INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

MaliceSpotter: Revolutionizing Cyber Security with Machine Learning for Phishing Resilience

Shwetambari Borade[1]*, Parshva Chetan Doshi[2], Darsh Bhavesh Patel[2]

1 Assistant Professor, Cyber Security, Shah & Anchor Kutchhi Engineering College, Mumbai, Maharashtra, India
2 Student, Cyber Security, Shah & Anchor Kutchhi Engineering College, Mumbai, Maharashtra, India

OPEN ACCESS

Received: 19-01-2024
Accepted: 30-01-2024
Published: 23-02-2024

Citation: Borade S, Doshi PC, Patel DB (2024) MaliceSpotter: Revolutionizing Cyber Security with Machine Learning for Phishing Resilience. Indian Journal of Science and Technology 17(10): 870-880. https://doi.org/10.17485/IJST/v17i10.148

* **Corresponding author**.

shwetambari.borade@sakec.ac.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To enhance cyber security by implementing advanced algorithms to swiftly identify and neutralize phishing threats. Also, to bolster user protection, fortify data integrity, and ensure a resilient defense against evolving cyber threats. **Methods:** MaliceSpotter aims in classifying user-entered URLs by analysing 28 features, using algorithms like Logistic Regression, Random Forest, and KNN, combined via a Voting Classifier. Dataset on Kaggle provides diverse samples for evaluation. This methodology's unique aspects include multiple algorithm integration and the utilization of Kaggle as a data source. **Findings:** MaliceSpotter demonstrates a commendable accuracy of 95%, effectively classifying input URLs as phishing or legitimate. The system's uniqueness lies in its provision of a detailed report on URL behavior, facilitating informed decision-making. The implementation of ensemble learning is notable, particularly the introduction of the Voting Classifier. This approach leverages various algorithms, successfully incorporating bagging and voting concepts. Through the Voting Classifier, MaliceSpotter gains insights into the working of machine learning algorithms, enhancing the scrutiny of URL behavior. This innovative feature sets MaliceSpotter apart, offering a nuanced perspective on the reliability of URLs through the collective input of diverse algorithms. **Novelty:** MaliceSpotter uniquely combines diverse algorithms, leveraging a voting classifier for robust results. Continuously updating in real time, it meticulously dissects URLs into 28 parts, ensuring thorough scrutiny and effective detection.

Keywords: Phishing; Machine Learning; Web Security; Voting Classifier; Bagging

1 Introduction

The ever-expanding landscape of cyber threats, particularly the pervasive menace of phishing attacks, necessitates innovative solutions for robust detection and prevention. In the research, researchers delve into the realm of phishing detection using machine

learning, with a specific focus on the project named MaliceSpotter. As the digital domain becomes increasingly fraught with deceptive links and malicious schemes, MaliceSpotter emerges as a cutting-edge tool designed to evaluate the credibility of URLs. The research explores the effectiveness of MaliceSpotter's advanced machine learning algorithms in fortifying cybersecurity infrastructure, thwarting potential threats, and contributing to a safer online environment. The existing methodologies which provide efficient results, but the proposed system implements ensemble learning which is one step ahead of machine learning. The main goal of the work is to attain accurate results through this ensemble learning technique called Voting Classifier. This separates the work from the rest as many methodologies discussed Ensemble learning but not implemented it. Another stand out feature is the display of client side as well as server-side output display, a proper explanation in point format is displayed in server-side result for better understanding to the user.[(1)] When seamlessly integrated with an organization's Intrusion Detection System, MaliceSpotter becomes a force multiplier, fortifying the overall cybersecurity infrastructure. The evolution of the Internet has brought about an unfortunate surge in online frauds and scams. MaliceSpotter, with its advanced capabilities, stands as a vigilant guardian, effectively identifying and exposing hoax links. Through real-time alerts, it empowers users to navigate the digital realm with confidence, contributing to a safer online environment and bolstering the resilience of individuals and organizations against evolving cyber threats.

In a comprehensive literature study on machine learning-based phishing site detection, diverse approaches and methodologies were explored across various research articles. The investigation was initiated with reference to[(1)], which proposed a three-step URL classification process, highlighting the resource-intensive nature of downloading multiple online pages in the initial step. Another significant contribution was made by[(2)], which introduced Ensemble Learning to combine results from multiple machine-learning models, resulting in an impressive overall accuracy of 97% in phishing site detection. In[(3)], conducted an exhaustive evaluation of phishing detection technologies, concluding that machine learning emerged as the most effective method for identifying phishing websites. On a contrasting note,[(4)] adopted a unique manual entry approach for phishing URLs on an admin site, leading to network blocking and email alerts upon attempted access. Content-based detection was highlighted in[(5)], focusing on the importance of specific words in URLs, achieving a high accuracy of 99.4% in detecting legitimate sites.[(6)] presented an innovative solution with a dedicated Browser Extension for detecting phishing websites. Support Vector Machines were implemented in both[(7)] and[(8)], with the latter storing identified phishing URLs in a text file.[(9)] introduced a distinctive method by calculating the proportionate distance between input and database URLs, incorporating a Favicon Images Recognition Algorithm alongside established approaches like False Positive and False Negative.[(10)] Adopted a simple yet effective machine learning approach for phishing URL detection.

