User-Centric Web Application for Phishing URL Detection by Machine Learning Model

Palida Yingwatchara
International School of Engineering (ISE)
Faculty of Engineering
Chulalongkorn University
Bangkok, Thailand
poonpalida@gmail.com

Pavarit Wiriyakunakorn
International School of Engineering (ISE)
Faculty of Engineering
Chulalongkorn University
Bangkok, Thailand
pavarit.guide@gmail.com

Thiti Srikao
International School of Engineering (ISE)
Faculty of Engineering
Chulalongkorn University
Bangkok, Thailand
thiti.chopper555@gmail.com

Sirin Nitinawarat
International School of Engineering (ISE)
Faculty of Engineering
Chulalongkorn University
Bangkok, Thailand
sirin.n@chula.ac.th

***Abstract*— Phishing attacks threaten individuals and organizations globally. While existing phishing detection tools predominantly focus on technical aspects, there exists a notable void in addressing the informational needs of users beyond numerical data. Additionally, there is a lack of studies focusing on users and few have explored feature selection techniques in machine learning-based phishing detection. To bridge this gap, we have developed a user-centric phishing URL detection tool powered by machine learning models. The objective is twofold: to effectively categorize phishing links and to provide users with contextual understanding and educational resources to enhance their online safety. This paper offers unique contributions by a comprehensive, user-focused approach to phishing detection combined with ML techniques. For the ML aspects, three feature selection approaches—Recursive Feature Elimination with Cross-Validation (RFECV), Particle Swarm Optimization (PSO), and Random Forest selection—were explored using three different machine learning algorithms: Random Forest, LightGBM, and SVC. The study found that the combination of LightGBM with RFECV provided the highest performance, achieving an accuracy of 95.07% after hyperparameter tuning. By focusing on user-centric design, this research not only enhances phishing detection capabilities but also empowers users with the knowledge to make informed decisions, thereby improving overall online safety.**

Keywords—cyber security, feature selection, phishing, machine learning

I. Introduction

In today's digital world, the prevalence of online threats, particularly phishing attacks, presents significant challenges to personal cybersecurity [1]. Phishing, often disguised as legitimate communication, aims to trick users into revealing sensitive information such as passwords, financial details, or personal data [2]. These attacks exploit various vulnerabilities that make individuals susceptible to deception, including a lack of awareness about Uniform Resource Locators (URLs) and their functions, difficulty in distinguishing trustworthy URLs from potential threats, URL concealment or redirection, accidental clicks, and the inability to differentiate between legitimate and phishing URLs [3].

To address these multifaceted challenges, various approaches have emerged, including heuristic analysis, visual similarity assessment, list-based evaluations, and advanced techniques rooted in machine learning and deep learning [4], [5]. These approaches are considered automated anti-phishing solutions [6]. However, while these automated approaches exist, they cannot guarantee complete protection due to their limited accuracy and the constantly evolving nature of phishing tactics. Therefore, end users ultimately serve as the last line of defense [7]. Numerous studies emphasize the importance of user awareness and education [6]-[8], advocating for user-centric approaches to phishing detection [9]-[11].

User-centered design plays a crucial role in enhancing the efficacy of phishing detection by creating interventions that are both technically sound and user-friendly for real-world application [7], [9]. Despite this emphasis, research indicates that developers often overlook users' decision-making processes, leading to user-centric weaknesses and usability issues that increase vulnerability to phishing attacks [9]. Additionally, available web applications for phishing intervention primarily focus on the technical aspects of detecting phishing links, leaving a notable gap in addressing users' informational needs comprehensively [12]. Users, often lacking deep technical knowledge, require more than just numerical data or verdicts; they seek contextual understanding and educational resources to navigate the digital landscape safely [13], [14].

Moreover, current research in phishing detection has predominantly focused on feature selection as a critical component of machine learning models [15]-[18]. However, the depth of exploration into feature selection techniques remains insufficient in many cases since existing approaches to feature selection are often heuristic-based [18], [19]. This limitation underscores the necessity for further research to explore and compare a broader range of feature selection approaches, ensuring that the most pertinent and effective features are identified to bolster robust phishing detection models [20].

