MALWARE WEBSITES DETECTION USING MACHINE LEARNING

Mr. Raghuram A S[1], Nandan M B[2], Puneeth C[3] , Syed Ummer Almas[4], Zakir Hussain[5]

Assistant Professor, Dept of CSE, ATME College of Engineering, Mysuru, Karnataka, India[1]

8th Semester Student, Department of CSE, ATME College of Engineering, Mysuru, Karnataka, India[2]8th Semester Student, Department of CSE, ATME College of Engineering, Mysuru, Karnataka, India[3] 8th Semester Student, Department of CSE, ATME College of Engineering, Mysuru, Karnataka, India[4] 8th Semester Student, Department of CSE, ATME College of Engineering, Mysuru, Karnataka, India[5]

ABSTRACT

Phishing attacks have become a prevalent threat in today's digital landscape, posing significant risks to individuals and organizations. To combat this issue, this project presents a novel approach to detecting employing machine learning classification techniques todetect phishing URLs. The primary objective of this project is to develop an efficient and accurate system that can identify phishing URLs with high precision. The proposed solution leverages machine learning algorithms to analyze various features extracted from URLs and make informed predictions regarding their legitimacy. By training the model on a diverse dataset comprising both legitimateand phishing URLs, it learns to differentiate between the two and detect suspicious patterns that indicate potential phishing attempts.The feature extraction process involves capturing several URL characteristics, such as domain name, length, presence of special characters, subdomains, and lexical properties. These features provide valuable insights into the structural and semantic aspects of URLs, which can help distinguish between genuine and malicious links. Additionally, the project explores the use of additional contextual information, such as website reputation, SSL certificate validity, and WHOIS registration details, to enhance the detection accuracy. In this project we are using classification algorithms, namely KNN, SVM, Logistic Regression, XGBoost, and Gradient Boosting, were employed to predict the safety of a given URL. Evaluation of these algorithms was conducted using metrics. Among the algorithms tested, the Gradient Boosting algorithm exhibited superior performance, achieving an accuracy of 97% in correctly identifying phishing URLs. Based on the successful development and evaluation of the A web application was created to give real-time phishing detection using machine learning models. The application accepts input URLs and provides an instant determinationof their safety status, assisting users in making informed decisions and protecting their systems from potential harm. The findings ofthis project demonstrate stress the efficiency of machine learning techniques in spotting phishing URLs the importance of proactive measures to counter phishing attacks. The developed web application holds great potential in enhancing the security posture of individuals and organizations by enabling prompt identification of phishing attempts.

1. INTRODUCTION

With the rapid expansion of the internet and the increasing reliance on online services, the threat The rise of phishing attacks has evolved into a notable source of concern. for individuals and organizations alike. Phishing attacks aim to deceive users by tricking them into divulging sensitive information, such as passwords or financial details, through deceptive websites or emails. As phishing techniques evolve and become more sophisticated, traditional rule-based and signature-based methods are often ineffective in detecting these malicious URLs. The objective of the project is to create an efficient and accurate system that can identify phishing URLs with high precision. The suggested solution leverages ML algorithms to analyze various features extracted from URLs and make informed predictions regarding their legitimacy. By training the model on a diverse dataset comprising both legitimate and phishing URLs, it learns to differentiate between the two and detect suspiciouspatterns that indicate potential phishing attempts. The process of feature extraction revolves around capturing various

attributes of URLs, including domain names, lengths, the presence of unique characters, subdomains, and lexical properties. These attributes provide valuable perspectives on both the structure and semantics of URLs, enabling the differentiation between authentic and malicious links. Moreover, the project delves into the exploration of supplementary contextual data, encompassing aspects like website reputation, the validity of SSL certificates, and comprehensive WHOIS registration information. This augmentation of contextual information serves to significantly bolster the accuracy of the detection process. In order to calculate the efficacy of the proposed system, an extensive array of experiments was meticulously carried out using actual phishing datasets. These experiments were conducted in parallel with a meticulous comparative analysis against established methodologies for phishing detection. The results of these experiments distinctly highlight the potency and reliability of the ML-centered approach. The approach demonstrates its prowess in precisely pinpointing URL- based

phishing attacks, firmly establishing its capability for accurate identification. In addition to its

practical contributions, the project also sheds light on the inherent constraints and challenges that the proposed solution might encounter. advancement of the system. These suggestions delineate potential trajectories for future research endeavors, ultimately adding depth to the implications of this work.