The various research gaps identified through in-depth research and study in the recent works are as follows:

Manual Inputs - In[(4)], a novel yet ambiguous method involves manually inputting phishing URLs into an admin site for subsequent blocking on the network. However, this approach poses a significant limitation as it fails to detect URLs, not present in the Admin Panel, potentially allowing newer malicious URLs to go undetected. MaliceSpotter is completely automated which checks for all kinds of URLs present in the email as well as in the entire website.

Browser Extension – A significant work of creating a browser extension is done in[(6)], but however the methodology does not involve any feature selection process. The dataset lacks various features which are required to improvise the training of the model. The accuracies achieved by the 3 algorithms are lesser than expected. MaliceSpotter is an enhanced approach of this work which involves a detailed feature selection stage followed by a large number of parameters to check the legitimacy of the input URL.

Support Vector Machine with Ensemble Approach – This existing methodology proposes a single algorithm which attains an accuracy of 98%. Ensemble learning methodology is proposed but it is not implemented. The proposed methodology not only takes 3 algorithms but also makes use of Voting Classifier which is an ensemble learning technique and implements it.

Limited Parameter Scrutiny – In[(5)], many machine algorithms are used but only limited features are scrutinized for results. Bag of words model is used which involves checking the content present inside the phishing website or email. In our proposed methodology, we have made use of 28 parameters which thoroughly checks the input URL right from the time it was created to the time it is used till date.

A detailed review of the research gap has been explained in the Results and Discussion section.

2 Methodology

Protecting data privacy and confidentiality by all means is the first consideration and step in the development of this project. The entire tool was constructed from the ground up in order to better comprehend the capability that is there.[(1)] To initiate the project, a machine learning model was created which helps to distinguish between a malicious or a legitimate URL.

In phishing, the attacker steals sensitive information of a user by tricking him to open a URL which has a high possibility of cracking into the system. The language used makes the user make decisions in haste which results in, him avoiding to check the

legitimacy of the URL. The project involves creation of a machine learning model which uses 3 kinds of algorithms to ensure a concrete classification.[2] The input URL is then passed through these 3 machine learning algorithms namely:

- Logistic Regression
- Random Forest Classifier
- K Nearest Neighbors Classifier

The provided input URL is passed through these algorithms and the results are displayed respectively. The algorithms classify the URL as a legitimate or a phishing website. To ensure a collective decision of all the 3 algorithms, implementation of Voting Classifier which is a part of ensemble learning was done.[3] Ensemble Learning combines and compares the results of all the algorithms. It displays the result in its own manner which takes into account the results of the previously used algorithms.

Figure 1 shows how a dataset[11] which contains URL is used for classifying URL as legitimate or phishing. Feature scaling was then applied on the dataset to get more insights on the integral part of a URL. Feature scaling helped the users to know which parameters will help to get more accurate results to classify a URL.

