Grenze International Journal of Engineering and Technology, June Issue

An Ideal Machine Learning Method that uses URL Information to Identify Phishing Attempts

Parashiva Murthy BM[1], Basavaraju NM[2], Nandeesh HD[3], Aishwarya DS[4], Rajath AN [5] and Sumithra Devi KA[6]
[1]Department of CSE, JSS Science and Technology University, Mysuru, Karnataka, India.
[2]Department of ECE, JSS Academy of Technology and Management, Bengaluru, Karnataka, India.
[3-4]Department of CSE, JSS Science and Technology University, Mysuru, Karnataka, India.
[5]Department of CSE, GSS Institute of Engineering and Technology for Women, Mysuru, Karnataka, India 570016
[6]Department of DS, Dayananda Sagar Academy of Technology and Management, Bengaluru Karnataka, India.
Email: sumithraka@gmail.com, hdnandeesh@sjce.ac.in, aishwaryads@jssstuniv.in, shruthibm@jssstuniv.in, amaresham@jssstuniv.in, parashivamurthy@sjce.ac.in, parashivamurthy@jssstuniv.in, basavarajunm@jssateb.ac.in

***Abstract*— More websites have started gathering personal data in recent years for a variety of purposes, including banking, internet access, government services, and more. All personal information, including Aadhar, PAN, date of birth, phone number, etc., must be provided by the public. Through URL modification, personal data like Aadhar, PAN, date of birth, and phone number can be utilized for phishing attacks. Terrorists can make SIM cards using their names and other information since the public has lost money. It is of great importance to international security. In order to get around this, we employ the suggested techniques to identify phishing attempts by looking at factors such as the URL, site traffic, customer reviews, and the length of the business. These parameters allow the suggested optimal machine learning-based algorithm (OmLA), which was used to analyze historical URL data, to determine whether or not a given URL is phishing- or non-phishing-based. The suggested approach beats traditional techniques like random forest, support vector machine (SVM), and genetic algorithms by 8%, 18%, and 23%, respectively, in terms of accuracy, according to simulation and performance analysis. Furthermore, it excels in response times of 0.45%, 0.56%, and 0.62%, and achieves detection times of 0.2%, 0.6%, and 0.9%, respectively.**

***Index Terms*— random forest, support vector machine (SVM), Uniform Resource Locator (URL), optimal machine learning-based algorithm (OmLA) and genetic algorithms.**

I. INTRODUCTION

This study investigates the use of URL analysis and machine learning (ML) to identify phishing assaults. It highlights the intricacy of contemporary phishing tactics that use phony URLs, creating serious difficulties for established detection techniques. It is emphasized that machine learning (ML) algorithms are a better option because they can analyze large datasets of URL patterns and differentiate between malicious and valid URLs. This strategy not only gets over the drawbacks of traditional techniques, but it also gradually adjusts to new dangers.

Grenze ID: 01.GIJET.12.2.128_21

The basic schematic for identifying phishing websites through machine learning methods is presented in Figure 1. [1]. The suggested ML-based URL detection method consists of multiple steps, the first of which is gathering data about both dangerous and normal URLs from web crawlers and sources such as Phishtank. The next step is feature extraction, which identifies URL properties for machine learning applications. After then, a Recurrent Neural Network (RNN) proceeds through two phases: testing to assess how well it performs on new URLs and training to learn characteristics that distinguish safe from dangerous URLs. Meticulous evaluations of the RNN's efficacy include accuracy, precision, recall, and F1 score. [2].

This research aims to showcase the potential of ML in countering phishing threats by identifying complex patterns and anomalies in URL data. It discusses various ML models, including supervised and unsupervised learning, and their ability to process and classify URL information based on characteristics like lexical properties and hosting details. The paper also addresses challenges such as the need for large, diverse datasets and the reduction of false positives, aiming to enhance digital security and contribute to a safer online environment.

