International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-12 Issue-1, May 2023

Real-Time Phishing Website Detection using Machine Learning and Updating Phishing Probability with User Feedback

Mitesh M. Adake, Atharva M. Belekar, Chinmay U. Ambekar, Dipika D. Bhaiyya

Abstract: ***Phishing attacks continue to pose a significant threat to internet users worldwide. Cybercriminals often send phishing links through various channels, such as emails, social media platforms, or text messages, to trick users into disclosing their sensitive information, including passwords, usernames, or credit card details. This stolen information is then used to perpetrate various types of fraud or sold on the dark web for profit. To combat this problem, various machine learning-based solutions have been developed for detecting phishing websites. However, these solutions vary in their effectiveness, with some focusing on URL-based algorithms while others concentrate on website content. This paper proposes a machine learning-based approach to real-time phishing website detection, focusing on the website's URL, domain, page content, and overall content. The proposed framework will be implemented as a browser plug-in, which can identify phishing risks as users visit websites. The framework integrates several techniques, including blacklist interception, whitelist filtering, and machine learning prediction, to improve accuracy, reduce false alarm rates, and minimise computation times. The proposed approach also incorporates user feedback to update the phishing probability over time, thereby increasing the accuracy of detecting phishing websites. This feedback loop involves users reporting suspected phishing websites to the system, which then updates the phishing probability calculation with new information to improve its accuracy. The significance of this research lies in its ability to provide real-time phishing detection capabilities, which can help protect internet users from falling victim to phishing attacks. Furthermore, the use of machine learning-based algorithms and user feedback ensures that the system is continuously updated to remain effective against new and emerging threats.***

Index Terms: ***URL, Phishing, Machine Learning, Cyber Security, Web Browser Extension***

Manuscript received on 20 April 2023 | Revised Manuscript received on 28 April 2023 | Manuscript Accepted on 15 May 2023 | Manuscript published on 30 May 2023.

*Correspondence Author(s)

Mitesh M. Adake*, Department of Computer Engineering, Pune Institute of Computer Technology, Pune (Maharashtra), India. E-mail: miteshadake29@gmail.com, ORCID ID: https://orcid.org/0009-0000-0408-2195

Atharva M. Belekar, Department of Computer Engineering, Pune Institute of Computer Technology, Pune (Maharashtra), India. E-mail: athbelekar32@gmail.com, ORCID ID: https://orcid.org/0009-0000-7807-5297

Chinmay U. Ambekar, Department of Computer Engineering, Pune Institute of Computer Technology, Pune (Maharashtra), India. E-mail: cuambekar@gmail.com, ORCID ID: https://orcid.org/0009-0008-1966-1278

Prof. Dipika D. Bhaiyya, Department of Computer Engineering, Pune Institute of Computer Technology, Pune (Maharashtra), India. E-mail: ddbhaiyya@pict.edu, ORCID ID: https://orcid.org/0000-0002-9906-9459

I. INTRODUCTION

Phishing has become a significant concern for security researchers because creating fake websites that resemble legitimate ones is not difficult. While experts can identify these fake websites, not all users have the same ability. The primary objective of these attacks is to steal bank account information. The success of phishing attacks is mainly due to a lack of user awareness. Since phishing exploits user weaknesses, mitigating these attacks is challenging, but improving phishing detection techniques is essential.

Phishing is a type of deception where a malicious website masquerades as a legitimate one to steal sensitive data, such as login credentials, account details, or credit card information. While there are anti-phishing programs and techniques for identifying potential phishing attempts in messages and detecting phishing content on websites, phishers constantly devise new and hybrid methods to bypass these programs and systems. Phishing utilises fake messages designed to appear legitimate and seem to originate from trustworthy sources, such as financial institutions or online businesses. Links in these messages redirect users to fraudulent websites. It is a combination of technology and social engineering. One method for detecting phishing websites is to analyse the URLs. This is a traditional method for identifying a phishing website. A warning pop-up in the browser to alert the user is an effective technique for preventing users from visiting phishing websites and securing their sensitive information.