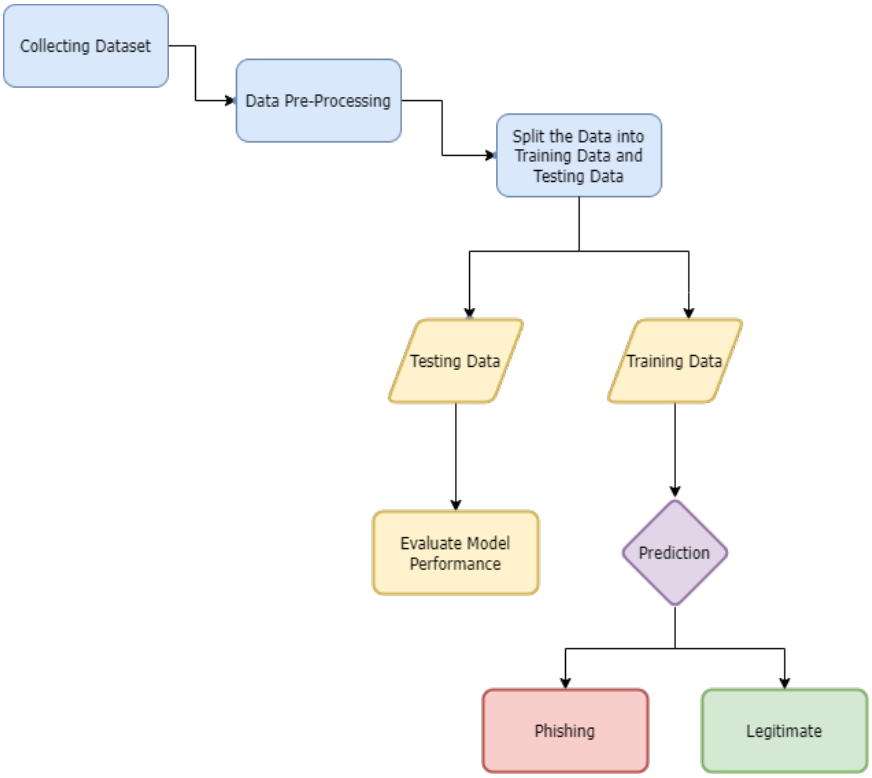

Fig 1. Schematic Representation of MaliceSpotter

Python scripts were developed to build the classification criteria for each parameter. In total scripts for 28 parameters which involve parameters such as Length of the URL, Domain Age, Presence of IP Address in the URL, HTTP/S Presence. It involves thorough checking of a URL which ranges from the most basic domain related checks to checking and analyzing the behavior of a web page.[4] Integration of a database is done which will store[12] all the results and it can be viewed on the website as a separate page. This also enhances the speed of classification of the model. Moreover, for faster classification results a single request for all the 28 parameters is implemented. In other words, a single request will start the calculation of results for all the parameters thus reducing time complexity.

MaliceSpotter aims to simplify the process of inspecting and categorizing a URL. Figure 2 illustrates how MaliceSpotter conducts the classification procedure. A user can input the URL through the frontend interface, as depicted in Figure 6. After the user submits the URL, the backend model, implemented on Flask powered by Python, undertakes processing tasks on the provided URL. The URL undergoes analysis through 28 parameters, and upon completion, an array is generated on the backend, storing the outcomes of the processing tasks. This array is then fed into the machine learning model, which makes predictions about the URL's classification. As seen in Figures 4 and 5, the ultimate conclusion determines whether the URL is authentic or perhaps phishing.

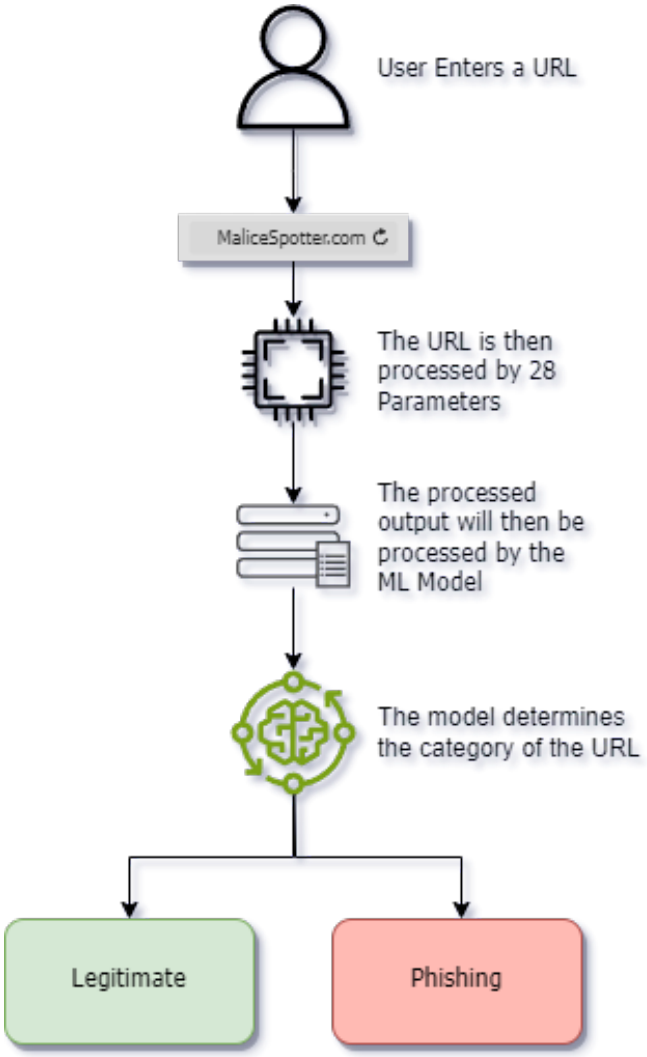

Fig 2. Architecture Diagram of MaliceSpotter

The 28 Parameters used in the Model[11] can be explained in brief as follows:

IP address -[11] Users can be certain if the personal information is trying to be stolen if the input URL consists of IP address instead of domain name.

Length of URL -[11] The input URL can be categorized as phishing if it exceeds 54 characters or more, this is because an input URL is not long as 54 characters.

Short URL -[11] Any kind of redirection which is achieved by shortening of URL, can be termed as phishing as it involves manipulating the domain name.

Redirection -[11] // represents redirection. This should always be present on the sixth position of the URL. If not, then it is known as phishing.

