International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 5
ISSN: 2088-5334

Phishing and Spoofing Websites: Detection and Countermeasures

Wee Liem Lai [a], Vik Tor Goh [a,*], Timothy Tzen Vun Yap [b], Hu Ng [c]

[a] *Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia*
[b] *School of Mathematical & Computer Sciences, Heriot-Watt University, 62200 Putrajaya, Malaysia*
[c] *Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia*
*Corresponding author: *vtgoh@mmu.edu.my*

***Abstract*— Website phishing and spoofing occur when unsuspecting users are tricked into interacting with a fraudulent website designed to impersonate a legitimate one. This is done with the intention of stealing login credentials or other personal information. The goal of this project is to develop a multi-layered URL-based malicious website detection system to counter such attacks. The proposed system employs several defence mechanisms, including whitelist filtering, API requests to domain blacklist providers, and string comparison algorithms, to accurately identify and classify websites as either legitimate or malicious. In brief, the first layer provides an initial check by matching the domain of the intended website with a predefined whitelist, while the second layer queries APIVoid (a domain blacklist provider) to conduct additional checks for domain age and reputation. Finally, to prevent typographical errors that could unintentionally redirect users to a malicious website, the last layer compares the domain of the intended website with entries in the whitelist to identify any significant similarities using the Levenshtein distance algorithm. To evaluate the system's performance, a comprehensive testing phase was conducted on a dataset containing 30 randomly selected websites, encompassing various scenarios of malicious and legitimate websites. The results show a high true positive rate of 0.94 and an overall accuracy of 0.93, indicating the system's ability to accurately classify legitimate and malicious websites. The proposed system shows promising results in accurately classifying websites and enhancing user awareness to prevent phishing and spoofing attacks.**

***Keywords*— Phishing attacks; domain name spoofing; user alert system; multilayer malicious website detection model.**

Manuscript received 22 Nov. 2022; revised 15 Mar. 2023; accepted 17 Jul. 2023. Date of publication 31 Oct. 2023.

I. INTRODUCTION

In today's interconnected digital world, phishing and spoofing attacks have become a significant threat, exploiting vulnerabilities in communication channels and posing risks to individuals and organizations. The increasing reliance on digital technologies and the surge in online activities, especially during the COVID-19 pandemic, have allowed cybercriminals to launch malicious activities. The lack of user education and awareness further contributes to the success of these attacks. A multilayer malicious website detection system is designed and developed to address this growing threat, incorporating various defense mechanisms to accurately identify and categorise websites as legitimate or malicious.

The main problem lies in users' ignorance and lack of awareness regarding phishing and spoofing attacks, leading to their susceptibility to deception by malicious attackers. Users often struggle to differentiate between real and fraudulent websites due to the increasing sophistication of these attacks. Hence, this project aims to develop an effective tool for identifying and mitigating phishing and spoofing attacks. The objectives include understanding social engineering attack techniques, evaluating detection and prevention mechanisms, proposing a domain-based verification algorithm, and designing a system to prevent users from accessing fraudulent websites.

A. Phishing and Spoofing Attacks

Cyber threats pose significant risks in today's interconnected digital world, as communication and transactions predominantly take place over the Internet. Phishing and spoofing have gained prominence due to their ability to deceive people. Phishing attacks involve malicious actors attempting to trick individuals into disclosing private information or performing specific actions by posing as trusted entities [1]. The term 'phishing' is derived from the analogy of fishing, where attackers cast a wide net to lure unsuspecting victims into their fraudulent schemes, similar to how a fisherman uses bait to catch fish [2].

Phishing attempts often occur through various communication channels, such as e-mail, webpages, phone calls, advertisements, or text messages [3]. This is done with the goal of obtaining valuable information for financial gain, identity theft, or unauthorised access to accounts and systems [4]. To make this social engineering attack more convincing, attackers frequently impersonate reputable companies, government organizations, or popular online services to gain victims' trust.

Spoofing attacks are closely related to phishing and play an important role in cyberattacks. Phishing aims to steal information, while spoofing is a technique used to enhance phishing attacks by posing as trusted sources to exploit users' trust. Several types of spoofing attacks exist, including e-mail spoofing, Internet Protocol (IP) spoofing, Domain Name System (DNS) spoofing, and website spoofing – which is the focus of our studies in this work. Website spoofing involves the creation of fake websites that mimic existing ones by copying their appearance, layout, and content [5]. Victims may be directed to these fraudulent websites via phishing e-mails or manipulated hyperlinks, allowing attackers to steal credentials and personal information.