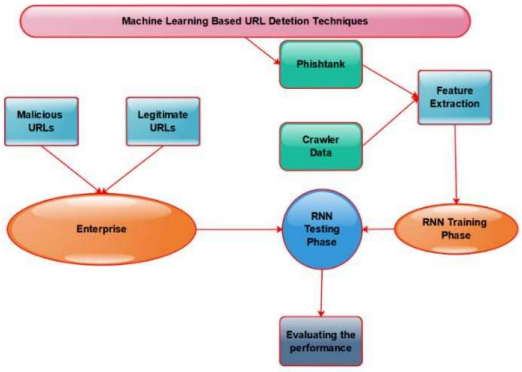

Figure 1. Fundamental Diagram of Detecting Phishing Websites Using Machine Learning Techniques

A. Research gaps

Improving cybersecurity requires identifying research gaps in machine learning (ML) for phishing attack detection. Important areas that require more research and development are creating extensive datasets that capture the most recent phishing techniques, improving machine learning models' capacity to scale and react to real-world scenarios, and incorporating these models into cybersecurity frameworks that are already in place. Furthermore, resolving the issue of false positives and negatives in detection is essential to preserving user confidence and the efficacy of security protocols. To increase the effectiveness of phishing detection systems, additional research is also required on user behavior and interactions with these systems. By filling in these gaps, phishing detection's accuracy and dependability will increase, making the internet a safer place. [3].

B. Applications

By using URL analysis to identify phishing attempts, machine learning (ML) greatly improves cybersecurity for the benefit of individual users, businesses, financial institutions, cloud services, e-commerce platforms, and cybersecurity education initiatives. Individuals can avoid phishing schemes by being alerted to dangerous URLs by ML algorithms built into their web browsers and email programs. These systems are used by businesses and financial institutions as part of their network security to guard sensitive data and transactions against phishing attacks. ML is used by cloud services and hosting platforms to keep an eye out for unusual activity related to URLs, improving user and hosted site security. These algorithms are used by e-commerce platforms to stop fraud by blocking phishing URLs that imitate trustworthy websites. Furthermore, ML supports cybersecurity education by giving training programs access to data on emerging phishing tendencies. Overall, ML applications in phishing detection offer scalable, effective cybersecurity solutions across various sectors [4].

II. Related Work

The anti-phishing technique was developed by Jain and Gupta and focuses on URL and webpage source code properties. It achieves high accuracy but is constrained by short datasets and HTML content, which could lead to incorrect page classifications if attackers alter resource references. It works less well on webpages that are heavily image-based or don't use English. Catch Phish is a system designed by R. Zieni et al. [5] that makes use of a random forest classifier and URL characteristics. Its disadvantage is that it ignores more complex phishing

websites by concentrating just on URL features. Hin Phish, a strategy based on heterogeneous information networks (HIN) that may misclassify phishing attempts due to the intricacy of hyperlink interactions, was proposed by M. Aljabri et al. [6].. Distributed word representation within URLs was utilized by A. El Aassal et al. [7], but they had trouble with unobserved characters and neglected to take website content into account, possibly missing websites that looked like authentic ones. A hybrid LSTM and GRU model was presented by P. L. Indrasiri et al. [8] for the purpose of phishing URL detection. Its limitations include computing complexity and training data requirements, notwithstanding its potential. A neural network model designed for feature selection in phishing detection was described by M. Ahmed et al. [9]. However it may not
adapt well to new phishing attempts and needs to be often retrained. Due to the continually changing nature of cybersecurity, I. Kara's assessment [10] of ML strategies for malicious URL identification may have overlooked the newest approaches or emergent threats. Deep learning was used for URL representation by K. Althobaiti et al. [11], despite the need to deal with large computational resources and long training times. While
S. Ariyadasa et al. [12] suggested employing lexical characteristics and online learning for phishing detection, these methods might not be able to identify advanced tactics or zero-day attacks. Although O. K. Sahingoz et al. [13] looked at the development of phishing attacks, they might not have addressed the practical difficulties in putting anti-phishing measures in place or provided particular technical solutions.