Prefix-Suffix -[11] Attackers generally use (-) symbol to manipulate with links and cause loss to the user.

Sub-domain -[11] The actual sub-domain is mapped to location of a place. Considering Europe then one can assume that .eu is the domain, and the rest of the URL is the sub-domain. The URL can be termed as phishing if there are more dots than one.

HTTPS -[11] Presence of HTTPS determines legitimacy of the URL.

Domain Registration Length -[11] Domain registration happens for at least one year. If a URL contains a domain which is registered only for few days, then it can be termed as phishing. Usually, phishing websites have domain registered for 35 days.

Favicon - [11] The code snippet should match the link of the favicon entered. If a favicon redirects to a different domain, then it can be termed as phishing.

Port - [11] Phishers can manually configure ports and change the status of the ports which will lead to compromising the confidential information about the user.

HTTP - [11] Phishers may spoof a URL by adding the HTTPS token to the domain portion.

Request URL - [11] A universal domain name taken for the URL, should be same as the domain name taken for the media files including APIs.

URL of Anchor - [11] The anchor tags should be same and uniform throughout for the entire functioning of the URL. If it is from any different domain, it can be termed as phishing.

Links In Script - [11]

Server Form Handler (SFH) - [11] Blank responses scan be termed as doubtful as the server needs to make a decision. URLs with blank responses can be termed as phishing.

Mail Link - [11] Various inbuilt functions in languages can be used to check the response to a mail. In JavaScript, check mailto() function.

Abnormal URL - [11] This parameter directly checks the host details with the WHOIS database.

Website Forwarding - [11] The links used to open new pages should be legitimate, if they are getting rendered on a phishing website then the entire URL can be termed as phishing.

On Mouseover - [11] This JavaScript function can be used to check if the status bar can be manipulated, shortened, tampered to change the existence of the URL.

Disabling Right Click - [11]

Using Pop-up Window - [11] Pop up window should never contain text fields, as it is used to display certain content, precisely alerts. If it contains any kind of input field then it can be termed as phishing.

IFrame Redirection - [11] Frameborder element can be used to manipulate the frame of the URL can tamper with the actual frames.

Domain Age - [11] A domain should be at least 6 months old in order to prove that it is of some worthy existence. If it is below 6 months, it can be termed as phishing.

DNS Record - [11] Any URL used always contains some sort of DNS records in the WHOIS database. If there are no records, then it can be termed it as phishing.

PageRank - [11] Phishing webpages have no PageRank. Furthermore, it can be discovered that the PageRank value of the final 5% of phishing websites could be as high as 0.2.

Google Index - [11] A webpage is always indexed by Google. Results displayed on the Google webpage include a response once rendered by Google. Phishing websites usually are not indexed by Google.

The core of MaliceSpotter lies in its sophisticated machine learning model architecture. Leveraging advanced algorithms, the model intricately analyzes diverse features extracted from URLs to discern patterns indicative of malicious or legitimate intent. The ensemble of algorithms, including Random Forests and others, collaborates to enhance the robustness and reliability of the predictive outcomes.

Use of Logistic Regression in MaliceSpotter

A classification model called logistic regression is applied when the dependent variable, or output, has a binary representation, such as 0 for suspicious, 1 for legitimate, or -1 for phishing. Because of this, logistic regression is a suitable approach for our job in determining if a URL, like in the case of MaliceSpotter, is legitimate (1) or a phishing URL (-1). Since our model employs binary classification, the Sigmoid function is used. Sigmoid Function is denoted as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Where f(x) = output of sigmoid function, e = natural logarithm with value (2.72), x = input features provided. In the case of our model, there are multiple parameters which are passed iteratively to the above formula to attain accurate results.

Logistic Regression, with its emphasis on binary classification and the utilization of the Sigmoid function, proves to be a well-fitted algorithm for the task at hand. The iterative application of multiple parameters ensures the accuracy of our model in discerning the legitimacy of URLs, contributing significantly to the efficacy of MaliceSpotter in identifying potential phishing threats.

Use Of Random Forest Algorithm in MaliceSpotter

Training and test data are separated out of the dataset. To obtain training data, the dataset is reduced by 80%. To obtain test data, the trained data is then lowered by 20% once more. Phishing URL detection is only one of the numerous applications

for the popular machine learning technique Random Forest. The method first constructs many decision trees before creating a forest of them. Every decision tree uses a random selection of features and is trained on a subset of the data to assist prevent overfitting.

Entropy is the criterion employed in random forests. It is a metric for disorder or impurity in a collection of data. Entropy is a measure used in Random Forests to assess how well a specific the entire algorithm works on a concept called 'Gini'. This attribute is used to select a feature with the lowest amount of impurity.