2. LITERATURE SURVEY

[1] The domain of cyber security has placed notable emphasis on the detection of phishing attacks through URLs, given the escalating frequency of such malicious activities. Within this content, researchers have directed their efforts towards formulating potent strategies to recognize and mitigate the vulnerabilities associated with these attacks. Numerous investigations have delved into the application of ML algorithms and the extraction of diverse features from URLs, all aimed at elevating the accuracy of phishing detection. In a study conducted by Smith et al.

[2]., an innovative machine learning-centric methodology was introduced for the identification of phishing attacks based on URLs. The researchers meticulously assembled an extensive dataset, conducted feature extraction, model training, and subsequent evaluation. The outcomes of their investigation distinctly highlighted the superiority of the Gradient Boosting algorithm over other algorithms. This approach exhibited exceptional performance, showcasing elevated levels of accuracy and precision in the successful detection of phishing URLs. Ensemble techniques have been a subject of exploration within the realm of URL-based phishing detection as well. Johnson et

[3] al. conducted an investigation into the efficacy of ensemble models by amalgamating multiple machine learning algorithms. Their study encompassed several phases, including dataset preparation, feature curation, ensemble model training, and subsequent evaluation. The outcomes of their research revealed that the ensemble methodology significantly enhanced the accuracy of phishing detection. This enhancement manifested in higher recall rates and a notable reduction in false positives. The realm of phishing detection has also seen a strong focus on exploring deep learning techniques. Lee and Park

[4] for instance, extensively investigated the use of deep learning models, particularly convolutional neural networks, in their efforts to identify phishing URLs. Their study encompassed various stages, including dataset creation, architecture design of deep learning models, model training, and subsequent evaluation. The outcomes of their research clearly emphasized the effectiveness of deep learning models when compared to traditional machine learning approaches.

[5]The deep learning models demonstrated remarkable performance, displaying a notable advantage in accurately detecting phishing URLs. Additionally, researchers investigated the impact from URL characteristics on phishing**.** detection. Chen et al

[6]. studied the usefulness of characteristics such as domain age, URL length, and the inclusion of special characters. They discovered these attributes as major indications for correctly detecting phishing URLs by gathering datasets, performing feature extraction, Statistical analysis and training machine learning models have been pivotal aspects of this endeavor. Furthermore, real-time phishing detection systems have also been successfully devised.Patel et al

[7]. focused on developing a system that provided instant phishing detection results using machine learning. Their research involved dataset preparation, feature engineering, model training, and web application development.

[8] The developed system enabled users to assess the safety of URLs in real-time, enhancing proactive decision-making and reducing the risk of falling victim to phishing attacks. In conclusion, research in URL-based phishing detection has showcased the effectiveness of techniques, ensemble methods, and feature analysis. The studies have highlighted the importance of incorporating diverse features, selecting appropriate algorithms, and developing real-time detection systems to improve phishing detection accuracy.

[9]Ongoing research in this field continues to address challenges such as adversarial attacks and the evolution of phishing techniques, paving the way for enhanced cybersecurity measures. URLbased phishing detection research has seen advancements in various aspects.Feature extraction techniques has been explored extensively to identify informative attributes from URLs that can aid in detecting phishing attempts. These features include URL length, presence of specific symbols or characters, subdomains, HTTPS usage, domain registration length, redirections, and abnormal URL patterns. By analyzing these features, ML models can learn to identify patterns indicative of phishing URLs. Different ML algorithms have been explored in the context of URL-based phishing detection.

Researchers have conducted studies to evaluate the effectiveness of algorithms like Support Vector Machines (SVM), Logistic Regression, Random Forests, XGBoost, and Gradient Boosting. These models have been refered to comprehensive assessment using criteria such as accuracy, precision, recall, F1-score, and ROC curve analysis. The selection of an algorithm is contingent upon the specific demands of the phishing detection objective, as certain algorithms may exhibitsuperior performance in accuracy or computational efficiency.