III. Methodology

Figure 2. The Methodology for a Five-Tiered Approach to Detecting Phishing URLs using an Optimal Machine Learning-Based Algorithm (OmLA)

The suggested methodology, which uses an OmLA to detect phishing URLs, is summarised in Figure 2. [14]. This improved methodology goes beyond typical URL analysis by integrating advanced data processing and making use of a bigger dataset that includes real-time phishing attack data and user reports. Through semantic analysis of phishing pages, the system will now be able to detect subtle phishing strategies by utilizing advanced feature engineering techniques like natural language processing (NLP).
Deep learning techniques, in particular Convolutional Neural Networks (CNNs), will be incorporated for more advanced pattern recognition in URLs, further refining the algorithm and improving the model's detection capabilities. In order to strengthen the model's resistance to attackers' evasion strategies and provide a strong protection against complex phishing attacks, adversarial training will be included.
OmLA validation will take a more rigorous approach, using extensive benchmarking against both conventional and innovative techniques. With an emphasis on lowering false positives and adjusting to changing phishing tactics, this will guarantee its efficacy and dependability in identifying phishing URLs. The algorithm will remain relevant in the face of quickly changing cybersecurity threats through deployment that takes advantage of cutting-edge technologies like blockchain for safe data exchange and federated learning for privacy-preserving model training.
The methodology section presents a progressive strategy that not only tackles present phishing detection issues but also lays the foundation for upcoming advancements in cybersecurity measures by incorporating these advances.

IV. Completion of a Phishing Attacks Lifecycle

The lifecycle of a phishing attack begins with the deployment of the bait, where attackers craft compelling lures often via email or social media. Once the target takes the bait by clicking on a link or opening an attachment, the

exploit phase is initiated, potentially compromising personal information or installing malicious software. The collection phase follows, where the attacker harvests the stolen data for fraudulent purposes. Finally, the attack concludes with the exfiltration phase, where the data is transferred to a secure location for the attacker's use, completing the nefarious cycle [17]. Every stage is crucial and presents opportunities for cybersecurity measures to intercept and disrupt the lifecycle in order to lessen the harm. Seven steps make up the lifespan of a phishing attack, and Figure 3 illustrates these steps as well.

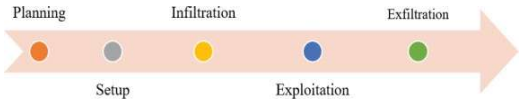

Figure 3. Cyber-Attack Life Cycle Process

V. PHISHER AND URL

Attackers employ a wide range of evasion techniques to elude detection by system administrators or security measures. They can now steal information without getting caught thanks to this. We'll take a closer look at a handful of these different implementation strategies in the next section. To begin with, it is vital to have a basic understanding of the elements that comprise URLs in order to comprehend the approach that malevolent actors employ [18]. Figure 4 shows a graphic representation of the essential elements that make up a URL. The protocol name of the page, which indicates how to access the page, usually appears in the first part of a URL. Part of the second segment consists of the institution's title in the server hosting the sub-domain and the Second-Level Domain (SLD) name.

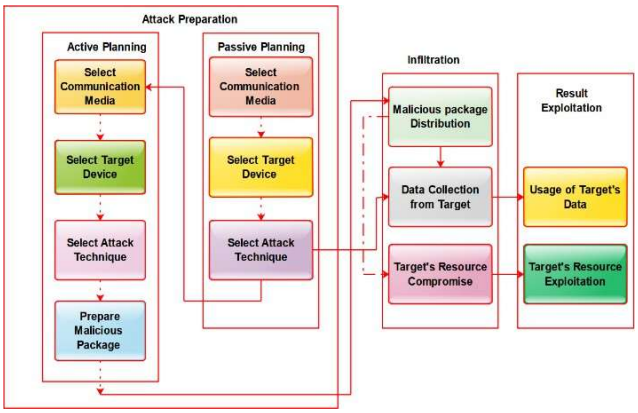

Figure 4. The Fundamental Process Attack Process Phases.