It is denoted by the formula:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2 \quad (2)$$

Use of K-Nearest Neighbors in MaliceSpotter

K-Nearest Neighbors (KNN) plays a pivotal role within MaliceSpotter, further enriching the model's capabilities in detecting and classifying phishing URLs. The incorporation of KNN, a versatile and intuitive machine learning algorithm, contributes to the holistic approach employed by MaliceSpotter in combating online threats.

The KNN component of MaliceSpotter relies on feature-based similarity metrics to quantify the resemblance between URLs. Features such as URL structure, domain attributes, and content characteristics are considered, allowing KNN to discern patterns that might indicate malicious intent. This fine-grained analysis enhances the overall accuracy of MaliceSpotter in identifying potential threats.

The majority class among the new data point's K-nearest neighbors determines its class for categorization purposes. It just takes the closest neighbor's class if K=1. It is indicated by:

$$Class\,(x) = arg\ max_c + \sum_{i=1}^{K} I(y_i = c) \quad (3)$$

Where:

Class(x) is the predicted class for data point x, I represent the indicator function, y_i is the class label of the i^{th} nearest neighbor.

Voting Classifier – The Pivotal Element of MaliceSpotter

[13] In the context of MaliceSpotter, a voting classifier combines the predictions from several machine learning models to arrive at a final determination regarding the maliciousness of a particular URL. The system's overall accuracy and resilience can be improved with the help of this ensemble technique. This is a high-level summary of how to configure a MaliceSpotter voting classifier.

MaliceSpotter uses a widely used voting classifier called Hard Voting. The class with the largest majority of votes—that is, the class with the best likelihood of being predicted by each classifier—is the projected output class in a hard voting system. Among the three prediction algorithms, the researchers can presume that two of them will forecast the URL as -1 (phishing), while one program would predict 1 (legal).

3 Results and Discussion

The line graph in Figure 3 provides a comprehensive overview of the accuracy achieved by various machine learning algorithms integrated into MaliceSpotter. Each algorithm's performance is assessed individually, showcasing their predictive capabilities on the given dataset. The achieved accuracy for Logistic Regression, Random Forest, KNN are 92%, 96%, and 94% respectively. On the other hand, MaliceSpotter achieved an Accuracy of 95% using Voting Classifier.

In comparison with [14], it was noted that the results obtained by the existing methodology utilized LGBM (Light gradient boosting algorithm), which could potentially be improved further. The proposed methodology, MaliceSpotter, employs three algorithms that are subsequently combined into a single algorithm to achieve robust results. The model attains an accuracy of 94%, surpassing the accuracy attained by the LGBM classifier (92%). Furthermore, MaliceSpotter utilizes 28 parameters, enhancing the model's ability to conduct a comprehensive examination of a URL, whereas the existing methodology in [14] utilizes only 6 parameters.

The graph prominently illustrates that Random Forest stands out with the highest accuracy among the considered machine learning algorithms. Its robust performance contributes significantly to the subsequent ensemble learning technique, the Voting Classifier.

Interestingly, the Voting Classifier makes use of the advantages of each individual algorithm, and Random Forest, the best performer, has a significant impact on the ensemble's total accuracy. The combined predictive power of MaliceSpotter is enhanced by the synergy of many algorithms.

The line graph serves as a valuable tool for understanding how each algorithm interacts with the dataset. It provides insights into the varying accuracies of different algorithms and their adaptability to the intricacies of phishing URL detection.

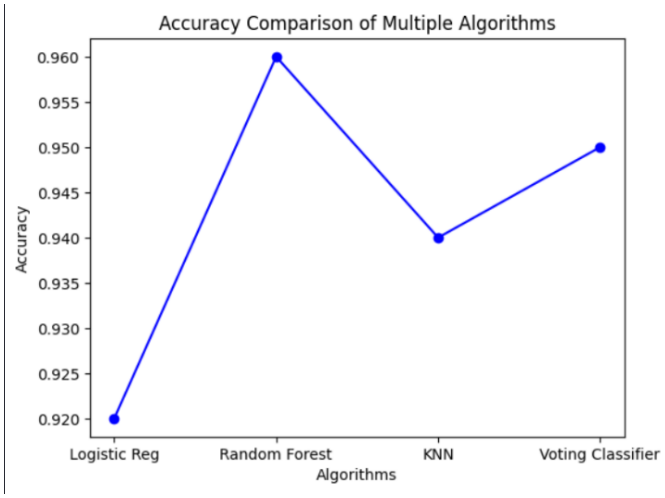

Fig 3. Line Graph of the Accuracy achieved by MaliceSpotter

The results obtained in [4] provide results by manually providing input to the system. MaliceSpotter checks for embedded URLs in images, favicons, admin panel and then concludes whether a URL is benign or legitimate. Redirection is another parameter which has been implemented in our research. All the above-mentioned parameters have been successfully implemented and are providing accurate results.

Limited parameters have been extended by making use of parameters which cover the time frame from the time the URL was created to the existing date of use. Checking the domain age as well as the length of URL enables a user to gain insights about the nature of the URL. Many other parameters are mentioned above have been implemented.