Website spoofing has been particularly successful in tricking users because it exploits the behavior of users wanting to be quick and, therefore, not pay attention to details. This behavior is further exacerbated by a user's implicit trust in a website simply by confirming visual similarities at a glance. As a result, users unwittingly reveal their login credentials, which often leads to significant financial losses.

B. Related Works

A common method often employed by banking websites to let customers verify the genuineness of the website is called mutual authentication. An example can be seen in Fig. 1, where in Malaysia such mutual authentication methods are named SecureWord (CIMB Bank), Personal Login Phrase (Public Bank), Security Phrase (Maybank), or Secret Word (RHB Bank). Although named differently by their respective organizations, the principle remains the same: the banks present users with confidential information (i.e., secure word/phrase/image) that only the users will know. This allows the user to distinguish a legitimate website from a fraudulent one.

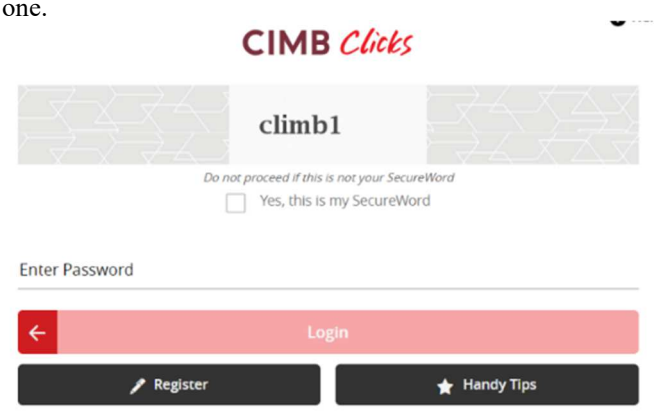

Fig. 1 SecureWord from CIMB Clicks.

Websites that do not employ this method raise the risk of users unintentionally keying their credentials into a fraudulent website [6]. However, this approach is not foolproof. For instance, the secure word or image could be replaced with an "under maintenance" message to deceive users. A study conducted in [7] found that the security image's effectiveness in preventing phishing attacks was limited. In their experiment, 73% of participants entered their passwords even when the security image or caption was not displayed, indicating that users are often careless and unaware of small differences on websites. This highlights the need to develop more effective schemes to better protect users from becoming victims of phishing website.

There are many approaches to detecting phishing websites, and [8] has divided these approaches into five categories: lists-based, visual similarity, heuristic, and machine learning techniques. List-based approaches such as those by [9]–[13] use either a whitelist, blacklist, or a variation to perform filtering that permits or denies access to websites. This approach is straightforward and has low false positive rates but can be easily bypassed by making small changes to the URL. Furthermore, the lists must be updated regularly for this approach to be effective.

On the other hand, visual similarity techniques perform an analysis of the visual features on a website. Studies in [14]–[19] compare the source code, page layout, website logo/icon, screenshots, and other visual elements of a suspicious website against a verified copy of the website that had been visited previously. This technique can detect visual cues that the user may have missed but requires more computational power and time because it uses image processing techniques for its analysis. Moreover, it cannot assess websites that have not been visited previously or even zero-day phishing attacks [18].

The heuristic approach is a feature-based method that analyses attributes or features of a suspicious website to identify details that could distinguish it from a legitimate website. Features such as URLs, website content, digital certificates, and metadata are gathered from various sources to be analyzed. The effectiveness of this approach depends on the reliability and trustworthiness of the external data sources, as well as the algorithms used for analysis. [20]–[23] used such an approach in their work.

Recent developments in machine learning and its application in various domains, such as agriculture [24], finance [25], medical [26], and environment [27], have made this approach very attractive, namely in its ability to learn and train itself to identify phishing websites. Research by [28]–[33] follow the same methodology where common attributes or features of phishing websites are first collected. Machine learning algorithms are then trained on this feature set to obtain a model representing phishing websites. Finally, the detection algorithm uses this model as the baseline upon which suspicious websites are compared against. According to [34], machine learning approaches have achieved more than 99% accuracy and have proven to be the most effective due to their adaptability. Despite that, the difficulty in applying machine learning techniques is the complexity of the algorithms and requirements for a large training data set.