Recognizing this critical gap, this article aims to develop a comprehensive user-centric phishing URL detection tool enhanced with machine learning models. This endeavor includes a thorough exploration of feature selection techniques to identify the most effective features for robust phishing detection. This approach enhances users' awareness of phishing risks and empowers them to make informed decisions online. The tool bridges the gap between technical detection and users' informational needs, contributing to a stronger defense against phishing attacks.

979-8-3503-5395-2/24/$31.00 ©2024 IEEE

II. Design Description

The design aims to provide effective phishing detection through two main components: Model Development, which trains a machine learning model to distinguish between safe and phishing URLs, and a Web Application, serving as the user interface for accessing detection functionalities.

A. Model Development

1.) Data Preparation: Data preparation ensures the dataset is ready for machine learning model training, starting with collecting both phishing and safe URLs. Next, feature extraction is performed on each URL to create a comprehensive feature set. After extracting features, duplicate entries are removed from the dataset to ensure each URL is represented uniquely, eliminating redundancy. Data balancing is then conducted to prevent model bias by randomly selecting URLs to ensure an equal number of phishing and safe URLs. This balanced dataset is verified to maintain the diversity and variety of features necessary for accurate model training.

2.) Feature Engineering and Experimentation: Extracted features undergo experiments for feature selection and model training. The best model is then selected for deployment in the web application. The chosen model uses the LightGBM algorithm and Recursive Feature Elimination with Cross-Validation (RFECV) for feature selection. Experiment details will be discussed in Sections III and IV.

B. Web Application

The goal of developing the web application is to effectively utilize machine learning models, achieved through the development of front-end and back-end applications.

Fig. 1. Overview of the application structure and technologies in each part.

1) Front-end Application: The Front-end application must allow users to input URLs for scanning, display scan results, and navigate through previous results. Choosing the right technology and deployment cost are crucial for establishing a solid development baseline.

a) Framework. NextJS is chosen to take advantage of its server-side rendering, routing, and other features it provides out of the box. With Typescript, static typing can further help improve code quality during the development process. Tailwind CSS is implemented to eliminate the need to write CSS classes for each individual UI component.

b) Deployment. The Front-end application is deployed on Vercel. The service is chosen due to the ease of deployment and other useful features such as automatic scaling, continuous deployment, and built-in analytics.

2) User Interface Design: The design needs to follow concepts such as simplicity, clarity, and consistency, which can be achieved by following UI/UX best practices.

a) URL Scan Page. Miller's Law and Gestalt Principles are utilized to help create a more cohesive UI by reducing distance between elements and giving them appropriate sizing with appropriate grouping to improve ease of use.

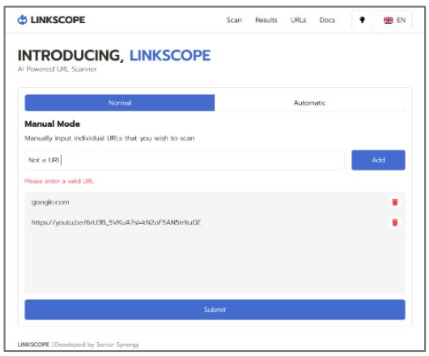

Fig. 2. Screenshot of the scan page.

b) Search Page. This page utilize Jakob's Law, which focuses on the use of UI design with established conventions. In this case, search and filters follows the design that would be commonly found on other established websites or applications.

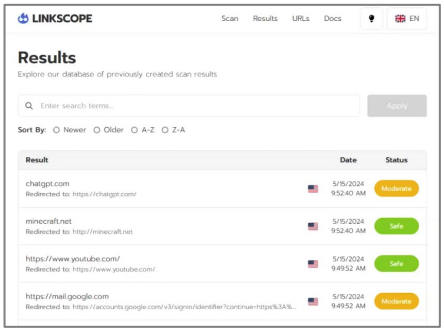

Fig. 3. Screenshot of the search page.

c) Result Page. To manage the extensive information on this page effectively, Miller's Law is applied by breaking down information into separate parts. This allows users to focus on each section of the results without feeling overwhelmed.