Figures 4, 5 and 6 depict the frontend of MaliceSpotter, supported by Flask and Bootstrap. The images below showcase three different test cases of the MaliceSpotter application. The phishing URL utilized for testing has been sourced from OpenPhish.

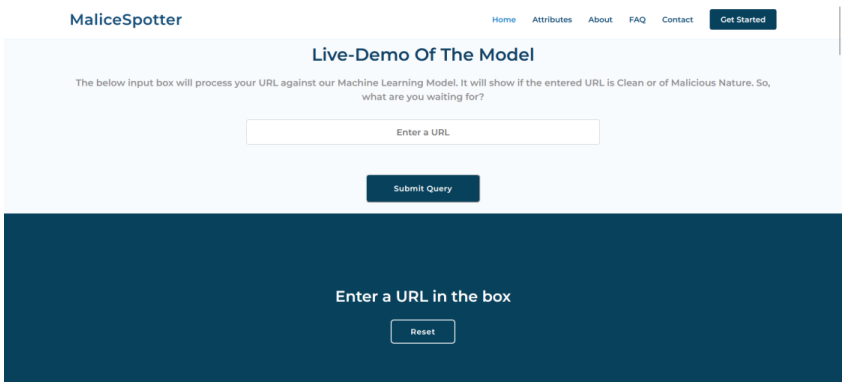

Fig 4. Frontend of MaliceSpotter

In Figure 7, the meticulous process of parameter calculation in MaliceSpotter is elucidated, illustrating the comprehensive approach taken to achieve precise predictions. Each input URL undergoes scrutiny through 28 distinct parameters, and a Legitimate prediction = 1, is determined based on predefined conditions derived from the analysis of these parameters.

Similarly, Figure 8 is obtained by performing analysis and detection of a Phishing URL and it shows an output of -1 which indicates that the model has detected a Phishing URL.

The Figure 9 depicts the Confusion Matrix of MaliceSpotter which shows that MaliceSpotter has achieved 1108 True predictions and 1539 True negative predictions.

TP - True positive refers to the prediction made which are actually true. MaliceSpotter predicted a URL as phishing, and it was actually phishing.

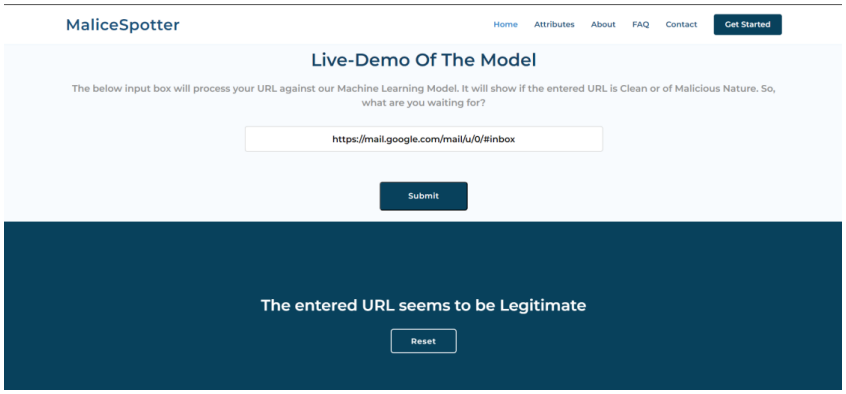

Fig 5. URL classified as Legitimate

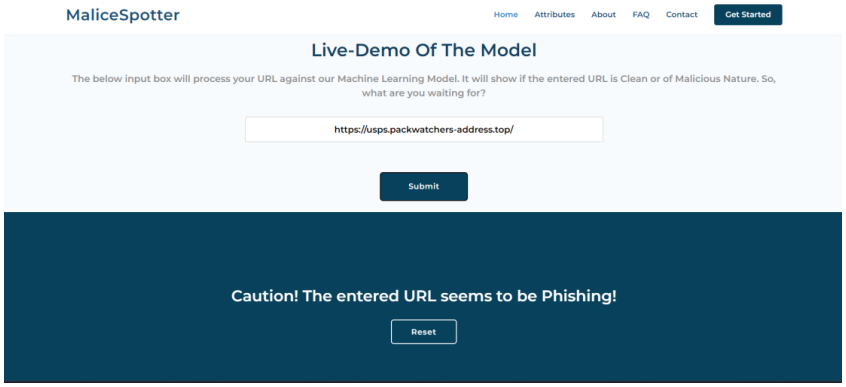

Fig 6. Phishing URL Detected by MaliceSpotter

```
IP: 1
Length:  1
Short URL:  1
@ Symbol 1
Redirect:  1
Prefix Suffix:  1
Subdomains:  1
HTTPS:  1
Domain Reg Len:  1
Favicon:  1
Port:  1
HTTP:  1
RequestURL:  -1
URL Anchor:  0
Links In Script:  -1
Error: 'NoneType' object is not subscriptable
SFH: -1
Mail Link:  1
AbnormalURL 1
WebForwards:  1
Onmouseover: 1
Right Click Disabled: 1
Popup: 1
IFrame: -1
6907
Domain Age:  1
DNSRecording:  1
list index out of range
Page Rank:  -1
Google Index:  1
Links Pointing to URL:  1
[[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, 0, -1, -1, 1, 1, 1, 1, 1, 1, -1, 1, 1, -1, 1, 1]]
[1]
```

