ML-based Real-Time URL Inspection with Hardware Acceleration for Enhanced Web Security

Majid Nezarat
School of Electrical Engineering
Iran University of Science and Technology
Tehran, Iran
majid_nezarat@elec.iust.ac.ir

Erfan Khedersolh
School of Electrical Engineering
Iran University of Science and Technology
Tehran, Iran
erfan_khedersollh@alumni.iust.ac.ir

Hadi Shahriar Shahhosseini
School of Electrical Engineering
Iran University of Science and Technology
Tehran, Iran
hshsh@iust.ac.ir

Amin Rezaei
Department of Com. Engineering & Com. Science
California State University Long Beach
Long Beach, CA, USA
amin.rezaei@csulb.edu

***Abstract*—With the proliferation of digital technology, a significant number of users leverage edge and IoT devices for web exploration. However, the inherent resource constraints of these systems often preclude the deployment of computationally intensive security mechanisms. Consequently, edge and IoT devices have become prime targets for cybercriminals. A critical issue arises when these devices encounter compromised websites capable of executing various cyberattacks, such as phishing, malvertising, and clickjacking. This paper proposes a machine learning-based approach for real-time URL inspection to provide immediate feedback on the legitimacy of accessed websites. We extract and analyze URL features to classify websites into categories such as benign, defacement, phishing, malware, and spam. We have developed a hardware-efficient neural network to perform this classification in real-time. Our solution can be integrated either as an extension to browser code or as a hardware module, thereby enhancing the security of edge and IoT devices during web browsing. The simulation results show an accuracy of nearly 98%, which is very favorable considering the real-time nature of the proposed model.**

***Index Terms*—Cybersecurity, Hardware Accelerator, Classification, Malicious Sites, Cyberattack, Real-Time Computing**

I. Introduction

The proliferation of Internet technology across sectors such as banking, education, social networks, and e-commerce has significantly enhanced human convenience [1], [2]. However, cybercriminals are trying to endanger the security of Internet users with their criminal activities in their pursuit of profit [3]–[6]. Malware, one of the most dangerous cyberattacks, is capable of causing millions of dollars of damage to important infrastructures in the world every year [7]. Phishing as another type of cybercrime that happens when the attacker tries to convince the victim to disclose their personal information [8]–[11]. In defacement attack, criminals try to display their desired messages, which are usually political, religious, etc., after hacking the website [12]. Also, in recent years, the production of spam has led to the dissatisfaction of many users of web search engines [13].

Most of the mentioned cyberattacks rely on illegitimate or fake websites to mislead the user and steal their sensitive information. Hence, the task of classifying benign websites versus infected or malicious websites has become an important issue. By means of early prediction and blocking users access to malicious websites, their privacy and security can be taken care of. The information extracted from the Internet addresses of websites (i.e., URLs) gives useful knowledge about the status of the websites. Machine Learning (ML) techniques can be helpful as an important monitoring tool in the field of cybersecurity and Internet of Things (IoTs) [14], [15]. Recent studies have revealed that IoT devices are particularly susceptible to new security threats, highlighting the urgent need for lightweight and low-power security solutions [16]–[18]. While ML techniques can be used to look into the URLs to predict malicious websites, they must not slow the web browsing experience down. For example, despite the wide application, edge and IoT devices are not able to inspect URLs in real-time due to the limited resources they have. In state-of-the-art studies, URL classifiers have been designed and presented without paying attention to hardware implementation [4], [19], [20]. In addition, their hardware implementation in real-time consumes a lot of hardware resources.

In this paper, the information obtained from the URLs is used to design an accurate classification for legitimate and malicious sites. Considering the browsing pace as a main factor from user experience, an optimal Multi-Layer Perceptron Neural Network (MLPNN) classifier is trained and implemented on FPGA that can perform classification in real-time. The contributions of this paper are as follows:

- Designing and training a lightweight MLPNN that reduces computation costs while maintaining high accuracy in classifying URLs as benign or malicious.
- Implementing the proposed method on a Zynq Ultra-

Scale+ MPSoC, demonstrating real-time URL inspection capabilities with minimal hardware resource consumption and low power usage.
- Showcasing the proposed model's superior decision metrics compared to existing methods, along with hardware resource utilization and timing parameters, to show its effectiveness for real-time web security applications.