In our work, we proposed a multilayer approach that utilizes some of the abovementioned techniques. By doing so, we can increase the effectiveness of the phishing detection system by adding more redundancies should one layer fail to detect a phishing website. This system is packaged as a Chrome extension for easy use and distribution.

II. Materials and Method

Due to their adaptive nature, cyberattacks constantly evolve to bypass existing defenses, making it challenging to prevent these attacks completely. As a response, a Chrome extension with a multilayer filter was developed as part of this project. The primary objective of this extension is to identify and prevent phishing websites by implementing multiple layers of filters and validations.

When a new tab is opened or activated, the extension performs several validations on the current website's domain to detect any malicious activities. Using multiple layers of filters, this system provides robust protection and promptly alerts users when they inadvertently access malicious websites. Real-time feedback is delivered through messages and alarms triggered for identified malicious websites.

The system comprises three distinct components: a domain-whitelist filter, API-based domain-blacklist filter, and finally, a string comparison filter. Each layer conducts specific verifications and validations, progressively enhancing the accuracy of the results.

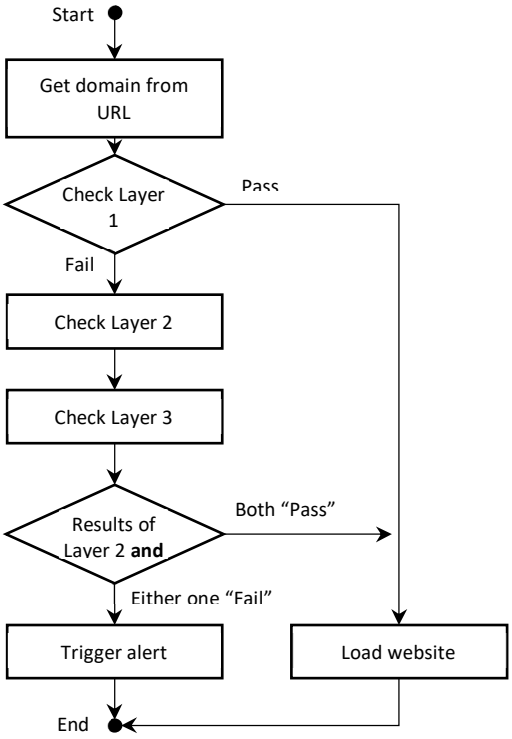

Fig. 2 Flowchart depicting the interactions between the three layers of the proposed phishing detection system.

A flowchart depicting the overall functionality of the phishing detection system is presented in Fig. 2. The layers within the phishing detection model are as follows:

- **Layer 1 – Domain-whitelist Filter:** Compares the domain against a predefined whitelist of trusted websites. If the domain is not found in the whitelist, the system automatically proceeds to the subsequent two layers.
- **Layer 2 – Domain-blacklist Filter**: Initiates API requests to APIVoid services [35] to assess blacklist detections and gather domain age data. A website is classified as malicious if it receives a high blacklist score or if it is relatively new.
- **Layer 3 – String Comparison Filter:** Compares the domain keyed in by the user with the predefined whitelist using the Levenshtein distance algorithm. If the similarity score is high and surpasses a threshold, it is deemed a potential typographical error, and an intended URL is suggested instead.

The Chrome extension employs event listeners and functions to conduct multilayer phishing detection. Upon opening a new tab, an event listener is activated, and the URL of the current tab is retrieved. The *Layer1* function is then invoked to ascertain whether the domain is listed in the whitelist. If it is, the extension allows the website to load normally. However, in cases where the domain is absent from the whitelist, the program proceeds to execute the *Layer2* and *Layer3* functions for further validation.

In Layer2, API requests are initiated to APIVoid to assess the domain's reputation. If the domain is flagged as malicious, an alert is triggered to notify the user. Finally, Layer3 compares the domain of the active tab against the whitelist from Layer1, utilizing string similarity as the basis. If the spelling is very similar but not exactly the same, this could indicate a typographical error or a domain spoofing attempt, thus triggering an alert as well. Details of these layers are described in the following sections.