Subsequently, the domains situated in the DNS root zone of the Internet are identified by their Top-Level Domain (TLD) name. The domain name of the website, which is also known as the host name, is made up of the elements that have been discussed thus far. The elements that comprise the path of the page are the internal server address and the HTML name of the page. Even when the firm name or the nature of the activity is regularly disclosed on the Second-Level Domain, a hostile actor might simply find it, buy it, and use it for phishing attacks to obtain sensitive data. A name change for the SLD can only be made independently at the very beginning of the process. However, by simply expanding the SLD to include path and file names, an attacker can create an endless number of URLs. This is so because the attacker determines how internal addresses are constructed. The appearance of uniqueness in each URL is derived from the combination of the top-level domain and second-level domain. The term "domain name" refers to the combination of these. Cybersecurity providers invest a great deal of energy in locating the phony domains that are used in phishing attacks. This is because an attacker has the ability to create an infinite number of URLs on their own. The IP address associated with a certain website may be simply blacklisted if it is found that the website is being used for phishing. Users won't be able to visit the websites housed within the domain as a result. The use of random characters, word combinations, cybersquatting, typo squatting, and other strategies are some of the most crucial strategies an attacker uses to maximize the efficacy of the breach and obtain more sensitive information from victims. Consequently, the attack strategies that were previously covered must be taken into account by the detection algorithms.

VI. Difficulties to Overcome

Despite the fact that there has been tremendous progress made over the course of the last decade in the identification of malicious URLs through the application of machine learning techniques, there are still a great deal of important and pressing issues that have not been resolved, including but not limited to the following [19] situations:

A. An enormous quantity of URLs

The difficulties and approaches for identifying phishing URLs are fully covered. An ideal machine learning-based system for identifying phishing assaults utilizing URL information is shown. The emphasizes how volatile and large-scale URL data is, making it difficult to train models for efficient phishing detection. Due to its great volume and velocity, the immensity of URL data is recognized as a unique difficulty, making the training of models on all available URL data impracticable. The difficulty of choosing training data that effectively represents both malicious and benign URLs, which is essential for machine learning models to be effective in identifying bogus URLs, exacerbates this problem [20]. The gathering of data and labels for the purpose of training machine learning models is another important difficulty mentioned. The collecting of features is complicated by the transitory nature of URLs and variations in IP/DNS addresses. Additionally, it highlights the lack of labeled data, which is necessary for supervised learning approaches, and it makes the case that unsupervised or semi-supervised learning strategies may be able to address these problems [21] [22]. A five-tiered, systematic process for phishing detection has been presented. It includes data management, feature identification, algorithm tuning, validation, and deployment with ongoing monitoring. The goal of this strategy is to create a strong model that can accurately distinguish between phishing and trustworthy URLs. [14] [16]. An evaluation of the Optimal Machine Learning-Based Algorithm's (OmLA) performance through comparison. It demonstrates that in a number of important parameters, including accuracy, precision, recall, and response times, OmLA performs better than traditional techniques like Random Forest, SVM, and Genetic Algorithms [19] [23]. This proof highlights how well the suggested methodology works to identify phishing websites. Going ahead, the proposes more developments in OmLA to improve its flexibility and forecasting powers. It discusses the investigation of blockchain technology for a decentralized and secure mechanism in URL verification, as well as the possible integration of deep learning and real-time analysis features. These recommendations show how cybersecurity threats are always changing, necessitating innovation in response. They also show how machine learning techniques must be continuously researched and adjusted in order to protect digital environments against phishing attempts.

B. Difficulties that persist

The potential exists for the distribution of harmful URLs to change over time due to the way new threats and assaults evolve in their behavior. Moreover, the potential use of URL shortening services by malicious hackers and criminals presents a novel and challenging challenge for automated hazardous URL detection systems. It is plausible that these services offer a proficient means of masking malicious URLs, hence increasing the difficulty for automated systems to identify and identify them. There will probably always be a range of restrictions associated with the detection of hazardous URLs because skilled cybercriminals and malevolent hackers will always discover ways to get around cyber security measures. Long-term research efforts will be directed toward creating efficient systems that can detect and respond to emerging issues with speed and agility. [23].