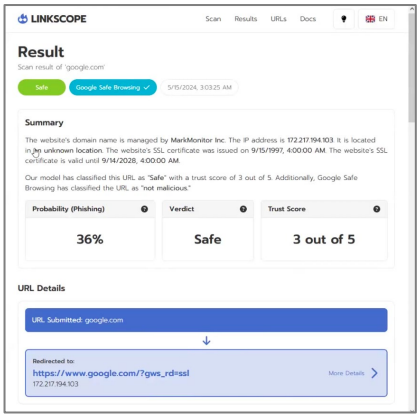

Fig. 4. Screenshot of the result page.

3) Back-end Application: The back-end application needs to facilitates all the requests from the Front-end application.

a) Python for Back-end.. FastAPI is utilized to create RESTful APIs for the Front-end application. It also unifies the need for a python environment in the back-end application for the serialized trained model to be deployed and utilized

b) Database. MySQL is chosen as the relational database management system due to its robust support for a wide range of applications and its seamless integration with AWS.

c) Deployment. AWS Lambda was chosen for the back-end due to its cost efficiency and sustainability benefits.

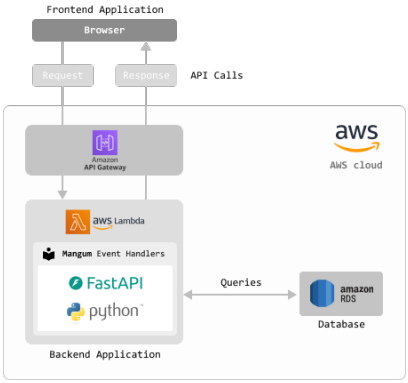

Fig. 5. Diagram showing the back-end application structure and its interactions within the application..

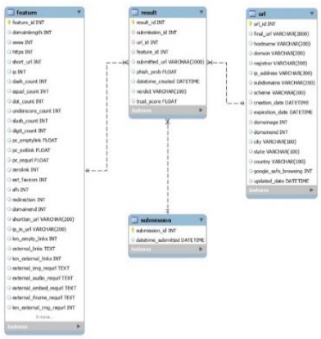

Fig. 6. Entity relationship diagram.

III. Experiment Setup and Methods

The experimental setup includes developing a phishing detection model and usability testing to understand users' needs. This section is divided into two parts: experimental setup and methods for both the model experiment and usability testing.

A. Model Experiment

For the Model Experiment Setup, four components must be defined before conducting the experiment: dataset, features explored, feature selection approaches, and algorithms used.

1) Dataset: The dataset consists of a total of 70,000 URLs, with 35,000 phishing URLs collected from PhishTank and PhishScore, and 35,000 safe URLs derived from top SEO domain lists. All the links have been checked, and no duplication was found. The data then underwent k-fold cross-validation with k= 5. This technique divides the dataset into five equally sized subsets (folds). For each fold, the model is trained on four folds (56,000 URLs) and tested on the remaining fold (14,000 URLs). This process is repeated five times, with each fold serving as the test set exactly once. This approach ensures that the model is validated on different subsets of the data, providing a more robust evaluation of its performance.

2) Features Explored: In this study, 29 features are selected for analysis, categorized into four types:

a) Address Bar-based: This category examines the characteristics of the URL address.

TABLE I. Address Bar Based Features Explored.

Feature Name	Explanation
domainlength	count the characters in hostname string
www	If url has 'www' as subdomain then return 0, else 1
subdomain	If url has more than 1 subdomain then return 1, else 0
https	If url contains 'https' then return 0, else 1
http	If url contains 'http' then return 0, else 1
short_url	If url is a short url returns 1,else 0
ip	If url contains 'ip address' returns 1,else 0
@	Count the '@' characters in url
-	Count the '-' characters in url
=	Count the '=' characters in url
.	Count the '.' characters in url hostname
_	Count the '_' characters in url
/	Count the '/' characters in url
digit	Count the digits (0-9) characters in url
log	If url contains 'log' word in url then return 0, else 1
pay	If url contains 'pay' word in url then return 0, else 1
web	If url contains 'web' word in url then return 0, else 1
cmd	If url contains 'cmd' word in url then return 0, else 1
account	If url contains 'account' word in url then return 0, else 1

b) HTML/DOM Structure-based: This category involves analyzing the HTML or DOM structure of the web page.