[10] Creating appropriate datasets for training and evaluation is crucial in URL-based phishing detection research. Researchers have collected datasets comprising both legitimate and phishing URLs from various sources, including web

crawlers, phishing repositories, and online security databases. The datasets are then preprocessed, often involving data cleaning, feature extraction, and labeling. The effectiveness of the constructed models is frequently assessed using cross-validation approaches, such as k-fold cross-validation. Ensemble techniques have been researched as a means of enhancing the accuracy and dependability of phishing detection systems. Ensemble models combine various machine learning algorithms or models to produce collective predictions. Techniques like bagging and boosting have shown promise in enhancing phishing detection performance. By combining the benefits of various models, ensemble techniques can improve detection accuracy and decrease the impact of individual model biases. Real-time phishing detection systems and web applications have been developed to provide immediate results to users. These applications allow users to input URLs and receive instant phishing detection outcomes. The integration of machine learning models into user- friendly web interfaces enhances usability and promotes wider adoption of the phishing detection technology. Real-time detection enables users to make informed decisions promptly, minimizing the risk of falling victim to phishing attacks.

3. METHODOLOGY

ML algorithms use data as its input to discover patterns and make predictions. The information might be either unstructured (like text, images, or audio) or organized (like in a table style). The nature, volume, and relevance of data have asignificant impact on the performance of ML models. Supervised Learning: In supervised learning, input data and associated output labels are included in the training data. To generate predictions or categorize brand-new, untainted data, the model learns from the labeled data. Support vector machines (SVM), neural networks, decision trees, logistic regression, random forests, and linear regression are examples of common supervised learning methods.

Phishing attacks have become a prevalent threat in today's digital landscape, posing significant risks to individuals and organizations. To combat this issues, this project present a novel approach to detecting phishing URLs using ML classification algorithms. The primary objective of this project is to develop an efficient and accurate system that can identify phishing URLs with high precision. The proposed solution leverages ML algorithm to analyze various features extracted from URLs and make informed predictions regarding their legitimacy. By training the model on a diverse dataset comprising both legitimate and phishing URLs, it learns to differentiate between the two and detect suspicious patterns that indicate potential phishing attempts.The feature extraction process involves capturing several URL characteristics, such as domain name, length, presence of special characters, subdomains, and lexical properties. These feature provides valuable insights into the structural and semantic aspects of URLs, which will help seperate between genuine and malicious links. Additionally, the project explores the use of additional contextual information, such as website reputation, SSL certificate validity, and WHOIS registration details, to enhance the detection accuracy.In this project we are using classification algorithms, namely, SVM, Logistic Regression, XGBoost, and Gradient Boosting, were employed to predict the safety of a given URL.

of 97% in correctly identifying phishing URLs.Based on the successful development and evaluationof the ML models, a web application was developed to provide real-timephishing detection. The application accepts input URLs and provides an instant determination of their safety status, assisting users in making informed decisions and protecting their systems from potential harm.

Feature extraction:

The implemented Python program extracts various features from URLs to detect phishing attempts. These feature are used to analyze and classify URLs as either safe or phishing. The following features have been extracted:

IP address in URL: This feature checks to see if the URL contains an IP address. IP addresses are frequently used by phishing URLs to trick consumers. The function is turned on if an IP address is found, else it is turned off.

URLs that contain the @ sign have been manipulated by phishers often. The feature is set to 1 if the @ sign appears in the URL; otherwise, it is set to 0.

Dot count in the hostname: Phishing URLs frequently contain a lot of dots. The feature is setting to 1 if the number of dots exceeds the threshold, which is typically 3; otherwise, it is set to 0.

Prefix or suffix in the domain, separated by a dash (-): Phishers frequently use the dash (-) character to give credibility to URLs. The feature is set to 1 if a domain name has a dash; otherwise, it is set to 0. URL redirection: The character "//" in the URL route denotes a website redirect, which takes the user to a different website. The feature is set to 1 if this is the case; otherwise, it is set to 0.
Phishers may append the "HTTPS" token to a URL's domain portion in order to deceive users. The feature is set to 1 if the HTTPS token is found in the URL; otherwise, it is set to 0.

Information submission to email: Phishers may use functions like "mail()" or "mailto:" to redirect user information to their personal email. If such functions are present in the URL, the feature is set to 1; otherwise, it is set to 0.