With the growth of machine learning techniques and their applications in various fields, including cybersecurity, machine learning can be used to detect phishing websites. To address the issue, several experts have proposed utilising deep learning and machine learning techniques. In this paper, we propose a tool that provides real-time detection of phishing websites in the form of a browser plugin. Despite the promising results shown by previous studies, a mechanism for identifying fake websites has not yet been widely used by the sector. This is because there isn't a thorough, accepted strategy for identifying these websites, as well as the challenges associated with real-time detection using machine learning algorithms. The URL space is also highly unbalanced, with a significantly larger number of benign URLs than phishing URLs, and is constantly changing, requiring regular updates to the classifier. Additionally, the growth of the URL space is unlimited, making it impossible to train on every URL using conventional batch learning techniques.

This research presents a fully automated approach for identifying phishing websites, aiming to address these issues. Although the framework is designed explicitly for phishing website detection, the overall strategy can be applied to identify all types of malicious websites.

II. LITERATURE REVIEW

Many previous researchers have used machine learning models and neural networks to detect phishing and malicious websites. The performance of the models depends on the dataset used, the feature set, and the algorithm used.

Tang et al. [1] implemented various types of RNN such as RNN-GRU and RNN-Long-short-term memory (LSTM). They compared Logistic Regression, Support Vector Machines (SVM), Random Forest, and Neural Network. The dataset used had a variety of URLs. However, the Neural Network lacks the feature of including page content, domain, and IP verification, and performs poorly with short URLs.

Alswailem et al. [2] compared the results of the following models: Decision Tree, Artificial Neural Network, Naive Bayes, Support Vector Machines, and KNN. The dataset consists of 36 features. However, obtaining these 36 features from a URL in real-time is a challenging task.

Xiang et al. [3] presented a new approach in identifying malicious URLs by incorporating two novel features: the HTML content of the website and the PageRank of the URL. Their work demonstrated that PageRank could be a valuable tool in detecting malicious websites, as they typically have a low PageRank. The main contribution of their study was the rapid detection of duplicate malicious websites through the use of hashes created from previously visited websites. In theory, this enabled the detection of reused malicious content across different URLs. However, the limitations of their study included a small dataset and a particular heuristic algorithm for detecting duplicates.

Somesha et al. [4] presented a study on using deep learning models for phishing website detection based on ten features obtained from HTML and a third-party service. They calculated the relevance of 18 features and compared the effectiveness of three deep learning models. The Long Short-Term Memory (LSTM) model achieved the best accuracy, with a score of 99.57%. However, the study only utilised one published dataset with 3526 instances, which is insufficient for deep learning training. The high accuracy rate in the results might be due to the imbalance and lack of diversity in the test data.

Zhao et al. [5] employed the use of features of a URL that change over time to determine if it transforms into a malicious website. They devised a suitable training metric to address the dataset imbalance, where benign URLs outnumber malicious URLs significantly. Their work was revolutionary, as they were the first to employ a selective sample online learning algorithm, which ensured the rapid identification of phishing websites. Although the researchers made a substantial contribution with their deliberate selection of dangerous URLs, they primarily utilised an existing dataset and made only minor adjustments to the URL identification system. Purwanto et al. [6] proposed a feature-free method for detecting phishing websites using Normalized Compression Distance (NCD), which compresses two websites and calculates the similarity between them. This eliminates the need to perform feature extraction. This method examines the HTML of web pages and calculates their similarity to known phishing sites. They perform phishing prototype extractions using the Furthest Point First method, choosing samples that are typical of a collection of phishing web pages. This approach has a low false positive rate (FPR) of 0.58%, a high actual positive rate (TPR) of nearly 90%, and an AUC score of 98.68%.

Jeeva et al. [7] relied on lexical features derived solely from the URL to distinguish between benign and malicious URLs. Their approach was based on statistical analysis, and they identified 14 statistics of the dataset to differentiate between malicious URLs manually. Although this method was speedy, it was vulnerable to human error and lacked the flexibility to adapt to the internet's dynamic nature. In the long run, it is expected that an ensemble technique like random forest will provide better performance than its constituent algorithms.