The rest of the paper is organized as follows: Section II reviews related works and research gaps. In Section III, the proposed method, which includes the introduction of the dataset, data pre-processing, feature dimensionality reduction, classifier design, and implementation of the proposed method on FPGA, is presented. In Section IV, results and analysis are discussed. Finally, Section V concludes this paper.

II. Related Works

In recent years, researchers have attempted to provide new methods to protect Internet users. One of these solutions is to create black-and-white lists that are used by web browsing providers. Due to the weak performance of black-and-white list methods against zero-day attacks, ML-based prediction of malicious sites can have a great impact on protecting users.

The Transfer-based Model for Malicious URL Classification (TMMUC) [19] introduces an unsupervised transformer model for detecting phishing attempts. While this approach has demonstrated high accuracy, its computational demands may hinder real-time application.

The Hybrid ML Model (HMLM) [4] introduces a combination of Support Vector Machine (SVM), Decision Tree (DT), and Linear Regression (LR), with results determined by a voting system. Although this hybrid approach improves classification accuracy, it adds computational, power, and hardware overhead.

Additionally, the Phishing Detection method based on Deep Learning (PDDL) [20] employs a Bidirectional Long Short-Term Memory (BLSTM) model to predict phishing sites using URLs. However, the BLSTM's implementation and real-time operation demand significant hardware resources, leading to increased power consumption.

The Detection of Online Phishing Email (DOPE) [21] method utilizes Reinforcement Learning (RL) for phishing detection, extracting features from email headers, HTML, URLs, and email text. However, the small dataset used in DOPE limits the accuracy of its results.

Also, the Phishing Website Detection using ML (PWDML) [22] proposes a phishing attack detection system utilizing Random Forest (RF). While PWDML has achieved high accuracy and low computational overhead, the limited dataset and small number of features pose challenges to its accuracy and performance.

While according to the European Union Agency for Cybersecurity (ENISA), phishing is the most common cyberattack, other types of cyberattacks are equally significant [23]. In order to close the research gap in this area, we introduce an accelerated hardware solution capable of real-time classification of benign, phishing, malware, defacement, and spam sites.

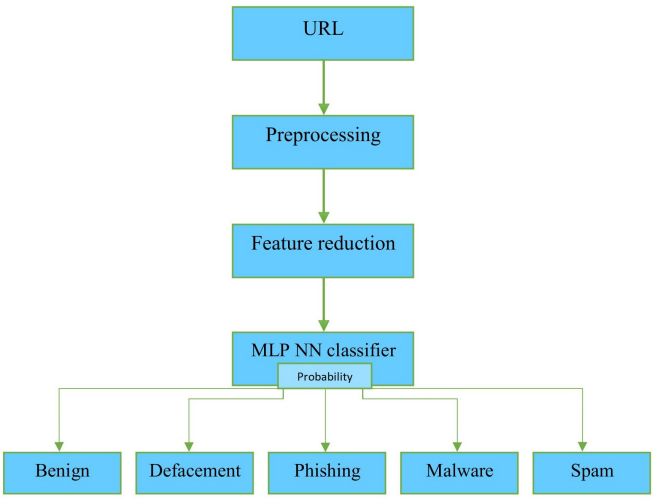

Fig. 1. Flowchart of **MERCEDES**.

III. MERCEDES: Real-Time URL Inspection

In this section, we introduce the dataset, data pre-processing, feature reduction, MLPNN classifier and hardware implementation of **MERCEDES**, an ML-based rEal-time URL inspeCtion for EnhanceD wEb Security.

