iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 12 Issue: VII Month of publication: July 2024

DOI: https://doi.org/10.22214/ijraset.2024.63539

Phish-blocker: Safeguarding against Phishing Attacks in Real-Time

Brindha. S[1], Sindhuja. K[2], Eyamini. C[3], Dhivyabharathi. D[4]
[1]Department of Computer Science and Engineering, EASA College of Engineering and Technology, Coimbatore,Tamilnadu,
[2]Department of Computer Science and Engineering, EASA College of Engineering and Technology, Coimbatore, Tamilnadu, India
[3]Department of Artificial Intelligence and Data Science, Kathir College of Engineering, Coimbatore, Tamilnadu, India
[4]Department of Computer Science and Engineering, Dhanalakshmi Srinivasan College of Engineering, Coimbatore, Tamilnadu, India

Abstract: Phishing is defined as a cyber attack that uses social engineering through digital tools to persuade victims to reveal personal information such as their passwords or credit card numbers. Finally, stolen personal information is used to fraudulently trust legitimate websites or financial institutions and generate illegal profits. Although there are different solutions against phishing, phishing attacks have increased significantly in the last few years. Some solutions rely on removing features by rules, while some features rely on third-party services, which can cause instability and prolong problems with prediction services. This project introduced Phish Wiper, a deep learning framework that uses neural networks (RNN) to identify URL and HTML features in a real-time search engine. Phish Wiper uses two deep encryption methods (URL blocks and HTML blocks) that are learned separately and together throughout the connection process, eliminating each network's publishing process as the final decision. This method examines the URL and HTML of web pages and calculates their similarity to known phishing websites to identify them. Phishing detection is a dual task; There are two categories: legitimate and phishing. We use this framework as a browser plug-in that can instantly detect and alert when a user visits a web page if there is a phishing risk. According to the experimental results, the proposed model achieved good performance with an accuracy of 96.79% to 98.90% when measured using different data.

Keywords: Cyber Attack, Phish wiper, Recurrent Neural Network, Binary Classification, Phishing detection.

I. INTRODUCTION

Phishing is a type of cybersecurity attack in which a malicious individual impersonates a trusted person or organization. Phishing messages direct users to perform tasks such as uploading malicious files, clicking on malicious links, or revealing sensitive information such as login credentials. Phishing, the most common form of social networking, is a general term that describes attempts to manipulate or deceive computer users. Social engineering is a more common threat in almost every security situation. Social engineering attacks such as phishing are often combined with other threats such as malware, code injection, and network attacks. Phishing, the most common form of social engineering, is tricking, coercing, or manipulating people into sending messages or valuables to unscrupulous individuals. Social engineering attacks rely on human error and manipulation to succeed.

A. How Phishing Attacks Work

Phishers can use publicly available information to gather background information about the victim's identity, occupation, interests, and activities. Mostly through social networks such as Facebook and Twitter. These sources are often used to reveal information such as victims' names, job titles, and email addresses. This information can be used to create trustworthy emails. Often victims receive messages that appear to be sent by a known person or organization. The attack is then carried out via malicious links or links to malicious websites. In both cases, the goal is to install malware on the user's device or redirect the victim to a fake website. The goal of fake websites is to trick victims into revealing personal and financial information such as passwords, account IDs or credit card information.

II. LITERATURE SURVEY

Ahmed et al. The study found that the difference forest algorithm achieved the highest accuracy at 98.46% accuracy in detecting phishing websites. This study provides insight into the effectiveness of different machine learning algorithms in identifying phishing websites. Hong Shihao et al. The proposed method achieved an accuracy of 99.03% and outperformed many modern phishing website detection methods.

This method uses the deep neural network model, which is a combination of convolutional neural network (CNN) and short-term memory (LSTM) layer. The model was trained on a phishing dataset and legitimate URLs. They often analyze website content, URLs, and other features to determine whether a website is a phishing site.

III. PROBLEM DESCRIPTION

An attacker may intentionally create a phishing website to avoid detection by machine learning models. For example, they may use obfuscation techniques to hide malicious code or manipulate web content to avoid detection. Machine learning models may perform well on data but poorly on new, unseen data. This can pose a problem for detecting phishing sites, as attackers are constantly developing new strategies to avoid detection. It can be difficult to understand how machine learning models make decisions, making it difficult to interpret false positives or identify the nature of the model. In general, using machine learning for phishing URL and HTML website detection can face many challenges. Phishing URL and HTML website detection systems using deep learning can solve these problems, helping organizations protect users' sensitive information and prevent financial losses from phishing attacks. The system must be able to detect the type of phishing attack and adapt to the attackers' changes. It also needs to be able to handle large amounts of data and operate at scale.