Maurya et al. [8] presented an anti-phishing system that includes a web browser extension. This browser plugin, in real-time, captures the current URL and extracts features based on the Document Object Model (DOM) structure. Then, the system determines if there is a risk of a phishing attack and alerts the user. There are three steps in the detecting procedure: whitelist matching, blacklist filtering, and prediction using a machine learning model. However, these criteria are susceptible to manipulation by attackers and may lead to the misidentification of legitimate URLs. To enhance accuracy, the authors employed an ensemble technique by combining three fundamental classification models.

III. PROPOSED WORK

Figure 1 shows the Machine Learning Classifier. The architecture is explained in the following subsections.

A. Overview of Data Flow

Initially, the user has access to the browser plugin (web browser extension). Users will enter the website's URL into the browser's address bar. Let us assume the user is interacting with the website using the Chrome browser. The Chrome plugin has functionalities such as version number compression, integration of built-in APIs, and permission control. Communication between the components is achieved through messaging and stored temporarily in Chrome storage. The plugin operates by capturing the URL of the current webpage the user is visiting. A background script is implemented to monitor changes in the browser tabs, retrieve the currently accessed URL, call the prediction service for analysis, and relay the result to a content script that presents the outcome on the page. The user is also presented with a pop-up HTML page, which displays detailed information, including the level of risk and other relevant details. Upon accessing the page with the web browser, the user can activate the plugin button on the right side of the toolbar to bring up the pop-up page. The background script is written in Python.

The script is capable of making an API call, allowing the plugin to receive the prediction output from the Machine Learning classifier. If the entered URL is identified as a potential phishing threat, a red background pop-up box appears, warning the user of the risk. In case of a false alarm, the user can acknowledge it by clicking the false alarm button.

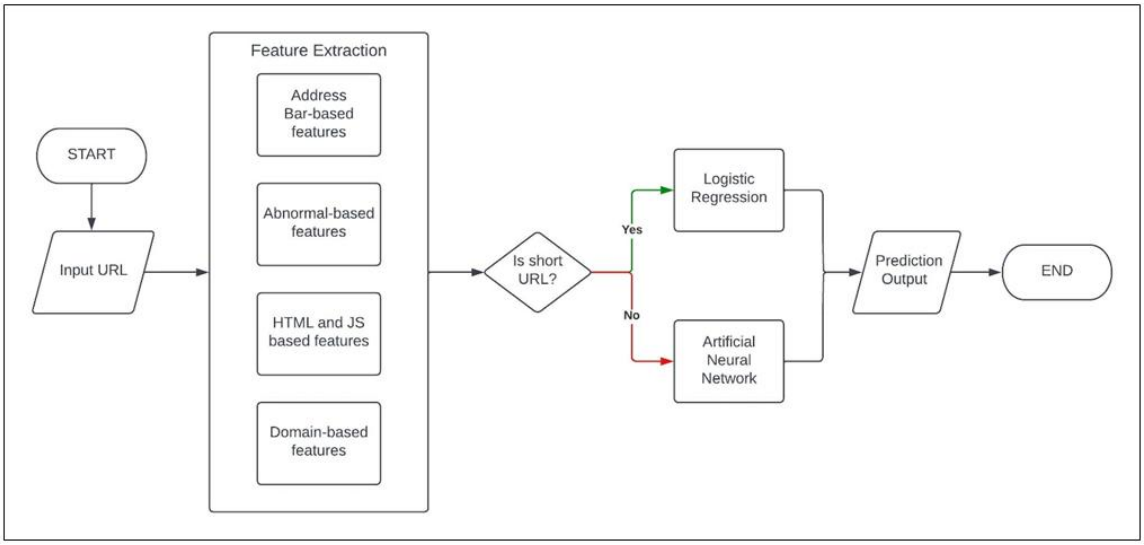

Fig. 1. Machine Learning Classifier

B. Feature Sets

The dataset we are using includes various features that are relevant when evaluating the legitimacy of a website URL. The following are the key elements that play a role in detecting and categorising phishing websites:

1. *Address Bar-Based Features:*

a) Using the IP Address:
It is essential to be cautious when encountering a URL that uses an IP address instead of a domain name. An IP address-based URL, such as "125.98.3.123", may indicate that the website is not legitimate, and the user's information may be at risk. To protect against these types of threats, it's recommended only to enter personal information on websites with domain names that have been verified as safe.

