Detection of QR Code-based Cyberattacks using a Lightweight Deep Learning Model

Mousa Sarkhi
Department of Information Technology, Majmaah University, Saudi Arabia
mn.sarkhi@gmail.com (corresponding author)

Shailendra Mishra
Department of Information Technology, Majmaah University, Saudi Arabia
s.mishra@mu.edu.sa

Received: 8 May 2024 | Revised: 10 May 2024 | Accepted: 21 May 2024

 DOI: https://doi.org/10.48084/etasr.7777

ABSTRACT

Traditional intrusion detection systems rely on known patterns and irregularities. This study proposes an approach to reinforce security measures on QR codes used for marketing and identification. The former investigates the use of a lightweight Deep Learning (DL) model to detect cyberattacks embedded in QR codes. A model that classifies QR codes into three categories: normal, phishing, and malware, is proposed. The model achieves high precision and F1 scores for normal and phishing codes (Class 0 and 1), indicating accurate identification. However, the model's recall for malware (Class 2) is lower, suggesting potential missed detections in this category. This stresses the need for further exploration of techniques to improve the detection of malware QR codes. Despite the particular limitation, the overall accuracy of the model remains impressive at 99%, demonstrating its effectiveness in distinguishing normal and phishing codes from potentially malicious ones.

Keywords-QR code; Machine Learning (ML); cybersecurity; Deep Learning (DL); lightweight deep learning

I. INTRODUCTION

The ever-expanding digital landscape, characterized by interconnected devices and reliance on online platforms, has become a double-edged sword [1]. While it fosters innovation and efficiency, it also exposes people to growing threats - cyberattacks [2]. These malicious attempts to exploit vulnerabilities in computer systems can range from stealing sensitive data to disrupting critical infrastructure. To protect against cyberattacks, organizations rely on Intrusion Detection Systems (IDSs) [3]. These systems monitor network traffic and system activities for signs of suspicious behavior that might indicate an attack in progress [4]. However, traditional IDS approaches often have limitations. Signature-based detection systems rely on predetermined patterns or signatures of known attacks [5]. Such methods struggle to identify novel or zero-day attacks that have not been encountered before. Traditional IDSs can generate a high number of false positives, which are alerts triggered by non-malicious activities. This creates an unnecessary workload for security analysts and reduces the effectiveness of the system [6].

As cyberattack tactics continuously evolve, traditional IDS might struggle to adapt and require frequent updates or rule changes [7]. DL, a subfield of artificial intelligence, has revolutionized various domains, including cyber security [8]. DL models excel at learning complex patterns from vast amounts of data, making them well-suited for identifying anomalies and suspicious activities in network traffic. Their ability to learn and adapt provides an edge over traditional signature-based detection [9]. However, conventional DL models can be computationally expensive, requiring significant processing power and resources [9]. This can limit their applicability in real-world scenarios, especially for resource-constrained environments such as embedded systems or mobile devices [10].

QR codes are often used for information sharing, marketing, and identification purposes [10]. These two-dimensional barcodes encode data that can be easily accessed by being scanned with a smartphone or a dedicated reader [11]. Although primarily employed for legitimate purposes, their nature, that is the ability to embed information, presents a potential vulnerability. Malicious actors can exploit QR codes to carry phishing links, malware downloads, or other malicious content [12]. The widespread use of QR codes in various domains, such as marketing and identification, offers a novel and rich dataset that can be leveraged to detect anomalies and potential security threats. Lightweight DL models can process QR code images in real-time, enabling rapid detection and response to cyber threats [13]. The integration of QR codes into detection frameworks adds a layer of security by ensuring that data transmitted and processed through these codes are verified and authenticated, reducing the risk of data breaches and enhancing the overall robustness of the IDSs. This innovative approach combines the practicality of QR codes with the

advanced capabilities of DL to create a more effective and comprehensive cyberattack detection strategy.

A. Why QR Codes?

QR codes have unique characteristics that require cyberattack detection. Their widespread use in marketing and identification offers a novel input source for enhancing security measures. Focusing on QR codes, the proposed approach not only ameliorates detection accuracy, but also adds an extra layer of security to digital systems [14].