Fig. 1 illustrates the flowchart of **MERCEDES**, the proposed model for detecting malicious sites. This approach employs an MLPNN with a single hidden layer, optimized to classify malicious and benign sites. The MLPNN architecture is designed to be in its most optimal state, balancing high accuracy with low computational overhead. The primary goal of this design is to achieve real-time prediction and classification with high accuracy.

A. Dataset

URLs consist of various components that provide valuable information about websites. Fig. 2 illustrates a URL address: Label 1 indicates the Hypertext Transfer Protocol (HTTP), Label 2 shows the subdomain, Label 3 is divided into the domain name and domain extension, and Label 4 represents the directory on the web server. Each of these elements contains extractable information.

In this paper, the ISCX-URL2016 dataset [24], compiled by the Canadian Institute for Cybersecurity (CIC), is utilized to classify legal and illegal websites. This dataset includes 36,707 samples categorized into five classes: Defacement, Benign, Phishing, Malware, and Spam. The number of samples per class is 7,930, 7,781, 7,586, 6,712, and 6,698, respectively.

Fig. 2. URL Segmentation.

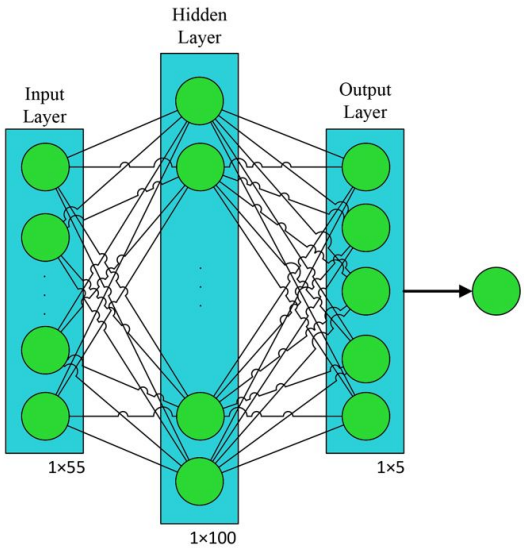

Fig. 3. Proposed MLPNN Structure.

Each sample has 80 features extracted from the URL, resulting in a dataset with 36,707 rows and 80 columns.

B. Data Pre-Processing

For data pre-processing, all columns are initially examined, revealing that nine columns contain missing values and one column has infinite values. The missing values are replaced with the average of their respective columns, and the column with infinite values is removed. Additionally, the column containing class labels is separated from the dataset. Next, feature values are normalized. Equation 1 illustrates the normalization method, where X represents the feature value. The feature values, ranging from -1 to 1424, are mapped to a range of 0 to 1 through normalization.

$$F_{Normalized} = \frac{F - F_{Min}}{F_{Max} - F_{Min}} \tag{1}$$

C. Features Reduction

To detect malicious sites, our primary goal is to speed-up the classifier. Long delays caused by intensive calculations can make web browsing tedious and lead to user dissatisfaction. One way to reduce computational complexity is by decreasing the number of features while still retaining the crucial information. We employ the Principal Component Analysis (PCA) algorithm to reduce the dimensions of the feature space. Initially, there are 79 features, which PCA reduces to 55 while retaining 99% of the information. This approach not only improves performance accuracy but also reduces computational overhead, directly impacting execution time.

D. MLPNN Classifier

MLPNN is a powerful tool for classification applications, offering high accuracy, low computational overhead, and flexibility. Reducing computational overhead also decreases the need for hardware resources. Given these advantages, we believe the MLPNN classifier is well-suited for a malicious detection system. Additionally, reducing computational overhead lowers power consumption and the silicon wafer area required in an Application-Specific Integrated Circuit (ASIC) platform, while also increasing operational speed.