A. Layer 1 – Whitelist Filter

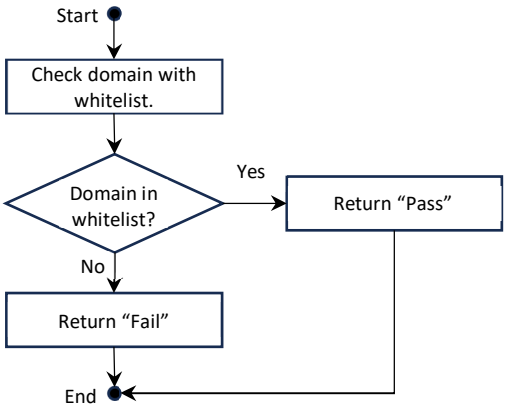

Fig. 3 Flowchart depicting functionality of Layer1.

The Layer1 function performs the initial domain validation against a predefined whitelist of popular and validated websites as shown in Fig. 3. It uses an *if-else* statement to check if the domain is included in the whitelist. If it is, the function returns a "Pass" flag, indicating that all is well. Conversely, if the domain is not found in the whitelist, the function returns a "Fail" flag, indicating that the domain has failed verification. This causes the extension to proceed with subsequent layers of verification.

Overall, Layer1 acts as the first layer of defense by ensuring that only reputable and validated websites (e.g., banks, social media, and e-mail) listed in the whitelist are granted access. It provides a quick and accurate level of protection against potential phishing or fraudulent attempts on these websites. Users can modify the whitelist according to their preferences.

B. Layer 2 – Domain Blacklist Filter

The Layer2 function takes a domain as input and makes API calls to the APIVoid threat analysis platform [35] to

determine if the domain is potentially malicious. This layer uses the domain reputation and the domain age APIs. The domain reputation API checks if the domain is blacklisted by domain blacklist services such as ThreatLog, OpenPhish, Spam404, and PhishTank. This analysis helps identify potentially malicious websites. However, it should be noted that newly created malicious websites may not yet be blacklisted.

To address this, the domain age API is utilized to determine the age of the domain. Information such as the registration date and the domain's age in days can be retrieved to identify suspicious recently registered domains. Since [36] indicated that 71.4% of phishing websites stop displaying phishing activities after 30 days, we use this value as a threshold to determine the trustworthiness of the website. Any domain "younger" than 30 days will be treated as suspicious.

Fig. 4 shows that the Layer2 function extracts the *blacklisted* and *age* values from the domain reputation and age APIs, respectively. A *blacklisted* value that is greater than or equal to one indicates that one or more blacklist services have flagged the domain as malicious. On the other hand, an *age* value that is less than 30 days means that the domain is potentially malicious.

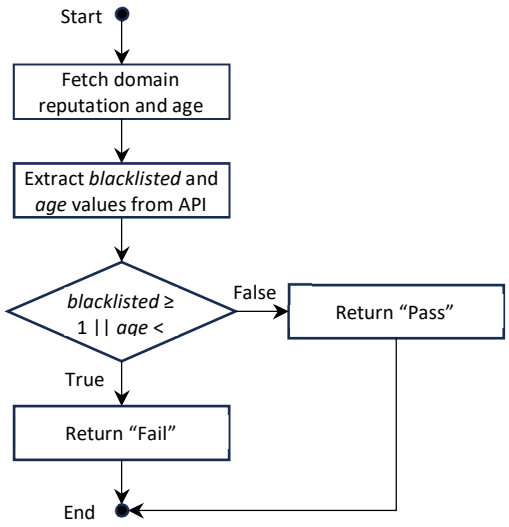

Fig. 4 Flowchart depicting functionality of Layer2.

The Layer2 function returns a "Pass" flag if both *blacklisted* **and** *age* values indicate a non-malicious domain. If either *blacklisted* **or** *age* indicates a malicious domain, the function returns a "Fail" flag and moves on to the next Layer3 for further checks.

C. Layer 3 – String Comparison Filter

Layer3's function is to prevent users from accidentally visiting spoofed websites with URLs that look like legitimate websites. The Layer3 utilizes the Levenshtein Distance algorithm [37] to compare the user's URL with a whitelist of domains and determine the similarity between them. The Levenshtein Distance, also known as the edit distance, is a metric that measures the minimum number of changes required to transform one string into another (insertions, deletions, or substitutions). This layer prevents users from mistakenly entering an erroneous domain name like those found in the whitelist, which could lead them to malicious websites. For example, www.g00gle.com instead of www.google.com. Even though www.g00gle.com is not valid, it could be a spoofed website designed to defraud unsuspecting users.