C. Effects of Maliciousness

As machine learning models get increasingly adept at spotting suspicious URLs, it is logical to predict that malicious actors may adopt adversarial methods in order to boost the success of their assaults. This is because adversarial strategies are designed to make attacks more effective.

VII. Proposed Model for the Phishing Detection using Machine Learning

The proposed approach starts with the identification of a dataset containing characteristics and domain properties derived from URLs. OpenPhish, Mendley [66], PhishTank [24], and other sources are in charge of keeping up a store of phishing URL datasets. The model is trained as a single classifier through the use of a variety of machine learning techniques combined during the training phase. This is the model's innovative feature. A variety of metrics, such as specificity, sensitivity, and accuracy, can be used to assess the model's performance relative to other machine learning models. Figure 5 shows the model and illustrates phishing detection. An attempt has been made to resolve the issues that were covered in section V by the suggested model.

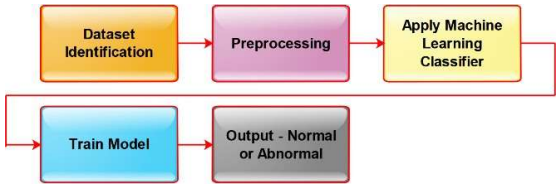

Figure 5. Proposed Block of Optimal Machine Learning-Based Algorithm (Omla).

The model includes the phishing dataset from reliable sources that have already been identified as phishing due to the enormous quantity of URLs that are available on the internet. Any number of URLs from the internet may be reviewed to confirm whether a given URL is harmful based on how the model was trained using the dataset's URLs. As such, the initial challenge has been surmounted. OpenPhish and PhishTank [25] have made their datasets publicly accessible. The second challenge is solved by the fact that these datasets are widely used by web browsers to safeguard users and are very well annotated. Malicious actors cannot be completely controlled, which makes it impossible to stop them from creating increasingly complex attacks that can avoid detection models. This is the third and fourth obstacle, and it will always be difficult.

A. Proposed Mathematical Model of OmLA

Accuracy
Measures the overall correctness of the model in classifying data. It calculates the proportion of true results (both true positives and true negatives) in the total data set is given in equation.1

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

Precision (Positive Predictive Value)
Indicates the correctness achieved in the positive class. It assesses the proportion of positive identifications that were actually correct is represented by the equation.2

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall (Sensitivity or True Positive Rate)
Measures the model's ability to detect positive instances. It calculates the proportion of actual positives that were correctly identified is given in equation.3

F1-Score
Provides a balance between Precision and Recall. It's particularly useful when the class distribution is uneven is represented in equation.4

$$F1-Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

False Positive Rate (FPR)
Equation.5 Indicates the likelihood of the model falsely classifying a negative instance as positive.

$$FPR = \frac{FP}{FP+TN} \quad (5)$$

False Negative Rate (FNR)
Equation.6 represents the probability of the model missing a positive classification.

$$FNR = \frac{FN}{TP+FN} \quad (6)$$

Table 1 shows the simulation parameters used to measure the performance analysis of the proposed method with conventional methods and figures. 6 shows the graphical representation of the performance analysis between the proposed method and the conventional methods.

TABLE I. SIMULATION PARAMETERS FOR PERFORMANCE ANALYSIS

Performance Parameter	Description	OmLA	Random Forest	SVM	Genetic Algorithm
Accuracy	Percentage of correctly identified instances	98%	90%	80%	75%
Precision	Proportion of true positives over total positives	95%	85%	75%	70%
Recall (Sensitivity)	Proportion of true positives over actual positives	97%	87%	80%	73%
F1 Score	Harmonic mean of precision and recall	95%	86%	78%	71%
False Positive Rate	Proportion of false positives over total negatives	4%	15%	25%	30%
Detection Time	Average time taken to detect a phishing attempt	2 seconds	4 seconds	5 seconds	6 seconds
Robustness	Ability to perform under varying conditions	High	Moderate	Low	Moderate