b) URLs having the '@' symbol:
When a URL includes an '@' symbol, the browser disregards everything before it and instead focuses on the content that follows the symbol. This can be a red flag, as it allows malicious actors to hide the true destination of the URL and trick users into visiting a fake or phishing website. It is essential to exercise caution and verify the authenticity of a website before entering any personal information.

c) Length of URL:
The length of a URL can be used as a factor in detecting a phishing website. Long URLs can sometimes be used to conceal the suspicious part of the address in the address bar, making it difficult for users to determine whether the website is legitimate or not. A shorter URL, on the other hand, may be easier to recognise and more trustworthy. However, it is essential to note that the length of a URL alone is not a foolproof indicator of a phishing website, as phishing attacks can also be carried out using short URLs. Therefore, multiple features and indicators should be considered together to determine the legitimacy of a website accurately.

d) Depth of URL:
The depth of a URL refers to the number of subdirectories or nested folders within the main URL. It is used as a feature in detecting phishing websites, as it can indicate the presence of a suspicious or malicious website. For instance, a lengthy and deeply nested URL structure could suggest that the website is attempting to conceal its true identity or purpose. On the other hand, a URL with a shallow depth, such as a simple domain name with no subdirectories, is often associated with legitimate websites. In this manner, the depth of a URL can be used as a feature in the detection of phishing websites. However, it should be noted that the depth of a URL alone may not be a foolproof method for detecting phishing websites, and other features should also be considered.

e) Redirecting using '//':
When a URL contains the '//' sequence, it may indicate a redirection to another website. The existence of '//' within the URL path can be a potential indicator of a phishing website, as users are redirected to a different website than what they intended to access. However, it is essential to note that not all instances of '//' within a URL necessarily imply a phishing attempt, as this can also occur in legitimate URLs for various reasons. To detect phishing websites, it is essential to consider other indicators along with the presence of '//' within the URL.

f) http or https:
The presence of HTTPS in a website's URL indicates that the connection between the user's device and the website is encrypted, and therefore more secure.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

However, simply having HTTPS is not enough to guarantee the legitimacy of a website. There are instances where phishing websites have also implemented HTTPS. Hence, users need to take additional measures, such as verifying the website's identity and checking for red flags, to determine the credibility of a website.

g) Using URL Shortening Services:
URL shortening services, such as TinyURL, enable the creation of shorter versions of URLs. This can be useful for sharing links on platforms with limited space for characters, but it can also potentially hide the true destination of a link. For example, a phishing website may use a URL shortening service to conceal its malicious intent. As a result, it is essential to exercise caution when clicking on shortened URLs and always verify the true destination before entering any sensitive information.

h) Prefix or Suffix separated by (-) to the domain:
When it comes to URLs, it is essential to examine the structure of the domain name. Some malicious websites might add prefixes or suffixes separated by a dash (-) to the domain name to trick the user into thinking they are dealing with a legitimate website. This is a technique used by phishers, and it is essential to be aware of it. However, it is worth noting that the dash symbol is not commonly used in legitimate URLs. So, it can be used as a feature to detect phishing websites.

2. *Abnormal Based Features:*

a) Abnormal URL:
The presence of an unusual URL structure may indicate a phishing website. This feature can be determined by analysing the WHOIS database for the website in question. A legitimate website typically includes information about its identity as part of its URL structure.

b) Server Form Handler:
The Server From Handler (SFH) is a feature used to detect suspicious websites. If an SFH contains an empty string or “about: blank”, it is considered dubious because it suggests that something will be done with the submitted information.

c) Submitting Information to Email:
The submission of personal information through web forms can pose a risk if the information is redirected to a phisher's email instead of a legitimate server for processing. To avoid this, it is essential to verify the authenticity and security of the website before entering any sensitive information. This can be done by looking for indicators such as the presence of a secure connection (HTTPS), a privacy policy, and a legitimate and trustworthy domain name.