Fig 7. Server Response of a Legitimate URL

```
IP: 1
Length:  1
Short URL:  1
@ Symbol 1
Redirect:  1
Prefix Suffix:  1
Subdomains:  1
HTTPS:  -1
Domain Reg Len:  -1
Favicon:  -1
Port:  1
HTTP:  1
RequestURL:  0
URL Anchor:  -1
Links In Script:  1
SFH: 0
Mail Link:  1
AbnormalURL -1
WebForwards:  0
Onmouseover: 1
Right Click Disabled: 1
Popup: 1
IFrame: -1
'NoneType' object is not subscriptable
Domain Age:  -1
DNSRecording:  1
list index out of range
Page Rank:  -1
Google Index:  1
Links Pointing to URL:  -1
[[1, 1, 1, 1, 1, 1, 1, -1, -1, -1, 1, 1, 0, -1, 1, 0, 1, -1, 0, 1, 1, 1, -1, -1, 1, -1, 1, -1]]
[-1]
```

Fig 8. Server Response of a Phishing URL

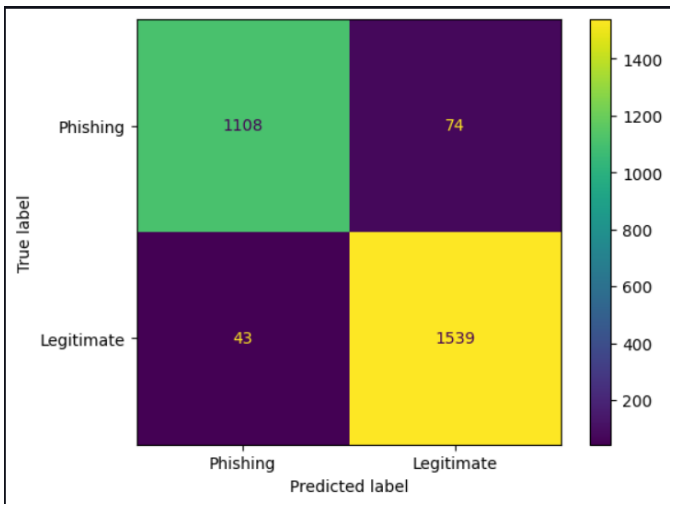

Fig 9. Confusion Matrix of MaliceSpotter

TN - True negative refers to the negative prediction made which is negative. It means you predicted an animal as dog, and it actually is.

FP - False positives are positive predictions which are false. You predicted a URL as phishing, but it is legitimate. (Type 1 Error) FN - False Negative are negative predictions which are negative. You predicted that URL is not phishing, but it is phishing. (Type 2 Error).

In comparison, [7] proposes a single algorithm which attains an accuracy of 98%. Ensemble learning methodology is proposed but it is not implemented. The proposed methodology not only takes 3 algorithms but also makes use of Voting Classifier which is an ensemble learning technique and implements it. The results visible in Figure 10 [7], portrays 842 as true positives and 1190 as false negative. The values obtained by the research clearly indicate that the dataset used by our research is large. Presence of more values indicates that the model is well trained in comparison. The accuracy obtained by the algorithms in [7] are less compared to the accuracy obtained by MaliceSpotter. In addition to the higher accuracy, the research also implements an ensemble learning technique called Voting classifier. An SVM ensemble machine learning technique is proposed in [7] whereas Voting classifier attains accuracy which is higher as compared to the normal algorithms used.

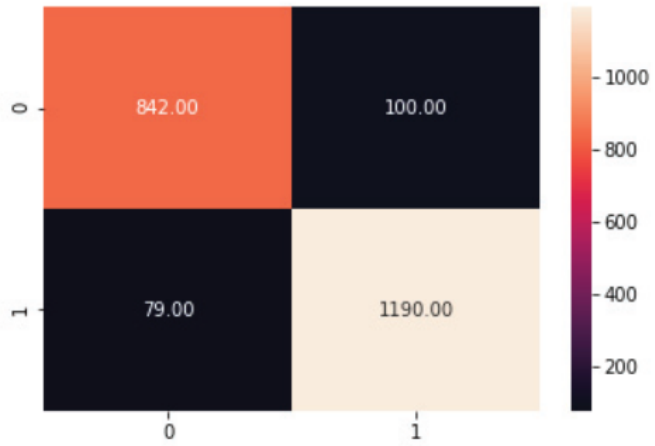

Fig 10. Confusion Matrix of existing research

4 Conclusion

In conclusion, MaliceSpotter represents a significant stride in the realm of cybersecurity, offering an effective phishing URL detection system. Leveraging a Flask-based frontend and employing machine learning models such as Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), and a Voting Classifier, with which we achieved an accuracy of 92.69%, 94.74%, 94.53%, 95.76% respectively. MaliceSpotter empowers users to promptly discern the legitimacy of URLs, safeguarding against malicious online threats. The project has already proven its worth in enhancing online security and aiding users in making informed decisions while navigating the web.