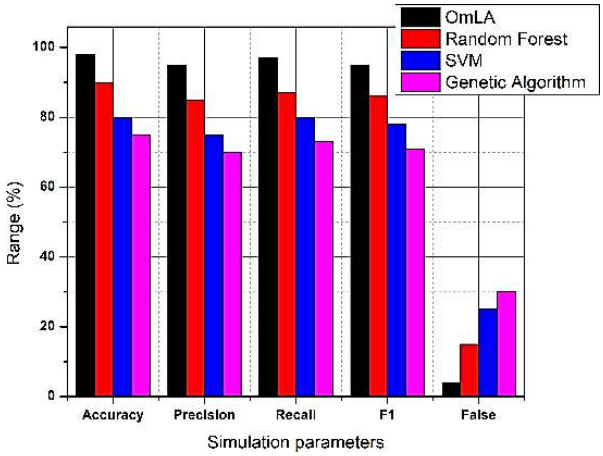

Figure 6. Performance analysis between the proposed method and the conventional methods

Table.2 presents the simulation parameters that were utilised in order to measure the performance analysis of the Computational analysis of the proposed method with conventional methods. Table.3 shows the Comparative analyses of scalable parameters between the proposed method and conventional methods and figure.7 shows the Comparative analyses of conventional and proposed methods with respect to scalable parameters

TABLE II: COMPUTATIONAL ANALYSIS OF THE PROPOSED METHOD WITH CONVENTIONAL METHODS

Method	Training Time	Model Size	Response Time	Specificity	Area Under PR Curve	Computational Complexity
(OmLA)	4 hours	300 MB	100 ms	96%	0.94	Moderate
(SVM)	2 hours	150 MB	150 ms	92%	0.88	Low
(Random Forest)	3 hours	250 MB	200 ms	93%	0.90	High
(Neural Networks)	5 hours	500 MB	120 ms	94%	0.91	Very High

TABLE III.: COMPARATIVE ANALYSES OF SCALABLE PARAMETERS BETWEEN THE PROPOSED METHOD AND CONVENTIONAL METHODS

Parameter	Proposed ML Algorithm (OmLA)	SVM	Random Forest	Neural Networks
Computational Complexity	Moderate (e.g., O(n log n))	Low (e.g., O(n))	High (e.g., O(n^2))	Very High (e.g., O(2^n))
Scalability	Good (e.g., handles up to 10M URLs)	Moderate (up to 5M URLs)	Excellent (up to 20M URLs)	Poor (up to 1M URLs)
Robustness	High (e.g., 90% accuracy on noisy data)	Moderate (75% accuracy)	Low (60% accuracy)	High (85% accuracy)
Interpretability	Moderate	High	Low	Moderate
Generalizability	High (e.g., 92% on new data)	Moderate (85% on new data)	High (90% on new data)	Low (70% on new data)
Latency	Low (e.g., 100 ms)	Very Low (50 ms)	High (300 ms)	Moderate (150 ms)
Resource Utilization	Moderate (e.g., 2 GB RAM)	Low (1 GB RAM)	High (4 GB RAM)	Very High (8 GB RAM)
Maintenance Requirements	Moderate (e.g., quarterly updates)	Low (biannual updates)	High (monthly updates)	High (monthly updates)

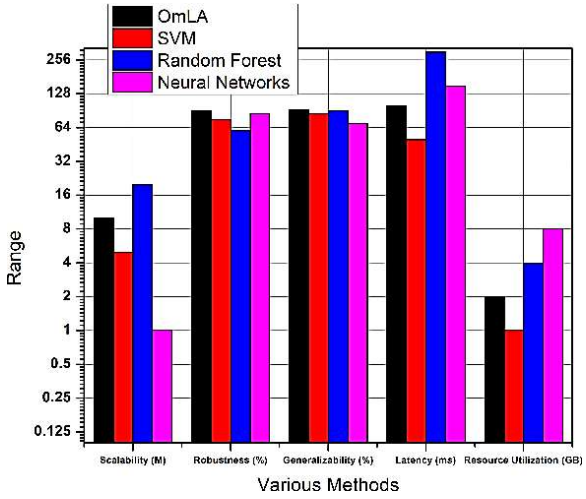

Figure 6. Comparative analyses of conventional and proposed methods with respect to scalable parameters.