TABLE II. HTML/DOM Structure Based Features Explored.

Feature Name	Explanation
pcemptylinks	Percentage of empty links, which are hyperlinks that do not navigate to a different web page, often resulting in staying on the current page or displaying a blank page.
pcextlinks	Percentage of empty links, which are hyperlinks that do not navigate to a different web page, often resulting in staying on the current page or displaying a blank page.
pcrequrl	Percentage of external resource URLs, such as images or embedded content, hosted on a different domain than the main or submitted URL.
zerolink	If the URL page has no links in the HTML body, return 1; otherwise, return 0.
extfavicon	If the favicon URL is from a different domain than the submitted URL, return 1; otherwise, return 0.

submit2email	If the html page contains "\\b(mail\(\)|mailto:?)\b" then return 1, else 0
sfh	SFHs that contain an empty string or “about:blank” or lead to different domain sites from submitted url, like form['action'] == "" or form['action'] == "about:blank" then return 1, else return 0

c) Abnormal-based: This category assesses the abnormal behavior of the submitted URL.

TABLE III. ABNORMALITY BASED FEATURES EXPLORED.

Feature Name	Explanation
redirection	Returns 1 if clicking the submitted URL redirects to another URL; otherwise, returns 0.

d) Domain-based: This category considers factors related to domain registration and expiration dates.

TABLE IV. LIST OF DOMAIN BASED FEATURES EXPLORED.

Feature Name	Explanation
domainage	The difference between expiration time and creation time, if the domain age is less than 6 months then return 1, else return 0.
domainend	Returns 1 if the difference between the current date and the expiration date (registration length) is less than or equal to one year; otherwise, returns 0.

3) Feature Selection Approaches: The purpose of feature selection is to select a subset of relevant features from a large number of available features to achieve similar or even better classification performance than using all features.

a) Recursive Feature Elimination with Cross-Validation (RFECV): RFECV is a robust technique for feature selection that combines recursive feature elimination with the power of cross-validation. The process begins by building a model with all available features and then recursively removing the least significant features, one by one. At each step, the model is evaluated using cross-validation to determine its performance. This approach helps to identify the optimal set of features that contributes to the best model performance while avoiding overfitting.

b) Particle Swarm Optimization (PSO): PSO is a population based technique to address feature selection problems in this project due to better representation, capability of searching large spaces, and being less expensive computationally. In PSO, a group of candidate solutions, known as a swarm, is represented by particles within a defined search space. The algorithm begins by randomly initializing the position of each particle. The particles then traverse the search space, adjusting their positions based on their own individual experiences, aiming to find the optimal solution.

c) Random Forest Feature Selection: Random Forest Feature Selection is an embedded method that uses the importance scores from a Random Forest model to select the most relevant features. As the model is trained, each feature's contribution to prediction is assessed by its frequency in splitting decision tree nodes. Features with higher scores are deemed more significant. This approach automatically identifies key features, reducing the need for manual selection, and is especially effective in high-dimensional data scenarios.

TABLE V. FEATURE SELECTION APPROACH COMPARISON

Feature Selection	Impact on Model Performance
RFECV	Improves performance by eliminating redundant features.
PSO	Achieves high performance by exploring a wide range of feature subsets.
Random Forest Feature Selection	Enhances interpretability and robust performance by using feature importance.

4) Algorithms: Algorithm selection involves choosing the most suitable algorithms for a specific task based on their performance characteristics. The following are the selected algorithms in three different approaches.

a) Random Forest (RF) Algorithm: RF is recognized for offering the highest accuracy among machine learning algorithms and is frequently used in phishing URL detection. RF addresses the problem of overfitting in decision trees.

b) LightGBM: Identified as the second-highest performer in the research. It is considered to be one of the best choices in various machine learning applications, including phishing URL detection, for its speed and efficiency.

c) Support Vector Classification (SVC): SVC is a machine learning method used to classify data into different categories, working by finding the best dividing line that separates data into its respective classes. It does this by maximizing the margin, which is the distance between the line and the nearest data points from each class.