B. Main Contributions

This study aims to:

- Introduce a hybrid strategy that uses lightweight DL models to detect cyberattacks through QR codes.
- Address shortcomings of traditional IDSs by leveraging innovative methodologies.
- Explore the effectiveness of cyberattack detection on QR codes.
- Develop a practical system for real-time cyberattack detection.

The most significant contribution is the implementation of a multiobjective optimization algorithm to identify the optimal parameters for DL models. This approach improves the performance of the detection system by fine-tuning the parameters of the lightweight deep-learning model, thus ameliorating detection rates and minimizing false positives. The use of multi-objective optimization allows for a comprehensive exploration of the parameter space, ensuring that the detection system achieves optimal performance across various metrics. Furthermore, this study highlights the advantages of employing lightweight DL models for cyberattack detection. Lightweight models exhibit resilience against adversarial attacks, further bolstering the security of the detection system. The diverse nature of cyber threats and the requirement of adaptability to evolving tactics pose challenges in developing models that can effectively detect and classify a wide range of attacks. Ensuring the applicability of the proposed strategy in various industries requires considering industry-specific nuances and ethical concerns [15]. Striking a delicate balance between effective cyberattack detection and user privacy and addressing these multifaceted challenges is essential for the successful development and implementation of a reliable and ethically sound cyberattack detection system.

II. RELATED WORKS

DL models have emerged as a powerful tool in the fight against cyberattacks. Their ability to learn intricate features from vast datasets allows them to identify complex patterns and anomalies within network traffic that might indicate malicious activity. In [17, 18], the effectiveness of DL models was demonstrated in detecting various cyberattacks, including malware, phishing attempts, and denial-of-service attacks. These studies stress the potential of DL to surpass traditional signature-based detection methods, particularly to identify novel zero-day attacks. Although DL offers significant advantages, conventional models can be computationally expensive and require substantial processing power and resources. This limitation hinders their deployment in resource-constrained environments, such as embedded systems or mobile devices. Lightweight DL models address this challenge by achieving similar accuracy with a smaller footprint and lower computational cost. In [19], various techniques for lightweight models, such as model pruning and quantization, were explored making the latter suitable for real-time applications on resource-limited devices.

Ethical consequences should also be considered, as DL algorithms can cause data privacy problems [20]. In addition, the possibility of bias in training data should be thoroughly investigated. Measures should be taken to ensure the algorithm's fairness and transparency. Minimizing the possibility of inadvertent prejudice is also critical [21]. Any IDS solution should be user-friendly and easy to understand for both security experts and non-technical people [22]. The method should offer users clear and concise warnings or notifications, allowing them to take the necessary action promptly. Additionally, the approach should be verified and validated using real-world datasets and situations. The complexity and variety of genuine cyberattacks may not be fully represented [23]. Using synthetic datasets and real-world data, experiments, and simulations can offer a more realistic assessment of a method's performance and efficacy [24]. In the end, it is critical to maintain communication with industry experts and stakeholders throughout the study process. Collaboration with cybersecurity specialists, system administrators, and end users can provide useful advice and feedback on the design and execution of the approach, ensuring that it satisfies the practical demands and requirements of its intended users [25, 26].

The strategies proposed in [27, 28] offer promise in detecting cyberattacks in a variety of applications. However, further studies are needed to evaluate their scalability. Performance limits and resource needs have ethical implications. By addressing these issues, a hybrid strategy can provide a complete and strong defense against cyberattacks, thus improving the security and resilience of important data and infrastructure [29].

Despite advances in cyberattack detection using traditional IDSs and recent explorations in DL models, there is a significant research gap in the incorporation of QR code images with lightweight DL models [29-31]. Previous studies focused on known patterns and system irregularities, lacking a specialized approach to address novel and sophisticated cyber threats. The proposed model aims to bridge this gap by introducing a unique strategy that utilizes QR code images, an underexplored resource, coupled with a lightweight DL model and a multiobjective optimization algorithm. This approach presents an opportunity to significantly enhance cyberattack detection capabilities, providing a more comprehensive and effective solution to the evolving challenges posed by cyber threats [16, 32, 33].