3. *Domain-Based Features:*

a) DNS Record:
DNS (Domain Name System) records contain information about a domain name, such as its IP address, mail servers, and domain registrar. When checking for phishing websites, the DNS record can be used to verify the legitimacy of the website. If the DNS record is not found, the website may be a phishing site, as the WHOIS database may not recognise the claimed identity. However, this method is not foolproof and may not always accurately determine if a website is a phishing site, as some phishing sites may have valid DNS records. Therefore, it is essential to employ a combination of different methods to assess the legitimacy of a website, such as verifying the SSL certificate, checking the URL, and looking for red flags, including requests for sensitive information or unusual URLs.

b) Website Traffic:
The number of visitors and pages visited on a website does not directly address plagiarism. However, it could potentially be an indicator of the credibility or reliability of the information found on a website. It should therefore be considered alongside other factors when evaluating the credibility of a source.

c) Age of Domain:
If a website has a relatively short lifespan, it may be less trustworthy and more likely to contain plagiarised content. This information can be obtained from the WHOIS database, and it is recommended to check if the domain has been registered for at least 6 months, which is considered the minimum age for a legitimate domain. This is just one of many factors to consider when evaluating a website's reliability and credibility. Still, it is essential to use multiple sources of information and exercise caution when relying on online content.

4. *HTML and JavaScript-Based Features:*

a) IFrame Redirection:
IFrame redirection is a technique used by phishing websites to trick users into visiting a malicious website by embedding it within another legitimate-looking website. It is essential to exercise caution when clicking on links within IFrames and to verify the authenticity of the website before entering any sensitive information. Additionally, keeping software and browsers up to date with the latest security patches can help prevent IFrame redirection attacks.

b) Status Bar Customisation:
The “mouse over” effect refers to what happens when the mouse cursor is moved over a hyperlink or other clickable element on a webpage. To check the impact of the mouse over on the status bar, the user should place the mouse cursor over the link and observe the URL displayed in the status bar at the bottom of the browser window. If the URL displayed in the status bar differs from the one shown in the hyperlink, it may indicate that the link is a phishing attempt, and the user should exercise caution before clicking on it.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

c) Disabling Right Click:
Disabling right-click is a technique used by phishers to prevent users from viewing and saving the source code of a webpage.
By disabling the right-click function, the phishers aim to hide the underlying HTML code that could reveal their intentions. To avoid falling for such phishing attempts, it is recommended to use browser extensions or keyboard shortcuts that allow users to view the source code of a webpage. Additionally, users can also use private browsing mode, which typically disables JavaScript, to view the source code of a webpage.

d) Website Forwarding:
Website forwarding refers to the process of directing a domain name or website to another destination. In the context of detecting phishing websites, the number of times a website has been redirected can be used as a feature to distinguish phishing websites from legitimate ones. Suppose a website has been redirected multiple times. In that case, it may be a sign of a phishing website as phishers often use multiple redirects to hide their true identity and deceive users into providing personal information. Additionally, customisation of the status bar, which displays information about links or pages, can also be used as a feature to detect phishing websites, as phishers may use this to hide the true destination of a link.

C. Machine Learning Classifier

Data Collection: The dataset used to train the model is taken from PhishTank, JPCERTCC, domcop.com and Kaggle. Twenty-four thousand six hundred thirty-four phishing URLs are obtained from Phish Tank. 108,401 phishing URLs from JPCERTCC. A total of 651,191 URLs, out of which 428,103 URLs are legitimate websites and 223,088 URLs are phishing sites, were obtained from the Kaggle Dataset. 164,146 URLs are legitimate websites from domcop.com. The above dataset is combined to train a logistic regression model. A random sample of 2,000 websites is chosen from the combined dataset above, ensuring an equal number of phishing and legitimate websites. This dataset is used to train a neural network.

Data Preprocessing: Preprocessing is a step in machine learning that involves removing null values, transforming the data, cleaning it, and reducing it to fit the model better. This paper used the Python pandas library to clean unnecessary data and transform text data into numerical data. After identifying the missing attributes in our dataset, they were replaced with values derived from the existing ones. Also, duplicate URLs are removed from the dataset.