According to Fig.1, the experimental process for selecting the best model begins after data preparation and feature extraction. The experiment encompasses three main processes: feature selection, algorithm selection, and model evaluation. For feature selection, as stated in Section 5.1.1, RFECV, PSO, and Embedded methods are used to select optimal features from the total of 29 features derived from the feature extraction process. This approach yields three optimal subsets of features, one from each selection method. Next, three algorithms—Random Forests, LightGBM, and SVC—are used to train models on each of these subsets.

Finally, model evaluation is conducted to select the best model and determine which subset of features and which algorithm produces the best performance. This assessment uses a range of performance metrics, including accuracy, precision, recall, and false positive rate. This comprehensive evaluation process allows for an accurate and reliable assessment of the model's effectiveness in detecting phishing attempts.

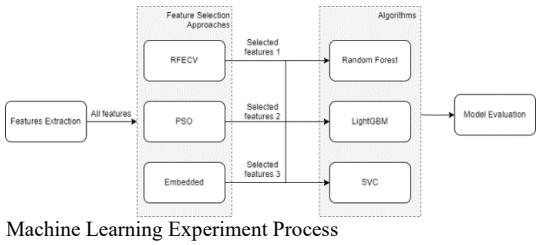

Fig. 7. Machine Learning Experiment Process

B. *Usability Testing*

Our primary objective in conducting usability testing for our phishing detection application is to assess its effectiveness, efficiency, and user satisfaction. To ensure diverse perspectives, the test engaged individuals from various backgrounds. Testing took place online via platforms like Zoom and Discord, as well as in-person. Participants began with an introductory phase, providing demographic information and details of prior experience with similar apps. During testing, participants performed tasks including using the scan and search functions. Task completion times and errors were recorded for performance evaluation. Following task completion, post-interviews were conducted to gather comprehensive feedback on usability, user satisfaction, and areas for potential improvement, ensuring alignment with user needs and expectations. Additionally, participants completed a questionnaire using a Likert scale to assess usability across various dimensions including self-efficiency, ease of use, user-friendliness, behavioral intention, and security awareness.

In assessing the usability of our phishing detection application, we focused on several key dimensions: self-efficiency, measuring users' independence and need for assistance; ease of use, evaluating navigation and overall design; user-friendliness, including design organization and visual appeal; behavior intention, gauging future usage likelihood; and security awareness, assessing how well the app conveyed its value in phishing detection and education. These dimensions collectively provided insights into the application's usability and user satisfaction.

IV. Results and Discussions

A. *Model Results*

The results are obtained from applying the machine learning algorithms with different feature selection approaches. In order to select the best model for the web application, LightGBM is chosen for its high accuracy, leading among three algorithms.

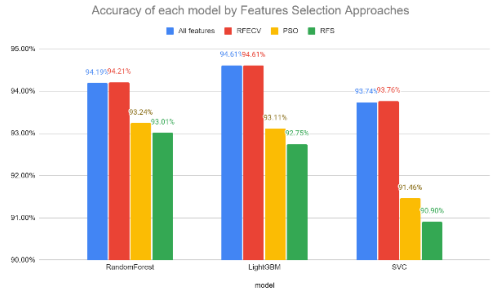

Fig. 8. Accuracy of each model by Feature Selection Approaches.

However, in the context of a web application, processing time is also a critical factor. When users submit URLs for analysis, the feature extraction process takes time, adding to the overall latency. This consideration makes it important to balance accuracy with processing time.

According to Fig. 8, LightGBM with all features and with RFECV provides approximately the same highest accuracies at 94.61%. However, having all features extracted is a drawback for the web application, where user experience depends on quick responses. Given this context, LightGBM with RFECV is chosen, reducing the features to 26.