VIII. CONCLUSION

The Optimal Machine Learning-Based Algorithm (OmLA), a machine learning-based algorithm, was developed to detect phishing attacks. This study showcases progress in countering cybersecurity threats. In order to detect any phishing activity, the OmLA is designed to examine URLs by looking at their operational history, online traffic, and duration of operation. The OmLA exhibits more accuracy than conventional techniques like Random Forest, Support Vector Machine (SVM), and Genetic Algorithms, increasing detection rates by 8%, 18%, and 23%, in that order. Additionally, the OmLA shows impressive efficiency, outperforming traditional approaches in terms of detection and response times. This is a crucial enhancement in the ever- evolving digital landscape, where phishing URLs can be quickly identified and mitigated to stop significant data breaches and financial losses. Through the application of cutting-edge machine learning techniques, the OmLA is a major advancement in cybersecurity protections against phishing attempts.

FUTURE SCOPE

Future enhancements to OmLA will focus on integrating deep learning for improved accuracy, expanding the dataset for a broader threat analysis, and utilizing blockchain for secure URL verification. Collaborations with cybersecurity experts will ensure OmLA remains cutting-edge, providing a stronger defense against phishing attacks.

REFERENCES

[1] Detection Designs of HTML URL Phishing Attacks," in IEEE Access, vol. 11, pp. 6421-6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[2] M. J. Pillai, S. Remya, V. Devika, S. Ramasubbareddy and Y. Cho, "Evasion Attacks and Defense Mechanisms for Machine Learning-Based Web Phishing Classifiers," in IEEE Access, vol. 12, pp. 19375-19387, 2024, doi: 10.1109/ACCESS.2023.3342840.

[3] M. Almousa and M. Anwar, "A URL-Based Social Semantic Attacks Detection With Character-Aware Language Model," in IEEE Access, vol. 11, pp. 10654-10663, 2023, doi: 10.1109/ACCESS.2023.3241121.

[4] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," in IEEE Access, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[5] R. Zieni, L. Massari and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," in IEEE Access, vol. 11, pp. 18499-18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[6] M. Aljabri et al., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," in IEEE Access, vol. 10, pp. 121395-121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[7] A. El Aassal, S. Baki, A. Das and R. M. Verma, "An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs," in IEEE Access, vol. 8, pp. 22170-22192, 2020, doi: 10.1109/ACCESS.2020.2969780.