To build the model, we define a machine learning pipeline that consists of two steps. The first step is the Count Vectorizer, which is a pre-processing step for text data. The purpose of this step is to convert the text data into a numerical representation that the machine learning model can use.

The Count Vectorizer takes two parameters: the first is the tokeniser, which is used to break the text data into individual words. In this case, the Regexp Tokeniser is used, a tokeniser based on regular expressions. The regular expression [A-Za-z]+ is used to tokenise only the words that contain alphabetical characters. The second parameter is the stop words, which are commonly used words in the text data that do not provide significant information to the machine learning model. In this case, the stop words are set to 'English'.

Machine Learning Algorithm: The second step in the pipeline is the Logistic Regression model, a classification model used to predict the class of an observation based on the input features. This is the model that will be trained on the numerical representation of the text data, generated by the Count Vectorizer, to classify the text data into different categories. The following equations are an illustration of the logistic regression model:

$$x = c_o + \sum_{i=1}^{n} c_i x_i \quad (1)$$

$$P(x) = \frac{e^x}{1+e^x} \quad (2)$$

In summary, a machine learning pipeline that takes text data as input, converts it into a numerical representation using the Count Vectorizer, and then uses the Logistic Regression model to classify the data into two categories. Following is the classification report:

```
Training Accuracy : 0.9568861104215095
Testing Accuracy : 0.9467334097659177

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.98      0.94      0.96    159123
        Good       0.87      0.96      0.91     62423

    accuracy                           0.95    221546
   macro avg       0.93      0.95      0.94    221546
weighted avg       0.95      0.95      0.95    221546
```

Fig. 2. Classification Report

Furthermore, another model is built that not only takes in URL features but also the page content as features. Once the feature extraction is done, there are redundant data points with the same feature values that need to be dropped from the dataset.

Implementation of a phishing website classification model using the Keras deep learning library. The model consists of a sequential architecture, with four dense layers, and the last layer features a sigmoid activation function. This is applied to input data of shape 16, which is a binary classification problem.

The first dense layer has 64 neurons, the second and third dense layer has 32 and 24 neurons, respectively. The activation function used in all layers is the Rectified Linear Unit (ReLU), except for the final layer, which uses a sigmoid activation function to output a binary value of either 0 or 1, representing the prediction of whether a website is phishing or not.

The model is then compiled with a binary cross-entropy loss function, using the Adam optimiser, and accuracy as the metric. The model is trained on the X_train and y_train data for 150 epochs, with a batch size of 32, and 20% of the data is used for validation.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The testing accuracy of the neural network is 86%.

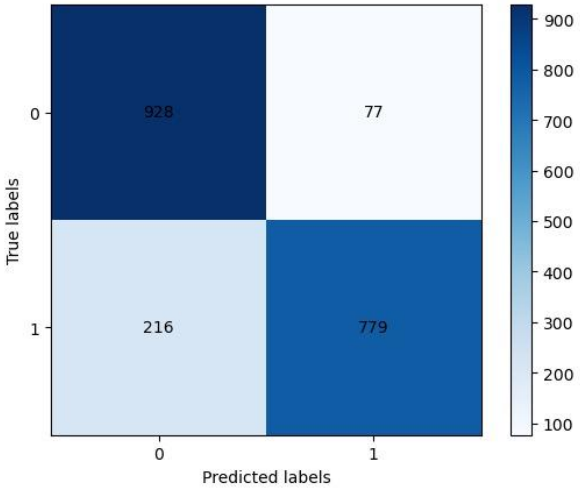

Fig. 3. Confusion Metrics

Suppose the input URL is short, containing only the domain name (50-54 characters). In that case, the logistic regression model performs better than the neural network because the neural network requires various features to be fed in, which cannot be extracted from short URLs.

D. Updating Phishing Probability with User Feedback

The phishing probability prediction produced by the Machine Learning Classifier model for a URL may not always be correct. Thus, the user's feedback is crucial in updating the phishing probability for the URL. The input can be based on the user's sentiment, which reflects their subjective feelings and emotions towards the website. For example, a user may feel suspicious or uneasy about a website's design, content, or behaviour, leading them to believe that the website is phishing. Alternatively, a user may feel confident and comfortable with a website, leading them to think that the website is not phishing.