TABLE VI. Results of LightGBM with RFECV

Algorithm	Accuracy	Precision	Recall	F1-score
LightGBM	94.61%	95.68%	93.44%	94.61%
LightGBM (Hyperparameterized)	95.07%	96.00%	94.05%	95.07%

According to Table VI, the classification results of LightGBM with RFECV are presented. The hyperparameter tuning process increased the accuracy to 95.07%. The features were selected using Recursive Feature Elimination with Cross-Validation (RFECV), resulting in a reduced set of 26 features. These features include 'domainlength', 'www', 'subdomain', 'https', 'short_url', '@', '-', '=', '.', '_', '/', 'digit', 'log', 'pay', 'web', 'account', 'pcemptylinks', 'pcextlinks', 'pcrequrl', 'zerolink', 'extfavicon', 'submit2email', 'sfh', 'redirection', 'domainage', and 'domainend'. This set of features, along with the hyperparameterized LightGBM model, is chosen for deployment in the application.

Additionally, the AUC score is 0.99 indicates the model's excellent ability to distinguish between phishing and safe URLs. This high score demonstrates the model's robustness in accurately identifying potential threats, reinforcing the importance of feature selection and analysis in achieving optimal performance.

While the LightGBM model with RFECV performs well in the current study, scaling it to larger datasets or more complex URL structures might present challenges. Future work should explore the model's performance with more extensive datasets and varied URL characteristics. As phishing tactics evolve rapidly, the model requires continuous evaluation and regular updates with new data to maintain high detection rates.

B. *Web Application Usability Testing Result*

From the results from usability testing, average duration for completing each task ranged from 30 seconds to 1, providing valuable insights into the clarity and user-friendliness of the application interface. Nonetheless, no errors were encountered during task completion, indicating that even novice users were able to navigate and utilize the application effectively.

The usability testing might have biases due to a limited user group or specific scenarios that do not cover all possible use cases. For example, the testing group is not diverse in terms of demographics, technology proficiency, or usage patterns, potentially limiting the accuracy of results across the broader user population. Expanding to a more diverse user base, including different backgrounds and familiarity with phishing threats, and testing in varied real-world scenarios can provide more representative and actionable feedback.

TABLE VII. Result from the Questionnaire.

Category	Question	Avg. Score	Category Avg. Score
Self efficiency	I can use the application skillfully	3.88	3.67
	I don't need specialist to help using the application	3.50	
	I feel confident using the application	3.63	

Ease of use	Using the application is easy and straightforward	3.75	4.04
	I find it easy to navigate the application	4.25	
	The steps to use the application are clear and simple	4.13	
User friendliness	The application has a user friendly interface	4.25	4.13
	The design of the application is visually appealing	4.00	
	The features in the application are well-organized and easy to find.	4.13	
Behavior Intention	I intend to use the application myself in the future	4.25	3.96
	I would recommend the application to my friends	4.13	
	I would likely choose the application over other similar tools in the future.	3.50	
Security Awareness	Using the application raises me awareness in security aspects	4.25	4.38
	I understand more about phishing websites after using the application	4.38	
	The application helps me recognize phishing website characteristics	4.50	
	Average Score	**4.03**	

V. Conclusion

This report outlines the development of a user-centric phishing URL detection tool, leveraging interpretable machine learning models. By addressing the gap in existing tools that often overlook user information needs, we utilized Recursive Feature Elimination (RFECV) technique for feature selection and LightGBM model due to its highest performance. The resulting web application received positive feedback from usability testing. In the future, efforts will focus on enhancing the user interface, integrating multilingual support, and continuously updating the detection model to counter evolving phishing techniques.

References

[1] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, “Fighting against phishing attacks: state of the art and future challenges,” Neural Computing and Applications, vol. 28, no. 12, 2017, pp. 3629–3654, doi: 10.1007/s00521-016-2275-y.

[2] A. V. Ramana, K. L. Rao, and R. S. Rao, “Stop-Phish: an intelligent phishing detection method using feature selection ensemble,” Social Network Analysis and Mining, vol. 11, no. 1, 2021, pp. 1-9, doi: 10.1007/s13278-021-00829-w.

[3] E. Büber, Ö. Demir and Ö. K Şahingöz, “Feature Selections for the Machine Learning Based Detection of Phishing Websites,” 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2017, pp.1-5, doi: 10.1109/idap.2017.809031.