The MLPNN structure comprises an input layer, one or more hidden layers, and an output layer. Fig. 3 illustrates the proposed MLPNN classifier structure. The feature matrix, processed by the PCA algorithm, is fed into the MLPNN, where calculations are performed through the layers, and the classifier outputs the predicted class number. It is important to note that fewer hidden layers result in lower computational overhead and higher execution speed. Table I details the MLPNN structure parameters.

E. Hardware Implementation

For hardware implementation, ASIC is the most optimal platform, but ASIC design is time-consuming and not cost-effective to manufacture. CPU and GPU are general-purpose and cannot meet expectations well in real-time systems. FPGA is cheaper than ASIC and also outperforms CPU and GPU in real-time systems with parallel processing capability.

In this paper, we implement **MERCEDES** on the Zynq UltraScale+ MPSoC with XCZU9EG-FFVB1156-2-E part number. XCZU9EG-FFVB1156-2-E consists of two parts: Programmable Logic (PL) and Processing System (PS) which contains Quad ARM® Cortex®-A53 MPCore. For hardware implementation, the malicious detection system is modeled using mathematical relations of matrix addition and multiplication. Fig. 4 shows the flowchart of the proposed hardware implementation based on FPGA. It should also be noted that to reduce resource consumption on the chip, the matrix calculations are stream-based. To optimize the use of hardware resources, matrix calculations are performed in each step in advance, and the bit length is adjusted according to the largest number obtained.

IV. Results and Analysis

In this section, we present the evaluation of **MERCEDES**. First, samples are prepared to be used for training and testing of the model. Specifically, 70% of the dataset is allocated for training, and the remaining 30% is used for testing. Also, to achieve better accuracy in evaluation and to prevent the model from overfitting, the training samples are shuffled before training the network. Design, train, and test of the proposed

TABLE I
MLPNN Parameters.

Model	Sequential
Hidden Layer (Type, Dimension)	Dense, [1,100]
Output Layer (Type, Dimension)	Dense_1, [1,5]
Trainable Parameters (Number, Size)	6105, 23.85 KB
Not Trainable Parameters (Number, Size)	0, 0 KB

TABLE II
PERFORMANCE EVALUATION PARAMETERS.

	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)
MERCEDES	97.68	97.08	98.17	97.62
HMLM [4]	98.12	96.33	97.31	95.89
BLSTM [20]	95.47	95.37	95.60	95.67
TMMUC [19]	98.55	98.57	98.55	98.56