The feedback can be valuable for all other users. However, it is essential to acknowledge that the feedback mechanism may not be entirely reliable, as users' sentiments can be subjective and influenced by factors such as prior experiences, biases, and emotions. Therefore, we should use feedback from multiple users to verify the authenticity of a website and determine if it is truly phishing or not.

Algorithm 1 Pseudocode to update phishing probability based on user feedback for a website

Require: feedback: user feedback value ranging from [0-10]
Ensure: Updated phishing probability

$threshold \leftarrow variable$ ▷ to trigger update
$factor \leftarrow 0$ ▷ probability deviation
$feedbacks \leftarrow 0$ ▷ number of registered feedbacks
while new feedback registered **do**
  $feedbacks \leftarrow feedbacks + 1$
  $factor \leftarrow factor + 0.02 * feedback - 0.1$
end while
if $feedbacks \geq threshold$ **then**
  $new\_probability \leftarrow old\_probability + \frac{factor}{feedbacks} * 100$
  $new\_probability \leftarrow max(min(new\_probability, 100), 0)$
end if

To trigger an update for a website's phishing probability value, a sufficient amount of user feedback must be collected. Assuming an average user visits 130 websites per day, we estimate that the probability of a user providing input for a given website is 0.0077. To update the phishing probability value by at least 10% of its current value (p), an average scaled feedback value of 0.1p or higher is required. For a current probability value of p = 0.5, this equates to at least 10 users providing feedback for the website. However, the required number of feedback may vary based on the initial probability value, the desired update threshold, and the distribution of user feedback scores.

IV. RESULTS & DISCUSSION

The following are the results of Chrome extension testing on various URLs:

1. Legitimate Website:
   URL: https://www.youtube.com/

Fig. 4. Legitimate URL

2. Phishing Website:
   URL: https://m.info.packaged.1rx5.cfd/

Fig. 5. Phishing URL

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

3. User Feedback for Website:
   a. Before Update:
      URL: https://docs.google.com/presentation/u/0/

As shown in Figure 6, the phishing probability was 30.37%.

   b. After Update:
      URL: https://docs.google.com/presentation/u/0/

As shown in Figure 7, after registering multiple feedback from users, the phishing probability was updated to 24.7%.

Fig. 6. Initial Prediction

Fig. 7. After Update

We have implemented a machine learning model and a neural network. Furthermore, we have combined both results to enhance the prediction accuracy of classification. Additionally, user feedback was utilised to improve predictions. This work lays the foundation for future research and provides a basis for fine-tuning the system to achieve a suitable balance between accuracy and performance without compromising either one.

V. CONCLUSION & FUTURE WORK

This research paper proposes a framework for detecting phishing websites in real-time browsing environments. Despite the existence of numerous machine learning-based solutions for phishing detection, there is a lack of analysis and research regarding their effectiveness in live browsing environments. Therefore, our proposed framework aims to address this gap in the current literature. Future research can explore the implementation and validation of this framework to improve the overall security of online browsing.

Our research has demonstrated the effectiveness of utilising feedback data to enhance the performance of machine learning models. A high-quality dataset is crucial for model training, and incorporating user feedback can substantially improve the accuracy of a model's predictions. Our proposed methodology uses feedback data to determine user-specific probabilities. By employing this approach, we have demonstrated how machine learning models can achieve improved performance. Future research can explore the application of this feedback-driven methodology in various domains to enhance the quality of machine learning models.

The deployment of the entire system to a cloud platform is planned for future work. This would allow for greater scalability and accessibility of the system. By moving to the cloud, the system would be able to accommodate a larger number of users and be more easily accessible from different locations. This would also provide an opportunity further to evaluate the system's performance in a cloud environment. Future research could investigate the potential benefits and drawbacks of cloud deployment, as well as assess the system's performance in this environment.

We will publish the browser plugin on the Chrome Web Store. In addition, we will make the tool ``Mr Phish'' available for other browsers.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	Not relevant.
Availability of Data and Material/ Data Access Statement	Dataset links are provided in the article.
Authors Contribution	All authors have equal participation in this article.