IV. EXPERIMENTAL METHOD

Phishing attacks are one of the most common online threats today. To prevent these attacks, we developed a technique called "Phish Wiper", which uses a monitoring technique with recurrent neural networks to detect and block phishing websites in a timely manner. The system consists of several stages, including data collection, preprocessing, image extraction (URL and HTML), classification, design and training, and performance evaluation. This information is collected from various sources and pre-processed to extract relevant features such as domain age, IP reputation and SSL certificate. URL and HTML attributes are extracted using various techniques such as domain analysis, HTML parsing and content analysis. The classification phase involves using the extracted features to train a recursive neural network model to classify websites as legitimate or phishing. Model; It is trained and evaluated using various performance metrics such as accuracy, precision, recall, and F1 score. The system was developed using Python Flask and MySQL. Administrators are responsible for training the model and updating the database with new phishing information. Users can open a browser and enter a URL into PhishWiper and PhishWiper will predict and block the URL if it is identified as a phishing website. The system also stores attack information in user accounts on the PhishWiper website, allowing users to track their activity and take appropriate action. The system has two main modules: training module and detection module. and evaluate the effectiveness of the model. It takes the user's login URL, extracts its features, feeds it to a training model, and returns a prediction that blocks that URL if it is identified as phishing. The system also stores attack data in user accounts on the PhishWiper website for further analysis. The proposed method is intended to provide a more efficient and effective solution than traditional methods for instantly detecting and blocking phishing websites. It also provides a user-friendly interface and an easy way to store and analyze attack data.

V. FLOW DIAGRAM

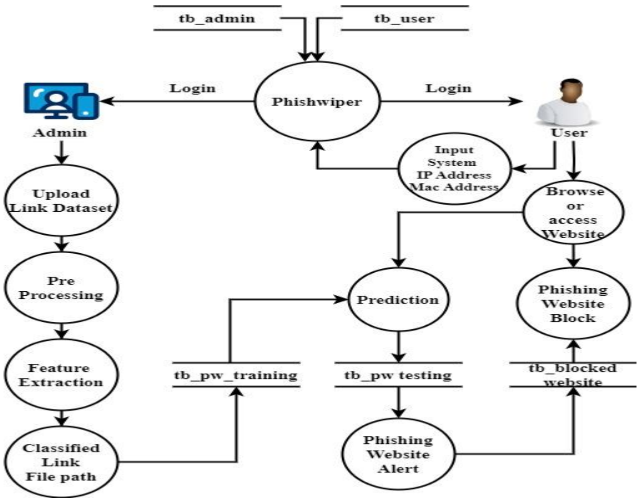


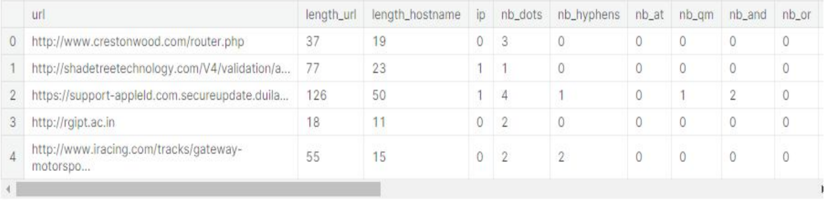
Figure 1: Diagram of Proposed System

VI. IMPLEMENTATION

PhishWiper web application is built using MySQL and Python Flask. The program is designed to provide instant phishing website detection and blocking solutions and URLs. To identify and train phishing websites and URLs, this project uses RNN combined with tracking techniques. The project is divided into several stages: data collection, preprocessing, extraction of features (HTTP and URL), classification of data, creation and training of the RNN, and evaluation of the results. Collect information: Collecting HTML and phishing URLs is the first step in this process. This information is available from many sources, including company security, training materials, and online databases. There must be a sufficient number of models in the dataset for the RNN model to identify phishing features. Since the system uses RNN tracking theory, phishing attack detection and response is accurate and effective. User feedback enters the system regularly, helping to improve classification and make it more resilient to new phishing attacks. The PhishWiper website provides an affordable suite of security software and hardware to detect and block phishing attacks. The aim of the project is to provide a reliable and effective solution for instant identification and blocking of phishing websites and URLs. The effectiveness of the system in preventing cyber security crimes is ensured by its ability to be constantly improved and adapted to new phishing attacks. This research is important because it aims to combat the growing threat posed by phishing attacks.