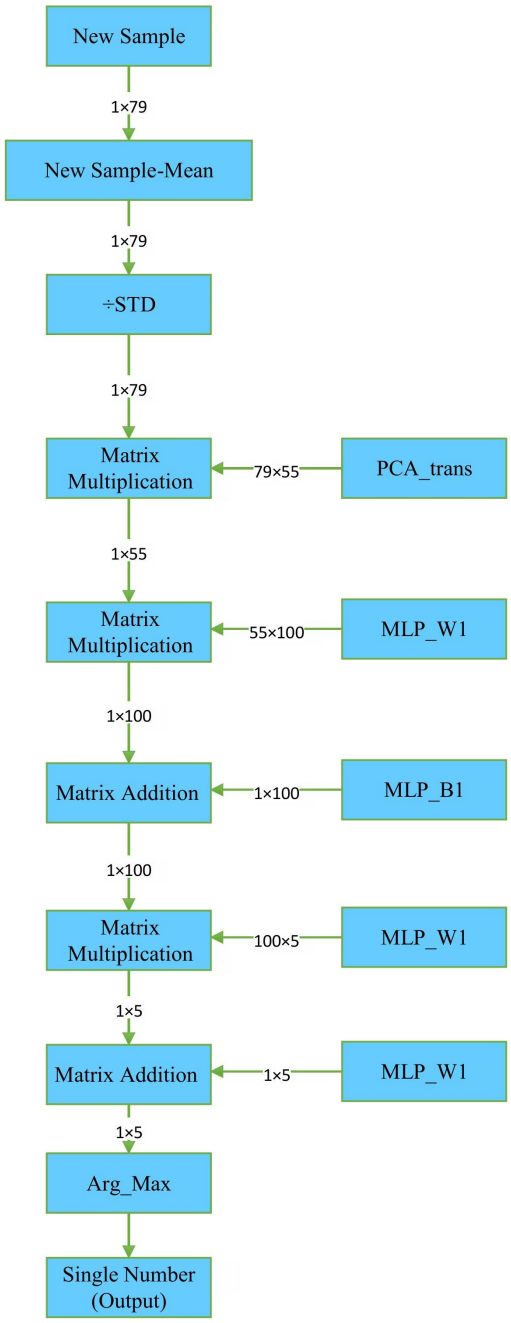

Fig. 4. Flowchart of **MERCEDES** Hardware Implementation.

classifier have been done by TensorFlow library with Python, and the hardware implementation has been done with the High-Level Synthesis (HLS) tool by the Vivado simulator.

A. Performance Metrics

Table II shows the accuracy, recall, precision, and F1 score of **MERCEDES** compared with state-of-the-art works in phishing detection.

As can be seen, the accuracy of **MERCEDES** is 97.68%, which is higher than the BLSTM [20] model's 95.47%. This indicates that **MERCEDES** is more reliable in correctly identifying both benign and malicious URLs. With a recall of 97.08%, **MERCEDES** demonstrates a high ability to correctly identify malicious URLs. This is crucial for security applications where missing a malicious URL can have severe consequences. In addition, the precision of 98.17% shows that when **MERCEDES** identifies a URL as malicious, it is highly likely to be correct. This reduces the number of false positives, which can be disruptive in a real-world application. Finally, the F1 score, which balances precision and recall, is 97.62% for **MERCEDES**. This indicates that the model performs well across both metrics, making it a robust solution for real-time URL inspection.

Although HMLM [4] reports a slightly higher accuracy (98.12%), it uses a combination of three classifiers (LR, SVM, and DT) and a software voter, which introduces computational overhead and delays. Furthermore, TMMUC [19] achieves an accuracy of 98.55%, but its use of a Convolutional Neural Network (CNN) for feature extraction results in high computational demands, making it less suitable for real-time applications.

As it is shown by Table III by reducing the number of features, the number of parameters in proposed neural network drops from 8505 to 6105 (first row). In addition, she second

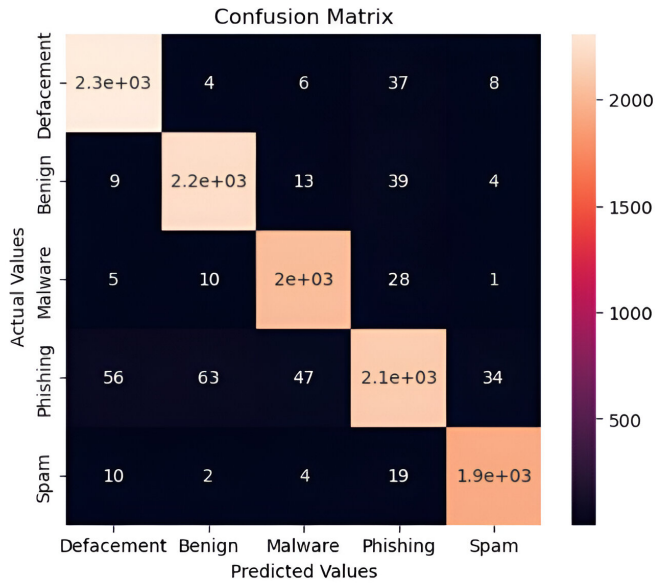

Fig. 5. Confusion Matrix of **MERCEDES**.

row of the table presents that the utilized memory space is decreased from 33.22 to 23.85 KB. Furthermore, Table III illustrates that using PCA can improve the convergence of model. Rows three to six present the accuracy of the model with and without PCA after 25, 50, 100 and 200 epochs. After 25 epochs the model trained using proposed approach achieved 95.13 percent accuracy which is 0.38 percent greater the accuracy achieved using without-PCA method. In row six and after 200 epochs, the final accuracy of presented method is slightly better than other method. Based on the data provided in table III, using PCA results in smaller number of parameters leading to less used memory space, faster convergence and better accuracy.