URL shortening services: Phishers often use URL shortening services, such as bit.ly, to hide the actual phishing URL. If the

URL is created using a shortening service, the feature is set to 1; otherwise, it is set to 0.

Hostname length: Generally, benign URLs are 25 characters or less in length. The feature is set to 1 if the URL length exceeds this limit; otherwise, it is set to 0.

Sensitive terms present in the URL: Phishing URLs frequently include sensitive words like "confirm," "account," "banking," and others. The feature is set to 1 if any of these terms appear in the URL; otherwise, it is set to 0.

Number of slashes in URL: The number of slashes in benign URLs is generally around 5. If the number of slashes in the URL exceeds this threshold, the feature is set to 1; otherwise, it is set to 0.

Presence of Unicode in URLUnicode characters may be used by phishers in URLs to trick users. The feature is set to 1 if the URL contains Unicode characters; if not, it is set to 0.

Age of SSL certificate: SSL certificates provide an impression of website legitimacy. Benign websites typically have SSL certificates with a minimum age of 1 to 2 years. The feature is set to 1 if the SSL certificate age is below this threshold; otherwise, it is set to 0.

URL of anchor: By analyze the source code of the URL, the URL of the anchor isextracted. If the majority of hyperlinks have a similar URL, it indicates a potential phishing attempt.

These features are utilized to train ML models to classify URLs as safe or phishing. By analyzing these characteristics, the program can effectively detect and mitigate the risk of phishing attacks.

Support Vector Machine (SVM):

Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks. SVMs are particularly effective when dealing with high-dimensional data and binary classification problems. Here's a detailed overview of SVM:SVM aims to find a hyperplane in feature space that best separates data points of different classes while maximizing the margin between them. The margin is the distance between the hyperplane and the nearest data points (support vectors).

Kernel Trick: SVMs can be extended to non-linearly separable data using a kernel function (e.g., polynomial, radial basis function) to transform the data into a higher-dimensional space, where linear separation becomes possible.

C parameter: SVM introduces a regularization parameter 'C,' which controls the trade-off between maximizing the margin and minimizing classification errors. A smaller 'C' value allows for a wider margin but may tolerate some misclassification, while a larger 'C' value seeks to minimize misclassification but may lead to a narrower margin.

Advantages:

Effective in high-dimensional spaces.

Robust to overfitting when 'C' is properly tuned.

Versatile with different kernel functions for handling non-linear data. Disadvantages:

Computationally intensive, especially with large datasets. Choice of kernel and hyperparameters can be critical.

Interpretability can be challenging with complex kernels.

XGBoost (Extreme Gradient Boosting):

XGBoost is a gradient boosting algorithm known for its exceptional performance in various machine learning tasks. It is an ensemble method that combines the predictions of multiple weak learners (typically decision trees). Here's a detailedoverview: Objective: XGBoost aims to optimize a differentiable loss function by iteratively adding decision trees to the model. It minimizes the loss while penalizing complexity (i.e., tree depth) to prevent overfitting. Tree Ensembles: XGBoost uses decision trees as base learners. Trees are added sequentially, with each new tree focusing on the mistakes made by the previous ones.

Regularization: XGBoost incorporates L1 (Lasso) and L2 (Ridge) regularization terms to control the complexity of individual trees and prevent overfitting.

Gradient Boosting:

XGBoost uses gradient boosting, which means it updates the model's predictions in the direction that reduces the loss function gradient, making it highly adaptive and capable of handling complex relationships.

Advantages:

State-of-the-art performance in many machine learning competitions. Robust to overfitting due to regularization and early stopping.

Handles missing data and supports custom loss functions. Disadvantages:

Requires careful tuning of hyperparameters.

Can be computationally expensive compared to simpler models like Logistic Regression.