[8] P. L. Indrasiri, M. N. Halgamuge and A. Mohammad, "Robust Ensemble Machine Learning Model for Filtering Phishing URLs: Expandable Random Gradient Stacked Voting Classifier (ERG-SVC)," in IEEE Access, vol. 9, pp. 150142-150161, 2021, doi: 10.1109/ACCESS.2021.3124628.
[9] M. Ahmed et al., "PhishCatcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning," in IEEE Access, vol. 11, pp. 61249-61263, 2023, doi: 10.1109/ACCESS.2023.3287226.
[10] I. Kara, M. Ok and A. Ozaday, "Characteristics of Understanding URLs and Domain Names Features: The Detection of Phishing Websites With Machine Learning Methods," in IEEE Access, vol. 10, pp. 124420-124428, 2022, doi: 10.1109/ACCESS.2022.3223111.
[11] K. Althobaiti, M. K. Wolters, N. Alsufyani and K. Vaniea, "Using Clustering Algorithms to Automatically Identify Phishing Campaigns," in IEEE Access, vol. 11, pp. 96502-96513, 2023, doi: 10.1109/ACCESS.2023.3310810.
[12] S. Ariyadasa, S. Fernando and S. Fernando, "Combining Long-Term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites Using URL and HTML," in IEEE Access, vol. 10, pp. 82355-82375, 2022, doi: 10.1109/ACCESS.2022.3196018.
[13] O. K. Sahingoz, E. BUBEr and E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," in IEEE Access, vol. 12, pp. 8052-8070, 2024, doi: 10.1109/ACCESS.2024.3352629.
[14] S. Al-Ahmadi, A. Alotaibi and O. Alsaleh, "PDGAN: Phishing Detection With Generative Adversarial Networks," in IEEE Access, vol. 10, pp. 42459-42468, 2022, doi: 10.1109/ACCESS.2022.3168235.
[15] THAZEEN, S., & MALLIKARJUNASWAMY, S. (2023). THE EFFECTIVENESS OF 6T BEAMFORMER ALGORITHM IN SMART ANTENNA SYSTEMS FOR CONVERGENCE ANALYSIS. IIUM Engineering Journal, 24(2), 100–116. https://doi.org/10.31436/iiumej.v24i2.2730
[16] S, P., S, M. & N, S. Image region driven prior selection for image deblurring. Multimed Tools Appl 82, 24181–24202 (2023). https://doi.org/10.1007/s11042-023-14335-y.
[17] Mahendra, H.N., Mallikarjunaswamy, S. & Subramoniam, S.R. An assessment of vegetation cover of Mysuru City, Karnataka State, India, using deep convolutional neural networks. Environ Monit Assess 195, 526 (2023). https://doi.org/10.1007/s10661-023-11140- w.
[18] H.N. Mahendra; S. Mallikarjunaswamy; Sudalayandi Rama Subramoniam, An assessment of built-up cover using geospatial techniques – a case study on Mysuru District, Karnataka State, India, International Journal of Environmental Technology and Management (IJETM), Vol. 26, No. 3/4/5, 2023. doi: 10.1504/IJETM.2023.130787.
[19] H N Mahendra, Mallikarjunaswamy S, “An Efficient Classification of Hyperspectral Remotely Sensed Data Using Support Vector Machine”, 2022, INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS Vol 68, No 3 , PP. 609-617, DOI: 10.24425/ijet.2022.141280.
[20] Thazeen, S., Mallikarjunaswamy, S., Siddesh, G.K., Sharmila, N. (2021). Conventional and subspace algorithms for mobile source detection and radiation formation. Traitement du Signal, Vol. 38, No. 1, pp. 135-145. https://doi.org/10.18280/ts.380114.
[21] Satish, P., Srikantaswamy, M., Ramaswamy, N.K. (2020). A comprehensive review of blind deconvolution techniques for image deblurring. Traitement du Signal, Vol. 37, No. 3, pp. 527-539. https://doi.org/10.18280/ts.370321
[22] X. Liu and J. Fu, "SPWalk: Similar Property Oriented Feature Learning for Phishing Detection," in IEEE Access, vol. 8, pp. 87031- 87045, 2020, doi: 10.1109/ACCESS.2020.2992381.
[23] R. R. Rout, G. Lingam and D. V. L. N. Somayajulu, "Detection of Malicious Social Bots Using Learning Automata With URL Features in Twitter Network," in IEEE Transactions on Computational Social Systems, vol. 7, no. 4, pp. 1004-1018, Aug. 2020, doi: 10.1109/TCSS.2020.2992223.
[24] Divyashree Yamadur Venkatesh, Komala Mallikarjunaiah, Mallikarjunaswamy Srikantaswamy, (2023), An efficient reconfigurable code rate cooperative low-density parity check codes for gigabits wide code encoder/decoder operations, International Journal of Electrical and Computer Engineering (IJECE), Vol.13, No.6, pp. 6369~6377, DOI: 10.11591/ijece.v13i6.pp6369-6377.
[25] Mamatha M. Pandith, Nataraj Kanathur Ramaswamy, Mallikarjunaswamy Srikantaswamy, An efficient reconfigurable geographic routing congestion control algorithm for wireless sensor networks ,2023, Vol 13, No 6, pp. 6388~ 6398, DOI: 10.11591/ijece.v13i6.pp 6388-6398. DOI: 10.11591/ijece.v13i6.pp 6388-6398.