A. Importing and Searching Data

The "Importing and Searching Data" module in the PhishWiper RNN model is responsible for loading and analyzing the dataset that will be used to illustrate the model. This model first imports data from a CSV file or database using MySQL. It then checks the quality of the data, including completeness, consistency and accuracy, and detects any significant or negative aspects.



	url	length_url	length_hostname	ip	nb_dots	nb_hyphens	nb_at	nb_qm	nb_and	nb_or
0	http://www.crestonwood.com/router.php	37	19	0	3	0	0	0	0	0
1	http://shadetreetechnology.com/V4/validation/a...	77	23	1	1	0	0	0	0	0
2	https://support-appleld.com.secureupdate.duila...	126	50	1	4	1	0	1	2	0
3	http://rgipt.ac.in	18	11	0	2	0	0	0	0	0
4	http://www.iracing.com/tracks/gateway-motorspo...	55	15	0	2	2	0	0	0	0

Figure 2: Diagram of Dataset from a CSV file

Data is usually split into training and testing using an 80/20 split; where 80% of the data is used for training and 20% for testing. The training set is further divided into equal parts and used to train the RNN model. This model also enables Data Analysis (EDA) to gain insight into the data set, such as the distribution of target variables, the relationship between different features among factors, and whether there are patterns or inconsistencies. EDA helps identify defects in the material and select appropriate pretreatment and extraction processes. Overall, the Dataset Import and Search module plays an important role in preparing the data used to train the PhishWiper RNN model and ensure that the model is robust and accurate.

B. Pre-processing

These steps include:

It is converted into tokens at some level, in the form of URL and HTML code. This is done to represent each URL and HTML code as a sequence of numbers that can be fed into the RNN model. The padding step adds zeros to the ends of short segments, ensuring that all strings have the same length. This is done using single-bit encoding or word embedding techniques. Training method is used to train the RNN model, and validation method and testing method are used to evaluate the performance of the training model.

C. Feature extraction

PhishWiper's feature extraction module is responsible for extracting relevant features from pre-processed URLs and HTML pages. This model uses techniques such as Bag of Words (BoW), Time Inverse Document Frequency (TF-IDF), and n-gram to extract important features from documents. This module extracts properties for URLs such as URL length, number of dots and slashes, domain age, and some keywords. For HTML content, remove attributes such as certain HTML tags, JavaScript functions, and embedded objects. The extracted values are then converted into character vectors and used as input to the Recurrent Neural Network (RNN) model for classification. If the value is greater than the threshold, the URL is classified as phishing, if it is lower than the threshold, the URL is classified as legitimate.

D. Figures and Tables

All figures in the manuscript should be numbered sequentially using Arabic numerals (e.g., Figure 1, Figure 2), and each figure should have a descriptive title. The figure number and title should be typed with single-spaced, and centered across the bottom of the figure, in 8-point Times New Roman, as shown below. The figure captions should be editable and be written below the figures.

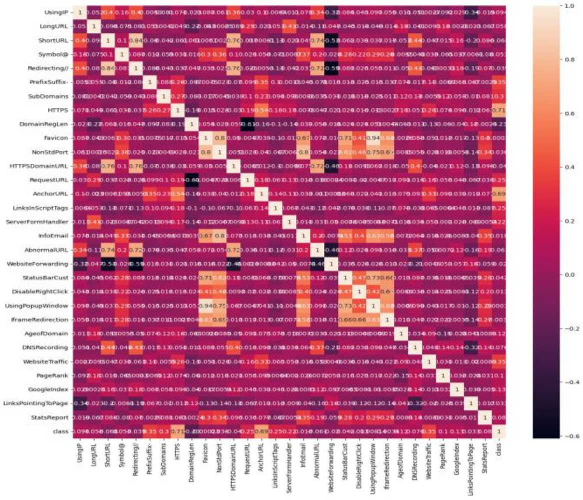


Figure 3: Feature Extraction

This model uses a trained RNN model to preprocess and extract features from the data. The suggested URL is first converted into a feature vector using the feature extraction module described earlier. The feature vector is then fed into an RNN model to predict whether the URL is legitimate or phishing. The RNN model generates a probability value between 0 and 1 for each input URL, indicating the probability that the URL is a phishing website. This value is compared to the administrator threshold.

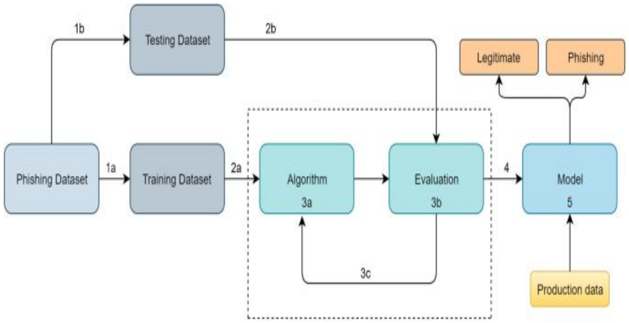


Figure 4: Classification

VII. PERFORMANCE ANALYSIS

Performance analysis of the "PhishWiper" system can be done using the following patterns and models:

Confusion Matrix: Confusion matrix is a table often used to describe the effectiveness of classification models. The confusion matrix is a 2x2 table containing the number of true positives, negatives, negatives, and negatives. The number of URLs in the collection.