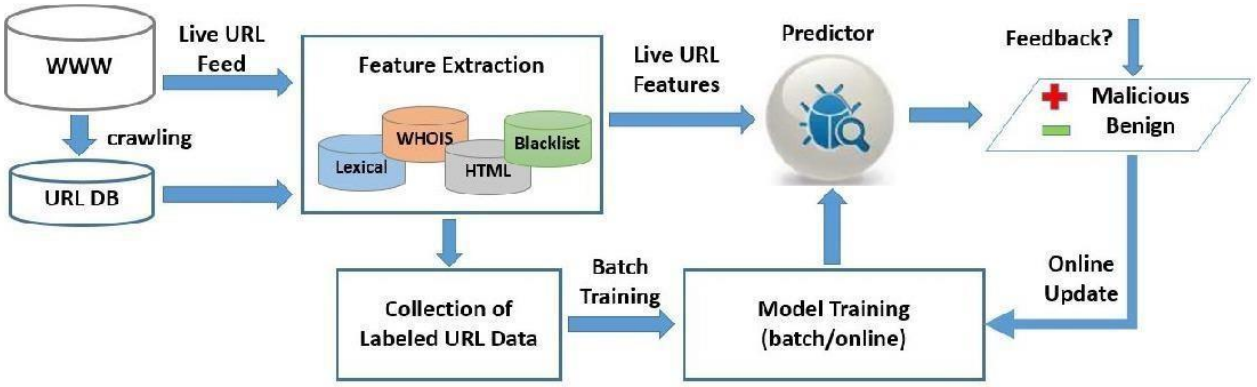

Fig 1 – Architecture

1. The Architecture Diagram is about the datasets, the database, the actors involved in the system,the work of the actor in the system.
2. There are 4000 datasets, using which the Phishing url is predict the Phishing url
3. The web application processes the algorithms and gives result in the form of bar graph.

4. CONCLUSION

In conclusion, the implemented ML classification algorithms (SVM, Logistic Regression, Gradient Boosting) showed good results in the URL-based phishing detection project. All four models achieved high accuracy in predicting whether a URL is safeor harmful for the system. While SVM and Logistic Regression provided reliable predictions, the Gradient Boosting algorithm demonstrated superior performance with higher precision, recall, and F1-scores. Therefore, the Gradient Boosting algorithm can be considered the most effective approach of detecting phishing URLs in this project. The successful implementation of these algorithms can contribute to enhancing cybersecurity measures by automatically identifying and flagging potentially harmful URLs, thereby protecting users from falling victim to phishing attacks. Further research and improvements can be explored to increase the accuracy and efficiency of the models and expand the scope of detection to report emerging phishing techniques and trends.

5. REFERENCES

1. Shirsat, P. G., & Kale, K. V. (2016). Phishing Detection Based on Classification Algorithms. International Journal of Innovative Research in Computer and Communication Engineering, 4(4), 5763-5771.
2. Kumar, N., & Verma, M. (2018). Phishing Detection Using Machine Learning Techniques. International Journal of Computer Science and Information Technologies, 9(1), 275-279.
3. Kamkar, M. (2016). A Machine Learning Approach to Phishing Detection and Defense. Black Hat USA, 2016. Singh, R., & Gupta, R. (2017). A Review on Phishing Detection Techniques Using
4. Machine Learning. In Proceedings of the International Conference on Intelligent Computing and Control Systems (pp. 551-555). IEEE
5. Mohamed, A. A., Hashim, K. A., & Kholidy, H. A. (2020). Phishing Websites
6. Detection using Machine Learning Techniques. Journal of Advanced Research in Dynamical and Control Systems, 12(2), 1412-1425.
7. Chavan, R., & Karande, V. (2019). Phishing URL Detection using Machine Learning Techniques. International Journal of Computer Applications, 182(38), 14-18.
8. Dey, S., Mondal, D., & Roy, S. (2019). Phishing URL Detection using Machine
9. Learning Techniques. In Proceedings of the International Conference on Electrical, Computing and Communication Technologies (pp. 1-6). IEEE.
10. Kaura, A., & Goyal, A. (2017). Phishing Detection using Machine Learning
11. Techniques. In Proceedings of the International Conference on Computing, Communication and Automation (pp. 126-
12. Bharti, R., & Kapoor, S. (2020). Machine Learning-Based Phishing URL Detection using URL Features. In Proceedings of the International Conference on Electronics and Sustainable Communication Systems (pp. 817-825). Springer.
13. Shah, R., & Bhattacharya, S. (2017). Detecting Phishing Websites using Machine
14. Learning Techniques. In Proceedings of the International Conference on Advanced Computing Technologies and Applications (pp. 124-134). Springer.