

Despite the progress made in phishing website detection, several challenges and open problems remain. One ongoing issue is the cat-and-mouse game between cybercriminals and security measures. As phishing techniques evolve and become more sophisticated, detection algorithms must continuously adapt to stay effective.

Another challenge is the presence of highly targeted and tailored phishing attacks, known as spear phishing, which are often harder to detect using traditional methods. Developing techniques capable of detecting such attacks, possibly by incorporating user behavior analysis or contextual information, remains an open problem.

Additionally, the ethical implications of deploying automated phishing detection systems raise important questions regarding privacy, data security, and potential biases in algorithmic decision-making. Addressing these concerns and ensuring the responsible and transparent use of such technologies is crucial for the future development and deployment of phishing detection systems.

8.REFERENCES

[1] Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 136, 103545-.
https://doi.org/10.1016/j.cose.2023.103545

[2] R. M. Mohammad, F. Thabtah and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," 2012 International Conference for Internet Technology and Secured Transactions, London, UK, 2012, pp. 492-497.

[3] Y. Pan and X. Ding, "Anomaly Based Web Phishing Page Detection," 2006 22nd Annual Computer Security Applications Conference (ACSAC'06), Miami Beach, FL, USA, 2006, pp. 381-392, doi: 10.1109/ACSAC.2006.13.

[4]https://machinelearningmastery.com/save-gradient-boosting-models-xgboost-python/

[5]https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5

[6]https://www.phishtank.com/developer_info.php

[7]https://www.unb.ca/cic/datasets/url-2016.html