REFERENCES

1. L. Tang and Q. H. Mahmoud, “A Deep Learning-Based Framework for Phishing Website Detection,” in *IEEE Access*, vol. 10, pp. 1509-1521, 2022, doi: 10.1109/ACCESS.2021.3137636. [CrossRef]
2. A. Alswailem, B. Alabdullah, N. Alrumayh and A. Alsedrani, “Detecting Phishing Websites Using Machine Learning,” 2019 2nd *International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, 2019, pp. 1-6, doi: 10.1109/CAIS.2019.8769571. [CrossRef]

3. G. Xiang, J. Hong, C. Rose, and L. Cranor. 2011. "CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites," *ACM Trans. Inf. Syst. Secur*. 14, 2, Article 21 (September 2011), 28 pages, doi: 10.1145/2019599.2019606. [CrossRef]
4. M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, "Efficient Deep Learning Techniques for the Detection of Phishing Websites," *Sadhana*, vol. 45, no. 1, pp. Jun. 2020, doi: 10.1007/s12046-020-01392-4. [CrossRef]
5. P. Zhao and S. C. Hoi, "Cost-sensitive online active learning with application to malicious URL Detection," in Proceedings of the 19th *ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 919–927, doi: 10.1145/2487575.2487647. [CrossRef]
6. R. W. Purwanto, A. Pal, A. Blair, and S. Jha, "PhishSim: Aiding Phishing Website Detection With a Feature-Free Tool," in *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1497-1512, 2022, doi: 10.1109/TIFS.2022.3164212. [CrossRef]
7. S. C. Jeeva and E. B. Rajsingh, "Intelligent Phishing URL Detection using Association Rule Mining," *Human-centric Computing and Information Sciences*, vol. 6, no. 1, p. 10, 2016, doi: 10.1186/s13673-016-0064-3. [CrossRef]
8. S. Maurya, H. Singh, and A. Jain, "Browser Extension Based Hybrid Anti-phishing Framework using Feature Selection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 1–10, 2019, doi: 10.14569/IJACSA.2019.0101178. [CrossRef]

AUTHORS PROFILE

Mitesh M. Adake is a final-year undergraduate student pursuing Computer Engineering from SCTR's Pune Institute of Computer Technology, Pune, India. During his academic journey, he completed two internships where he worked on data handling and machine learning product development. He has a keen interest in the field of Data Science, Machine Learning, and Quantum Computing. He is also a Qiskit Advocate, actively contributing to the Qiskit Community - an open-source SDK for working with quantum computers. With his interest and passion for research, he aims to make meaningful contributions to the field of Machine Learning and Quantum Computing.

Atharva M. Belekar is a final-year undergraduate student pursuing Computer Engineering from SCTR's Pune Institute of Computer Technology, Pune, India. With a keen interest in cybersecurity and computer networks, he completed an internship where he worked on network security. He is a Fortinet Certified Network Security Associate, which demonstrates his expertise in network security. With his academic background and hands-on experience in the field, he plans to pursue a career in cybersecurity and computer networks. With a passion for learning and a drive to succeed, he is committed to contributing to the field of cybersecurity through innovative research and practical applications.

Chinmay U. Ambekar is a final-year undergraduate student pursuing Computer Engineering at SCTR's Pune Institute of Computer Technology, Pune, India. He has an interest in developing software applications. He has a passion for designing and building responsive and user-friendly web applications using the latest front-end and back-end technologies. In addition to academics, he has completed two internships, where he developed responsive and user-friendly web applications. He plans to pursue a career in software development, hoping to utilise his skills to create innovative solutions that enhance people's lives.

Dipika D. Bhaiyya received a bachelor's degree in Computer Engineering from North Maharashtra University, Jalgaon, India, in 2009 and an M. Tech degree in Computer Science & Engineering from Jawaharlal Technological University, Hyderabad, India, in 2014. She is certified in Cybersecurity Essentials by Cisco Networking Academy. Currently, she works as an Assistant Professor in the Computer Engineering Department at SCTR's Pune Institute of Computer Technology, Pune, Maharashtra, India. Her research interests include deep learning and cybersecurity.