III. METHODOLOGY

A. System Design

Figure 1 shows the design of the proposed system. The framework for malicious QR code URL detection is structured into three distinct phases. The initial phase focuses on examining the URL for redirection to address the use of shortened or redirected URLs by redirecting them to the original website to facilitate evaluation through various features. The second phase performs classification based on four features. Its primary objective is to extract pertinent information from the URL to effectively identify and classify malicious URLs. The final phase involves an evaluation based on the features. If all features provide values, the process proceeds to the final computation. However, in cases where any feature does not deliver a value, the framework leverages the values of alternative features. The culmination of these values is then used for the comprehensive detection of malicious URLs.

1) URL Detection

In the face of cybercriminals employing obfuscation techniques to elude detection, particularly through the use of short URLs to trick users into perceiving malicious URLs as legitimate, the URL redirection phase becomes imperative to improve malicious URL detection. This phase addresses the challenge by redirecting input URLs from QR codes and subjecting them to an algorithm designed to detect obfuscated URLs. The primary objective is to evaluate the features after redirecting them to their final destination. This approach aims to mitigate the risk of misclassification caused by features that incorrectly evaluate websites. When URLs do not undergo redirection, the process commences with the QR code analysis. If the QR code contains textual content, the application informs the user that the QR is benign and displays its content. However, if it includes a link, the algorithm checks if the URL comprises specific data formats and file extensions, such as IP addresses, executable files, and multimedia formats. In such a case, the original URL is displayed, concluding this phase and sending the URL to the next stage [34]. In contrast, if negative, the algorithm opens the URL in a web view, a Chrome-powered system component that allows apps to present online content on Android devices. Within the Web view, the algorithm verifies whether the URL is original or has been redirected. If the URL is original, it is displayed, concluding this phase. If redirected, the algorithm opens the original website in a second web view, employing a method known as override URL loading to redirect to the first web view until the original URL is displayed. The first web view counts the number of redirections and the other exhibits the original URL. In particular, this phase restores the website to its original URL. If redirection occurs more than ten times in the final step, the URL is presented along with any redirection. At this point, the URL is prepared to advance to the next phase to process the features [35].

2) Feature Extraction and Classification

Selecting pertinent features is a challenging task, as ineffective feature selection can lead to low detection performance. This phase initiates the classification and extraction of relevant information essential for effectively characterizing the URL. The classes are derived through parsing and analyzing various URL components. Although the utilization of numerous classes is possible and may marginally improve detection accuracy, it can increase response time. This study focuses solely on critical classes that contribute substantially to detecting malicious URLs. Critical classes are those that extract essential statistical information from a URL, aiding in distinguishing between malicious and benign websites. The identification of these classes is performed by the results obtained from detecting malicious URLs across various datasets. The classification phase is based on four primary features, as portrayed in Table I: blacklist, lexical, host-based, and content-based.