The algorithm constructs a matrix where the rows represent the characters from one string, and the columns represent the characters from another string. According to [38], the Levenshtein Distance between two strings, a and b can be calculated using the formula shown in Eq. 1.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min\begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad (1)$$

The final similarity index is obtained by subtracting the result of Eq. 1 from 1. A higher similarity value indicates a closer match. To detect a potentially spoofed website, Layer3 assumes that the spoofed website will have a very high similarity index (domain that looks the same to confuse the user) but not identical (i.e., similarity index of 1). If such a website is detected, Layer3 returns a "Fail" flag. Otherwise, it returns a "Pass" flag as shown in

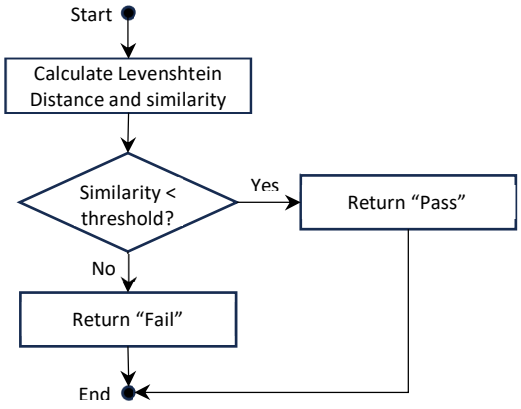

Fig. 5 Flowchart depicting functionality of Layer3.

The threshold for the similarity index upon which a website is assumed to be spoofed or otherwise is determined empirically and discussed further in Section III below.

III. Results and Discussion

A. Optimum Threshold of Layer3's Similarity Index

An empirical evaluation was conducted to determine the optimum threshold value for the string comparison functionality in Layer3. The evaluation focused on websites from the whitelist that had multiple subdomains by comparing them with their alternative domains. For example, websites like www.google.com have more than 50 different subdomains, including news.google.com, meet.google.com, and maps.google.com.

Similar websites such as www.nba.com and www.nfl.com were also chosen for testing. By comparing the domain names that appeared to be almost identical to those on the whitelist, the results in table 1 were obtained. Based on these results, an optimum threshold should be set at 0.818 or higher. Therefore, a threshold of 0.85 is chosen for the filter in Layer3. The threshold value denotes the minimum level of similarity required for a website to be considered an entry in the whitelist. Setting the threshold at 0.85 ensures that only

websites with a high degree of resemblance to the domain names in whitelist, thus indicating a spoofed website will return the "Fail" flag.

TABLE I
SIMILARITY BETWEEN SAMPLE DOMAIN SETS

Whitelist Domain	Similar Domain	Similarity
www.google.com	one.google.com	0.786
www.google.com	news.google.com	0.8
www.google.com	docs.google.com	0.733
www.ebay.com	www.ebay.de	0.75
www.ebay.com	www.ebay.com.au	0.8
www.facebook.com	m.facebook.com	0.8125
shopee.com.my	ads.shopee.com.my	0.765
www.yahoo.com	news.yahoo.com	0.786
www.nba.com	www.nfl.com	0.818
www.espn.com	www.epsa.com	0.75

B. System Evaluation

The system uses three layers to evaluate the legitimacy of the domains, resulting in different outcomes for various domain names. Table 2 provides a summary of the scenarios and their respective outcomes.

TABLE II
OUTCOMES FOR DIFFERENT DOMAINS

No.	Domain	Description
1	Verified website www.maybank2u.com.my	Layer1: Pass Outcome: Load website
2	Safe website www.good.com	Layer1: Fail Layer2: Pass Layer3: Pass Outcome: Load website
3	Blacklisted website www.bad.com	Layer1: Fail Layer2: Fail Layer3: Pass Outcome: Trigger alert
4	Typographical error www.maybank3u.com.my	Layer1: Fail Layer2: Pass Layer3: Fail Outcome: Trigger alert

Four possible scenarios can occur when the phishing detection model is applied to various URLs. These scenarios include:

- **Verified Website:** If the URL is on the whitelist, the model recognizes it as a verified website and allows the website to load. The process then ends without any further checks.
- **Safe Website:** www.good.com represents a trusted website but is not on the whitelist. If the website passes both Layer2 (*blacklisted* < 1 and *age* > 30) and Layer3 (similarity index > 0.85), it is classified as "Pass" and the website loads.
- **Blacklisted Website:** Layer2 determines using APIVoid that www.bad.com (hypothetical example of a blacklisted website) has the parameters *blacklisted* > 0 and *age* < 30. Hence, it is classified as a malicious website, and an alert is triggered.
- **Typo or Misspelled URL:** Another hypothetical URL, such as www.maybank3u.com.my, which is not on the whitelist yet passes Layer2, will have a similarity index > 0.85. This suggests that the user may have made a typographical error. As such, an alert is triggered, and a correct URL is suggested.

With these scenarios, a comprehensive study was conducted to evaluate the performance of the proposed model using a diverse set of URLs. The goal was to assess the effectiveness and feasibility of the model in real-world phishing prevention efforts. A list of 30 unique actual websites was generated, containing 17 malicious and 13 non-malicious websites. The evaluation results were carefully recorded to assess the accuracy and effectiveness of each layer within the model for detecting phishing websites.

- **True Positive (TP):** Malicious websites are correctly identified.
- **True Negative (TN):** Non-malicious and legitimate are correctly identified.
- **False Positive (FP):** Non-malicious and legitimate websites are misidentified as being malicious.
- **False Negative (FN):** Malicious websites are misidentified as being non-malicious and legitimate.

TABLE III
DETECTION RESULTS OF PROPOSED SYSTEM

Domain	Outcome	Result	Actual
www.rnudah.com	2	FN	TP*
www.freepik8888.com	4	FP	TN*
www.microsoft.com	2	TN	TN
www.instagram.com	1	TN	TN
zoom.us	1	TN	TN
todayjournal.net	1	TN	TN
www.oracle.com	2	TN	TN
www.netflix.com	1	TN	TN
www.speshbabies.com	3	TP	TP
www.alimama.com	4	TP	TP
www.hashmap.tw	3	TP	TP
www.malaysiaairlines.cow	4	TP	TP
davivienda.shop	3	TP	TP

Table 3 displays a list of samples, their corresponding outcomes, and the results obtained from the phishing detection system. Sixteen websites were correctly classified as True Positive (i.e., malicious), while another 12 websites were classified as True Negatives (i.e., non-malicious). However, it should be noted that the results showed one False Positive and one False Negative (FN). The system's performance is evaluated based on its accuracy in identifying different domain names across various scenarios. The system achieved a true positive rate (TPR) of 0.94 from the obtained results. The high TPR demonstrates the system's effectiveness in detecting malicious websites. Furthermore, the overall accuracy of the system is calculated to be 0.93. This suggests that the system is accurately categorizing websites with high accuracy across all scenarios.

IV. CONCLUSION

In conclusion, the phishing detection system developed in this project demonstrated its effectiveness in mitigating the risks associated with cyber-attacks and social engineering

tactics. By incorporating multiple layers of filtering and employing techniques such as domain reputation and string comparison, the system achieved high accuracy in identifying malicious websites. The evaluation results showed a TPR of 94% and an overall accuracy of 93%, indicating the system's ability to detect and distinguish between legitimate and malicious websites successfully.

However, certain challenges were encountered that highlighted the need for continuous improvements. Factors such as rigid threshold values used in the string comparison and invalid responses from API requests impacted the system's performance. Therefore, future research could explore additional features and patterns, such as webpage content analysis, HTML attribute examination, or visual element detection, to enhance the accuracy and effectiveness of the system. Additionally, developing real-time analysis techniques that dynamically monitor and assess website behavior can enable the system to adapt to emerging threats and detect new variations of malicious websites.

By leveraging techniques like network traffic analysis and behavioral monitoring, the system can provide a more comprehensive understanding of website activities and detect any changes that may indicate malicious intent. This dynamic approach would improve the system's ability to detect sophisticated phishing attempts and overcome traditional detection methods. Continued research and development in these areas will contribute to the ongoing efforts to prevent phishing and spoofing attacks and safeguard users' online security.

ACKNOWLEDGMENT

We sincerely appreciate and express gratitude for financial support from the Ministry of Higher Education, Malaysia, under the Fundamental Research Grant Scheme with grant number FRGS/1/2022/ICT07/MMU/03/1.