In addition, Fig. 5 shows **MERCEDES** confusion matrix for all five classes. The accuracy of the classifier provided for the classification of phishing, spam, malware, benign, and defacement classes is equal to 97% on average.

B. Resource Utilization

Table IV shows hardware resource utilization. As can be seen, PL has been used to implement **MERCEDES**. None of the PS resources are used and they can continue to work independently without interruption. The average resource utilization in floating-point mode is equal to 1.09% and in fixed-point mode is equal to 0.76%. Specifically, the utilization of Look-Up Tables (LUTs) and registers is low, with 0.44% and 0.17% in fixed-point mode and 0.69% and 0.53% in floating-point mode, respectively. This indicates that MERCEDES is efficient in its use of FPGA resources, leaving ample room for additional functionalities or applications.

Table V shows hardware timing parameters. As can be seen, the proposed hardware can detect malicious sites in real-time and does not cause any delay in web browsing. The proposed hardware can inspect URLs at the rate of 1976 inspections per second which indicates that it is capable of supporting multiple devices at the same time.

Fig. 6 shows the **MERCEDES** power consumption. High power consumption can increase chip temperature and lead to reduced reliability. Total power consumption in the proposed hardware at run-time equal to 1.171 watts (W). This power consumption is for when a URL check request is sent to **MERCEDES** and **MERCEDES** is processing, otherwise the power consumption is 627 mW. Due to the low power consumption of the proposed hardware, in addition to being real-time, it also has high reliability.

TABLE III
PCA EFFECTION.

Parameter	Without PCA	With PCA
Total Parameters	8505	6105
Occupied memory (KB)	33.22	23.85
Accuracy (after 25 epochs) (%)	94.75	95.13
Accuracy (after 50 epochs) (%)	96.31	96.57
Accuracy (after 100 epochs) (%)	96.82	96.98
Accuracy (after 200 epochs) (%)	97.03	97.68

TABLE IV
RESOURCE UTILIZATION OF **MERCEDES**.

	Fixed Point		**Floating Point**	
	Number	**Util. (%)**	**Number**	**Util. (%)**
CLB LUTs	1206	0.44	1903	0.69
CLB Registers	976	0.17	2918	0.53
CARRY8	77	0.22	83	0.24
CLB	255	0.74	500	1.45
LUT as Logic	1158	0.42	1828	1.03
LUT as Memory	48	0.03	75	0.05
Block RAM	32	3.50	35	3.83
DSPs	16	0.63	23	0.91
CLK Buffers	0	0	0	0
MMCM	0	0	0	0
PS8	0	0	0	0

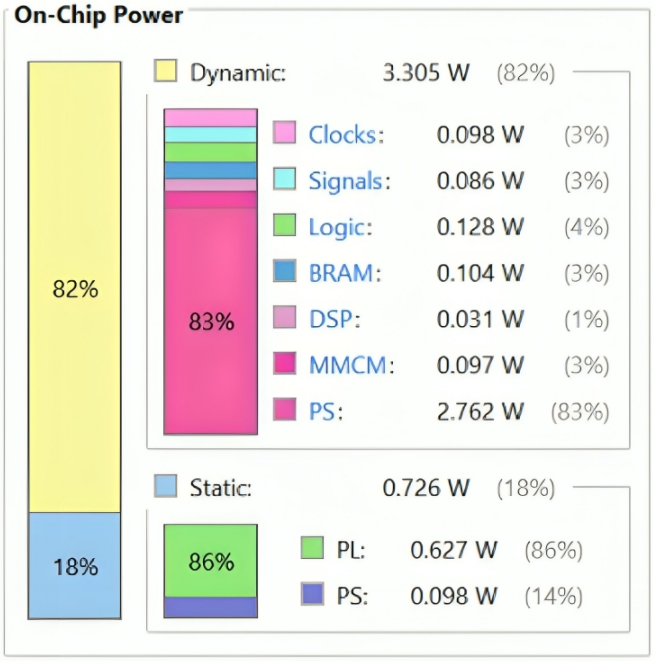

Fig. 6. On-Chip Power Consumption of **MERCEDES**.