Looking ahead, the future holds promising opportunities for MaliceSpotter. We envision extending our reach by developing browser extensions and mobile applications, providing users with on-the-go protection. Additionally, we plan to continually enhance the project by incorporating new features like analyzing the Email Headers to adapt to the evolving phishing techniques and digital threats. Through this ongoing commitment to innovation, MaliceSpotter will remain at the forefront of the battle against online malice, ensuring a safer and more secure digital environment for all.

References

1) Rashid J, Mahmood T, Nisar MW, Nazir T. Phishing Detection Using Machine Learning Technique. In: 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH). IEEE. 2020;p. 43–46. Available from: https://doi.org/10.1109/SMART-TECH49988.2020.00026.
2) Lakshmanarao A, Rao PSP, Krishna MMB. Phishing website detection using novel machine learning fusion approach. In: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). IEEE. 2021;p. 1164–1169. Available from: https://doi.org/10.1109/ICAIS50930.2021.9395810.
3) Alkawaz MH, Steven SJ, Hajamydeen AI, Ramli R. A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods. In: 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). IEEE. 2021;p. 82–87. Available from: https://doi.org/10.1109/ISCAIE51753.2021.9431794.
4) Alkawaz MH, Steven SJ, Hajamydeen AI. Detecting Phishing Website Using Machine Learning. In: 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA). IEEE. 2020;p. 111–114. Available from: https://doi.org/10.1109/CSPA48992.2020.9068728.
5) Paliath S, Qbeitah MA, Aldwairi M. PhishOut: Effective Phishing Detection Using Selected Features. In: 2020 27th International Conference on Telecommunications (ICT). IEEE. 2020;p. 1–5. Available from: https://doi.org/10.1109/ICT49546.2020.9239589.
6) Mathankar S, Sharma SR, Wankhede T, Sahu M, Thakur S. Phishing Website Detection using Machine Learning Techniques. In: 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP). IEEE. 2023;p. 1–6. Available from: https://doi.org/10.1109/ICETET-SIP58143.2023.10151640.
7) Jain S, Gupta C. A Support Vector Machine Learning Technique for Detection of Phishing Websites. In: 2023 6th International Conference on Information Systems and Computer Networks (ISCON). IEEE. 2023;p. 1–6. Available from: https://doi.org/10.1109/ISCON57294.2023.10111968.
8) Helmi RAA, Johar MGM, Hafiz MASBM. Online Phishing Detection Using Machine Learning. In: 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC). IEEE. 2023. Available from: https://doi.org/10.1109/ICAISC56366.2023.10085377.
9) Mohammed M, Prasanth KK, Subhash SVS. Phishing Detection Using Machine Learning Algorithms. In: 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE. 2022;p. 921–924. Available from: https://doi.org/10.1109/ICSSIT53264.2022.9716269.
10) Shoaib M, Umar MS. URL based Phishing Detection using Machine Learning. In: 2023 6th International Conference on Information Systems and Computer Networks (ISCON). IEEE. 2023;p. 1–7. Available from: https://doi.org/10.1109/ISCON57294.2023.10112184.
11) Mohammad RM, Thabtah F, Mccluskey L. An assessment of features related to phishing websites using an automated technique. In: 2012 International Conference for Internet Technology and Secured Transactions. IEEE. 2012;p. 492–497. Available from: https://ieeexplore.ieee.org/document/6470857.

12) Tanimu J, Shiaeles S. Phishing Detection Using Machine Learning Algorithm. In: 2022 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE. 2022;p. 317–322. Available from: https://doi.org/10.1109/CSR54599.2022.9850316.
13) Puri N, Saggar P, Kaur A, Garg P. Application of ensemble Machine Learning models for phishing detection on web networks. In: 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT). IEEE. 2022;p. 296–303. Available from: https://doi.org/10.1109/CCiCT56684.2022.00062.
14) Thahira A, John A. Phishing Website Detection Using LGBM Classifier With URL-Based Lexical Features. In: 2022 IEEE Silchar Subsection Conference (SILCON). IEEE. 2023;p. 1–7. Available from: https://doi.org/10.1109/SILCON55242.2022.10028793.