REFERENCES

[1] R. Alabdan, "Phishing Attacks Survey: Types, Vectors, and Technical Approaches," *Future Internet 2020, Vol. 12, Page 168*, vol. 12, no. 10, p. 168, Sep. 2020, doi: 10.3390/FI12100168.

[2] A. K. Jain and B. B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterp Inf Syst*, vol. 16, no. 4, pp. 527–565, 2021, doi: 10.1080/17517575.2021.1896786.

[3] M. P. Bach, T. Kamenjarska, and B. Žmuk, "Targets of phishing attacks: The bigger fish to fry," *Procedia Comput Sci*, vol. 204, pp. 448–455, Jan. 2022, doi: 10.1016/J.PROCS.2022.08.055.

[4] R. S. Rao, T. Vaishnavi, and A. R. Pais, "Phishdump: a multi-model ensemble based technique for the detection of phishing sites in mobile devices," *Pervasive Mob Comput*, vol. 60, p. 101084, Nov. 2019, doi: 10.1016/j.pmcj.2019.101084.

[5] R. S. Rao, T. Vaishnavi, and A. R. Pais, "Catchphish: detection of phishing websites by inspecting urls," *J Ambient Intell Humaniz Comput*, vol. 11, no. 2, pp. 813–825, Feb. 2020, doi: 10.1007/s12652-019-01311-4.

[6] R. Di Pietro, G. Me, and M. A. Strangio, "A two-factor mobile authentication scheme for secure financial transactions," *4th Annual International Conference on Mobile Business, ICMB 2005*, pp. 28–34, 2005, doi: 10.1109/ICMB.2005.12.

[7] J. Lee, L. Bauer, and M. L. Mazurek, "The effectiveness of security images in internet banking," *IEEE Internet Comput*, vol. 19, no. 1, pp. 54–62, 2015, doi: 10.1109/MIC.2014.108.

[8] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023, doi: 10.1016/J.JKSUCI.2023.01.004.

[9] T. Nathezhtha, D. Sangeetha, and V. Vaidehi, "WC-PAD: Web crawling based phishing attack detection," *Proceedings - International Carnahan Conference on Security Technology*, vol. 2019-October, Oct. 2019, doi: 10.1109/CCST.2019.8888416.

[10] R. S. Rao and A. R. Pais, "Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach," *J Ambient Intell Humaniz Comput*, vol. 11, no. 9, pp. 3853–3872, Sep. 2020, doi: 10.1007/S12652-019-01637-Z/METRICS.

[11] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, "Adopting automated whitelist approach for detecting phishing attacks," *Comput Secur*, vol. 108, p. 102328, Sep. 2021, doi: 10.1016/J.COSE.2021.102328.

[12] S. Maroofi, M. Korczynski, C. Hesselman, B. Ampeau, and A. Duda, "COMAR: Classification of Compromised versus Maliciously Registered Domains," *Proceedings - 5th IEEE European Symposium on Security and Privacy, Euro S and P 2020*, pp. 607–623, Sep. 2020, doi: 10.1109/EUROSP48549.2020.00045.

[13] P. A. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," *Comput Secur*, vol. 104, p. 102123, May 2021, doi: 10.1016/J.COSE.2020.102123.

[14] Y. Wang, Y. Liu, T. Wu, and I. Duncan, "A Cost-Effective OCR Implementation to Prevent Phishing on Mobile Platforms," *International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2020*, Jun. 2020, doi: 10.1109/CYBERSECURITY49315.2020.9138873.

[15] A. S. Bozkir and M. Aydos, "LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition," *Comput Secur*, vol. 95, p. 101855, Aug. 2020, doi: 10.1016/J.COSE.2020.101855.

[16] R. S. Rao and A. R. Pais, "Jail-Phish: An improved search engine based phishing detection system," *Comput Secur*, vol. 83, pp. 246–267, Jun. 2019, doi: 10.1016/J.COSE.2019.02.011.

[17] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, May 2019, doi: 10.1016/J.FUTURE.2018.11.004.

[18] A. K. Jain and B. B. Gupta, "Two-level authentication approach to protect from phishing attacks in real time," *J Ambient Intell Humaniz Comput*, vol. 9, no. 6, pp. 1783–1796, Nov. 2018, doi: 10.1007/s12652-017-0616-z.