[4] A. Safi and S. Singh, “A Systematic Literature Review on Phishing Website Detection Techniques,” J. of King Saud University: Computer and Information Sciences, vol. 35(2), 2023, pp. 590–611, doi: 10.1016/j.jksuci.2023.01.004.

[5] A. A.A. and P. K., "Towards the Detection of Phishing Attacks," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 337-343, doi: 10.1109/ICOEI48184.2020.9142967.

[6] I. Qabajeh, F. Thabtah, and F. Chiclana, “A recent review of conventional vs. automated cybersecurity anti-phishing techniques,” Computer Science Review, vol. 29, 2018, pp. 44–55, doi: 10.1016/j.cosrev.2018.05.003.

[7] O. Sarker, A. Jayatilaka, S. Haggag, C. Liu, and M. A. Babar, “A Multi-vocal Literature Review on challenges and critical success factors of phishing education, training and awareness,” Journal of Systems and Software, vol. 208, 2024, pp. 1-24, doi: 10.1016/j.jss.2023.111899.

[8] N. A. G. Arachchilage, S. Love, and K. Beznosov, “Phishing threat avoidance behaviour: An empirical investigation,” Computers in Human Behavior, vol. 60, 2016, pp. 185–197, doi: 10.1016/j.chb.2016.02.065.

[9] O. Sarker, S. Haggag, A. Jayatilaka and C. Liu, "Personalized Guidelines for Design, Implementation and Evaluation of Anti-Phishing Interventions," 2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), New Orleans, LA, USA, 2023, pp. 1-12, doi: 10.1109/ESEM56168.2023.10304861.

[10] L. Li, E. Berki, M. Helenius, and S. Ovaska, “Towards a contingency approach with whitelist-and blacklist-based anti-phishing applications: what do usability tests indicate?,” Behaviour & Information Technology, vol. 33, no. 11, 2014, pp. 1136–1147.

[11] S. Egelman, L. F. Cranor, and J. Hong, “You’ve been warned: an empirical study of the effectiveness of web browser phishing warnings,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2008, pp. 1065–1074.

[12] E. A. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad, and A. I. A. Alzahrani, “A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators,” Sensors, vol. 23, no. 9, 2023, pp. 1-27, doi: 10.3390/s23094403.

[13] A. Franz, V. Zimmermann, G. Albrecht, K. Hartwig, C. Reuter, A. Ben lian, and J. Vogt, “Sok: Still plenty of phish in the sea—a taxonomy of user-oriented phishing interventions and avenues for future research,” in Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021), 2021, pp. 339–358.

[14] P. Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M. A. Blair, and T. Pham, “School of phish: a real-world evaluation of anti phishing training,” in Proceedings of the 5th Symposium on Usable Privacy and Security, 2009, pp. 1–12.

[15] J. Stobbs, B. Issac and S. M. Jacob, "Phishing Web Page Detection Using Optimised Machine Learning," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 2020, pp. 483-490, doi: 10.1109/TrustCom50675.2020.00072.

[16] S. Shabudin, N. Samsiah, K. Akram, and M. Aliff, “Feature Selection for Phishing Website Classification,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 4, 2020, pp. 387-595, doi: 10.14569/ijacsa.2020.0110477.

[17] M. Zabihimayvan and D. Doran, "Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection," 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2019.8858884.

[18] R. B. Basnet and T. Doleck, "Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, India, 2015, pp. 220-223, doi: 10.1109/CICT.2015.63.

[19] M. Babagoli, M. P. Aghababa, and V. Solouk, “Heuristic nonlinear regression strategy for detecting phishing websites,” Soft Computing, vol. 23, no. 12, pp. 4315–4327, 2018, doi: 10.1007/s00500-018-3084-2.

[20] H. Zuhair, A. Selamat, and M. Salleh, “Feature selection for phishing detection: a review of research,” International Journal of Intelligent Systems Technologies and Applications, vol. 15, no. 2, pp. 147-162, 2016, doi: 10.1504/ijista.2016.076495.