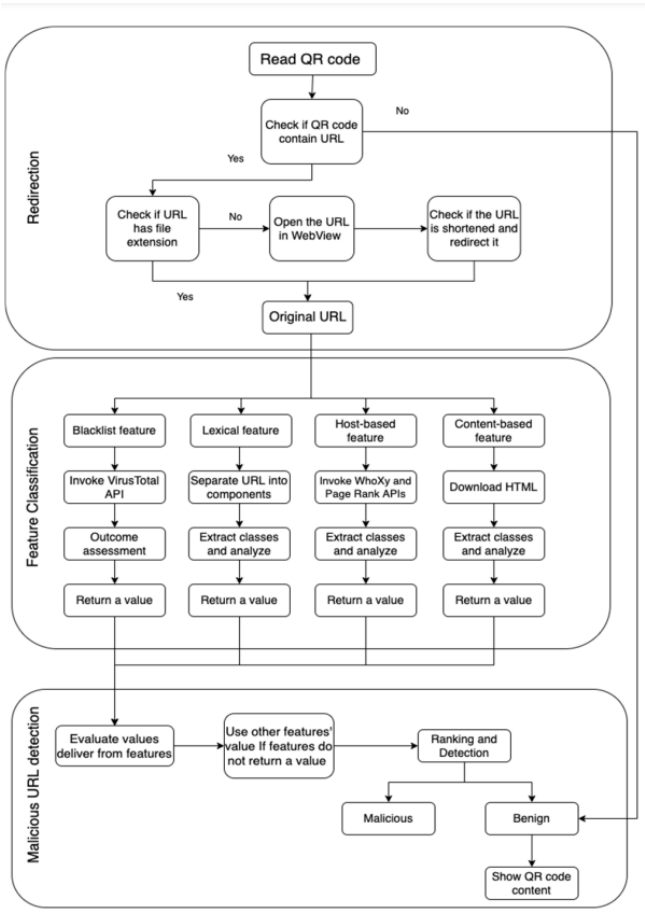

Fig. 1. Proposed framework.

The URL from the previous phase serves as input for each feature where information is collected through API calls. URL parsing or HTML downloading extracted information adheres to predefined values with classes categorized through numeric and binary values. Numerical values represent the class types that embed counting objects, while binary classes are defined by the presence of unique objects. This feature extraction and classification process is a crucial component of the overall framework, contributing significantly to the accurate detection of malicious URLs within the context of cyber-attacks involving QR codes.

TABLE I. PRIMARY FEATURES FOR CLASSIFICATION

Feature types	Details
Host-based	Extracting information about the host and web rank, this feature looks at a website's traffic statistics, popularity, host details, and the website owner's personal information.
Lexical	Focusing on the textual properties of a URL, this feature extracts various details from URL strings by breaking them into components such as the entire URL, hostname, path, and top-level domain (TLD). Each component is analyzed individually.
Content-based	This feature involves statistical information obtained by downloading the website's raw content from the server. It categorizes information into HTML, JavaScript, and certificate parts, with various classes searching through programming functions.
Blacklist	This feature scrutinizes the URL in multiple databases, determining whether it has been flagged as harmful (phishing, malware) by consulting over 90 different blacklists and web security service websites using the Virus Total API.

3) Detection of Malicious URLs

The malicious URL detection framework leverages the available data and their overall quantity. This framework is guided by a predefined static feature classification method as expressed in:

$$DF = \sum_{i=1}^{n}(F_i * 20) \quad (1)$$

where F_i denotes the feature value, with n representing the number of features. The total value is then multiplied by 20 to determine the richness of each feature on a scale of 100. The resulting DF is compared against a threshold value of 200. If it surpasses this threshold, the URL is classified as malicious, otherwise, it is deemed benign. The feature evaluation method is outlined in the rule:

if any $F_i = -1$ & other $F_i \geq 3.5$ or $blacklist \geq 3$

Then, F_i is assigned the greatest feature value.

This method evaluates the values derived from features, checking if all of them contribute to the final calculation. If any feature fails to provide a value, the method employs alternative feature values based on specific conditions. F_i is calculated using:

$$F_i = \sum_{i=1}^{n} C_i \quad (2)$$

where C_i denotes the value of the class and n represents the number of the classes for a feature. Each class can assume values based on various comparisons and conditions. The predefined values are assigned within a range to conduct analyses and comparisons, yielding a conclusive result to address the drawbacks identified in existing secure QR code scanners. The feature evaluation method is applied in this phase to enhance the accuracy of malicious URL detection. This method critically evaluates the values attained during the feature classification phase. It ensures that all features, except the lexical feature that consistently returns a value, contribute to the final calculation if any feature fails to provide a value. This framework deploys alternative feature values when faced with scenarios like API failures or unexpected server downtime. This distinctive aspect involves robust calculations, allowing the framework to accurately detect malicious URLs even if certain features do not respond. Algorithm 1 presents the operations of the malicious detection framework.