[19] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites," *IEEE Access*, vol. 8, pp. 142532–142542, 2020, doi: 10.1109/ACCESS.2020.3013699.

[20] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput Commun*, vol. 175, pp. 47–57, Jul. 2021, doi: 10.1016/J.COMCOM.2021.04.023.

[21] Y. Ding, N. Luktarhan, K. Li, and W. Slamu, "A keyword-based combination approach for detecting phishing webpages," *Comput Secur*, vol. 84, pp. 256–275, Jul. 2019, doi: 10.1016/J.COSE.2019.03.018.

[22] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft comput*, vol. 23, no. 12, pp. 4315–4327, Jun. 2019, doi: 10.1007/S00500-018-3084-2/METRICS.

[23] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput Appl*, vol. 31, no. 8, pp. 3851–3873, Aug. 2019, doi: 10.1007/S00521-017-3305-0/METRICS.

[24] P. Athisaya Sakila Rani, Ns. Singh, and A. Professor, "Paddy Leaf Symptom-based Disease Classification Using Deep CNN with ResNet-50," *International Journal of Advanced Science Computing and Engineering*, vol. 4, no. 2, pp. 88–94, Aug. 2022, doi: 10.30630/IJASCE.4.2.83.

[25] F. Zulfikri, D. Tryanda, A. Syarif, and H. Patria, "Predicting Peer to Peer Lending Loan Risk Using Classification Approach," *International Journal of Advanced Science Computing and Engineering*, vol. 3, no. 2, pp. 94–100, Oct. 2021, doi: 10.30630/IJASCE.3.2.57.

[26] P. Chaudhari, "Skin Cancer Classification Application Using Machine Learning," *International Journal of Data Science*, vol. 2, no. 1, pp. 47–55, Sep. 2021, doi: 10.18517/IJODS.2.1.47-55.2021.

[27] M. Yamin and A. F. Giyats, "Support Vector Regression Approach for Wind Forecasting," *International Journal of Advanced Science Computing and Engineering*, vol. 4, no. 2, pp. 95–101, Aug. 2022, doi: 10.30630/IJASCE.4.2.84.

[28] V. Patil, P. Thakkar, C. Shah, T. Bhat, and S. P. Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach," *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, Jul. 2018, doi: 10.1109/ICCUBEA.2018.8697412.

[29] K. L. Chiew, C. L. Tan, K. S. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf Sci (N Y)*, vol. 484, pp. 153–166, May 2019, doi: 10.1016/J.INS.2019.01.064.

[30] A. Basit, M. Zafar, A. R. Javed, and Z. Jalil, "A Novel Ensemble Machine Learning Method to Detect Phishing Attack," *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*, Nov. 2020, doi: 10.1109/INMIC50486.2020.9318210.

[31] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst Appl*, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/J.ESWA.2018.09.029.

[32] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R. E. Ulfath, and S. Hossain, "Phishing attacks detection using machine learning approach," *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, pp. 1173–1179, Aug. 2020, doi: 10.1109/ICSSIT48917.2020.9214225.

[33] A. B. Altamimi *et al.*, "PhishCatcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning," *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3287226.

[34] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen, and R. Ramli, "A comprehensive survey on identification and analysis of phishing website based on machine learning methods," *ISCAIE 2021 - IEEE 11th Symposium on Computer Applications and Industrial Electronics*, pp. 82–87, Apr. 2021, doi: 10.1109/ISCAIE51753.2021.9431794.

[35] APIVoid, Threat Analysis APIs for Threat Detection & Prevention, https://www.apivoid.com/ (accessed Aug. 25, 2023).

[36] M. Sytnik and E. Bubnov, "An analysis of the life cycle of phishing and scam pages | Securelist." https://securelist.com/phishing-page-life-cycle/105171/ (accessed Aug. 25, 2023).

[37] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Trans Pattern Anal Mach Intell*, vol. 29, no. 6, pp. 1091–1095, Jun. 2007, doi: 10.1109/TPAMI.2007.1078.

[38] S. Grashchenko, "Levenshtein Distance Computation | Baeldung on Computer Science." https://www.baeldung.com/cs/levenshtein-distance-computation (accessed Aug. 28, 2023).