TABLE V
TIMING PARAMETERS OF **MERCEDES**.

	Timing (Nanosecond or Cycles)
CLOCK Target	4.00 ns
CLOCK Estimated	3.454 ns
CLOCK Uncertainly	0.50 ns
Latency (Cycles)	63244 cycles
Latency (Absolute)	253,000 ns

V. Conclusion

The rapid expansion of Internet technology has significantly advanced various sectors but also increased cyberthreats, especially targeting edge and IoT devices. In this paper, we presented **MERCEDES**, an ML-based approach for real-time URL inspection, enhancing web security on these devices. We developed a lightweight MLPNN that reduces computation costs while maintaining high accuracy in classifying URLs as benign or malicious and then implemented the model on a Zynq UltraScale+ MPSoC. Our model outperforms existing methods in performance metrics, with efficient hardware resource utilization and timing parameters. Also, due to high accuracy and reduced hardware resources, **MERCEDES** can be embedded in edge devices in addition to being usable for web service providers.

Future work will focus on integrating cache memory to further enhance performance and explore additional security features. In addition, the integration of URL inspection with anomaly [25], [26] and hardware Trojan [27]–[29] detections can be pursued.

References

[1] O. Jaiswal, P. Asare, J. Gadgilwar, K. Rahangdale, P. Adekar, and L. Bitla, "Malicious address identifier (MAI): A browser extension to identify malicious URLs," In 11th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET-SIP), pp. 1-5, 2023.

[2] M. Aljabri, H. S. Altamimi, S. A. Albelali, M. Al-Harbi, H. T. Alhuraib, and N. K, Alotaibi, "Detecting malicious URLs using machine learning techniques: Review and research directions," In IEEE Access, vol. 10, pp. 121395-121417, 2022.

[3] A. S. Rafsanjani, N. B. Kamaruddin, H. M. Rusli, and M. Dabbagh, "Qsecr: Secure QR code scanner according to a novel malicious URL detection framework," In IEEE Access, vol. 11, pp. 92523-92539, 2023.

[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.

[5] T. Wu, Sh. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," In Proceedings of the Australasian Computer Science Week Multiconference, pp. 1-8, 2017.

[6] N. Reyes-Dorta, P. Caballero-Gil, and C. Rosa-Remedios, "Detection of malicious URLs using machine learning," In Wireless Networks, vol. 30, pp. 1-18, 2024.

[7] B. Xuan, J. Li, and Y. Song, "BiTCN-TAEfficientNet malware classification approach based on sequence and RGB fusion," In Computers & Security, vol. 139, pp. 103734, 2024.

[8] R. Prasad and V. Rohokale, "Cyber security: The lifeline of information and communication technology," Springer International Publishing, 2020.

[9] T. Nathezhtha, D. Sangeetha, and V. Vaidehi, "WC-PAD: Web crawling based phishing attack detection," In International Carnahan Conference Security Technology (ICCST), pp. 1–6, 2019.

[10] R. S. Jenni and S. Shankar, "Review of various methods for phishing detection," In EAI Endorsed Transactions on Energy Web and Information Technologies, vol. 5, no. 20, 2018.

[11] R. Ø. Skotnes, "Management commitment and awareness creation—ICT safety and security in electric power supply network companies," In Information and Computer Security, vol. 23, pp. 302–316, 2015.