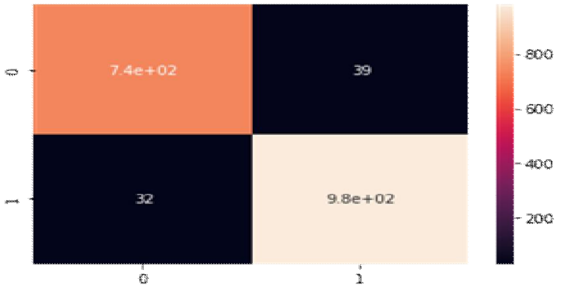


Figure 5 :Confusion Matrix

Accuracy: Accuracy is defined as the number of correctly predicted phishing and real URLs divided by the total number of URLs in the database.

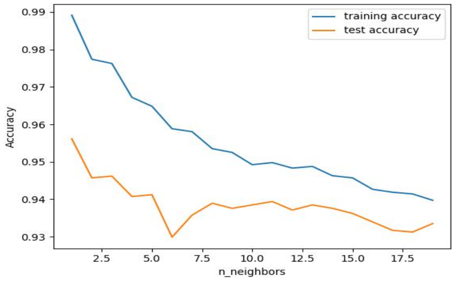


Figure 6 :Performance Analysis

Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

A. *Conclusion*

The PhishWiper system has passed all testing stages and meets all requirements. The system is designed to be distributed and used by end users. Based on the test results, we concluded that PhishWiper is a reliable and effective tool for instantly detecting and blocking phishing websites. The system can protect users from threats by identifying and blocking malicious websites. The evaluation also identified areas for improvement, including optimization to improve performance and improve the accuracy of forecast models. Overall, the testing process gave a good idea of the functionality and functionality of the PhishWiper system, and we believe that this system can provide effective protection against phishing attacks.

B. *Future Improvements*

Here are some potential areas where PhishWiper could be improved in the future: Integration with other web browsers: It is currently designed to work with certain web browsers. Future developments will include expansion to other browsers. Integration with other security tools: This project can be integrated with other security tools such as antivirus software or firewalls to provide a better way to prevent phishing attacks. User feedback and reporting: Allowing users to report potential phishing attacks and provide feedback on the accuracy of predictions can help improve the quality and accuracy of the system's performance.

REFERENCES

[1] Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2016). A machine learning approach for detecting phishing websites. Journal of Intelligent & Fuzzy Systems, 31(5), 2635-2645.
[2] Alazab, M., Watters, P., & Alazab, M. (2017). Machine learning-based phishing detection: An empirical study. Journal of Information Security and Applications, 32, 102-115.
[3] Alhazmi, O., & Almuhaideb, S. (2019). A comparative study of machine learning algorithms for detecting phishing websites. International Journal of Advanced Computer Science and Applications, 10(8), 283-289.
[4] Alturki, M. A., & Xiang, Y. (2017). A deep learning approach to detecting phishing websites. International Journal of Network Security, 19(6), 957-963.
[5] Arora, S., & Singh, S. (2017). An analysis of machine learning algorithms for phishing websites detection. International Journal of Computer Science and Mobile Computing, 6(2), 28-36.
[6] Belkhatir, M., Khoudour, L., & Elboukhari, M. (2018). Machine learning based approach for phishing website detection. International Journal of Computer Applications, 181(30), 28-34.
[7] Bhagavathy, S., & Balamurugan, M. (2018). A hybrid approach of machine learning and fuzzy logic for detecting phishing websites. International Journal of Emerging Trends in Engineering Research, 6(10), 107-114.
[8] Bhattacharyya, S., Kalita, J. K., & Das, S. (2018). Phishing websites detection using machine learning techniques: a review. Journal of Ambient Intelligence and Humanized Computing, 9(3), 625-645.
[9] Chen, X., Wu, Q., Wu, L., & Xie, X. (2017). A machine learning-based approach for detecting phishing websites using website features. International Journal of Communication Systems, 30(6), e3091.
[10] Gandotra, E., & Singh, S. (2018). An analysis of machine learning algorithms for phishing detection. In 2018 3rd International Conference on Computing and Communications Technologies (ICCCT) (pp. 46-50). IEEE.
[11] Hidayanto, A. N., Bayuaji, R., & Kurniawan, F. (2019). Phishing websites detection using machine learning and entropy feature selection. Journal of Physics: Conference Series, 1317(1), 012016.