[12] C. J. Hoffman, C. J. Howell, R. C. Perkins, D. Maimon, and O. Antonaccio, "Predicting new hackers' criminal careers: A group-based trajectory approach," In Computers & Security, vol. 137, pp. 103649, 2024.

[13] J. Bevendorff, M. Wiegmann, M. Potthast, and B. Stein, "Is Google getting worse? A longitudinal investigation of SEO spam in search engines," In Advances in Information Retrieval, 46th European Conference on IR Research (ECIR 2024), Lecture Notes in Computer Science, Springer. 2024.

[14] M. Sameen, K. Han and S. O. Hwang,"PhishHaven-An efficient real-time AI phishing URLs detection system," In IEEE Access, vol. 8, pp. 83425-83443, 2020.

[15] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," In IEEE Internet of Things Journal, vol. 6, pp. 6822–6834, 2019.

[16] J. Hwang, G. Kale, P. P. Patel, R. Vishwakarma, M. Aliasgari, A. Hedayatipour, A. Rezaei, and H. Sayadi, "Machine learning in chaos-based encryption: Theory, implementations, and applications," In IEEE Access, vol. 11, pp. 125749-125767, 2023.

[17] R. Vishwakarma, R. Monani, A. Hedayatipour, A. Rezaei, "Reliable and secure memristor-based chaotic communication against eavesdroppers and untrusted foundries," In Discover Internet of Things, vol. 3, article 2, 2023.

[18] R. Vishwakarma, R. Monani, A. Rezaei, H. Sayadi, M. Aliasgari and A. Hedayatipour, "Attacks on continuous chaos communication and remedies for resource limited devices," In 24th International Symposium on Quality Electronic Design (ISQED), pp. 1-8. 2023.

[19] N. Q. Do, A. Selamat, K. C. Lim, O. Krejcar, and N. A. Md. Ghani, "Transformer-based model for malicious URL classification," In IEEE International Conference on Computing (ICOCO), pp. 323-327, 2023.

[20] S. Wang, S. Khan, C. Xu, S. Nazir, and A. Hafeez, "Deep learning-based efficient model development for phishing detection using random forest and BLSTM classifiers," In Complexity, vol. 2020, pp. 1–7, 2020.

[21] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing e-mail using dynamic evolving neural network based on reinforcement learning," In Decision Support Systems, vol. 107, pp. 88-102, 2018.

[22] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," In Neural Computing and Applications, vol. 31, pp. 3851-3873, 2019.

[23] The European Union Agency for Cybersecurity, "ENISA Threat Landscape," https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024, 2024.

[24] Canadian Institute for Cybersecurity, "URL Dataset (ISCX-URL2016)," https://www.unb.ca/cic/datasets/url-2016.html, 2016.

[25] Y. Gao, H. M. Makrani , M. Aliasgari, A. Rezaei, J. Lin, H. Homayoun, and H. Sayadi, "Adaptive-HMD: Accurate and cost-efficient machine learning-driven framework for online malware detection using microarchitectural events," In 27th International Symposium on On-Line Testing and Robust System Design (IOLTS), pp. 1-7, 2021.

[26] Z. He, A. Rezaei, H. Homayoun, and H. Sayadi, "Deep neural network and transfer learning for accurate hardware-based zero-day malware detection," In Great Lakes Symposium on VLSI (GLSVLSI), pp. 27-32, 2022.

[27] R. Vishwakarma and A. Rezaei, "Risk-aware and explainable framework for ensuring guaranteed coverage in evolving hardware Trojan detection," In 42nd International Conference on Computer Aided Design (ICCAD), pp. 1-9, 2023.

[28] R. Vishwakarma and A. Rezaei, "Uncertainty-aware hardware Trojan detection using multimodal deep learning," In 27th Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1-6, 2024.

[29] J. Maynard and A. Rezaei, "Reconfigurable run-time hardware Trojan mitigation for logic-locked circuits," In 17th IEEE Dallas Circuits and Systems Conference (DCAS), pp. 1-6, 2024.