ALGORITHM 1: DETECT MALICIOUS QR CODE

```
function detectMaliciousQRCode(qrCodeData):
    decoded_data = decodeQRCode(qrCodeData)
    if decoded_data is not valid:
        return "Invalid QR Code"
    parsed_data = parseQRCodeData(decoded_data)
    if containsMaliciousPattern(parsed_data):
        return "Malicious QR Code detected"
    if isURL(parsed_data):
        if isMaliciousURL(parsed_data):
            return "Malicious URL detected"
    if isExecutableCode(parsed_data):
        if isMaliciousCode(parsed_data):
            return "Malicious code detected"
    if isEncryptedData(parsed_data):
        decryptedData = decryptData(parsed_data)
        if containsMaliciousPattern(decrypted_data):
            return "Malicious QR Code detected"
    return "QR Code is not malicious

function decodeQRCode(qrCodeData):
function parseQRCodeData(decodedData):
function containsMaliciousPattern(parsedData):
function isURL(parsedData):
function isMaliciousURL(URL):
function isExecutableCode(parsedData):
function isMaliciousCode(code)
function isEncryptedData(parsedData):
function decrypt data(encryptedData):
```

4) Performance Evaluation

Evaluation metrics serve as quantitative measures to assess the performance of a machine learning model and provide insights into how well it performs. The proposed method was evaluated using accuracy, precision, recall, and F1-score.

B. Dataset

A substantial dataset was amassed, comprising 651,191 URLs with 428,103 classified as benign or safe, 96,457 as defacement URLs, 94,111 as phishing URLs, and 32,520 as malware URLs. This dataset was meticulously curated from five distinct sources: ISCX URL 2016, malware domain blacklist, Faizan git repository, phish tank, and phish storm. Given the diverse origins, URLs from various sources were compiled into separate data frames, culminating in a final merging process to retain only the URLs and their respective class types. The final dataset was separated into training (70%), validation (15%), and testing (15%) subsets. The training dataset was employed to train the classification model, the validation dataset was utilized to tune hyperparameters and choose models, and the testing dataset was engaged to perform the final evaluation.

IV. IMPLEMENTATION

Figure 2 displays the distribution of various attack types present within the dataset. This figure offers valuable insights into the prevalence of different cyberattacks, aiding in understanding the model's training focus.

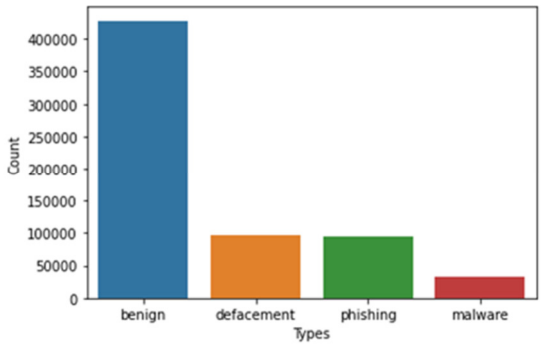

Fig. 2. Distribution of attack types.

A. *Feature Extraction*

In the feature extraction process, the code calculates the length of each URL and appends this information to the dataset. Additionally, it extracts the primary domain from each URL using a custom function, creating a new feature named domain. Additionally, it counts the occurrences of specific characters and features in each URL, introducing these counts as additional features. Finally, it identifies abnormal URLs by comparing the hostname with the entire URL and assigns a binary abnormal URL feature indicating the presence or absence of abnormality. In summary, feature extraction involves incorporating URL length primary domain character counts and abnormal URL identification as additional features in the dataset, providing valuable information for subsequent analysis and model training.

Figure 3 exhibits the results of identifying abnormal URLs within the dataset. A custom function was implemented to compare the hostname with the entire URL. This comparison helps identify URLs that deviate from standard structures, potentially indicating malicious intent. A new binary feature, named 'abnormal_URL', is introduced in the dataset, signifying the presence (1) or absence (0) of an abnormality. Figure 4, represents the extraction of character-specific features from the URLs. The code performs a character count, focusing on specific elements like symbols ("etc."). These counts are then incorporated as additional features within the dataset. This step delves into the URL composition, potentially revealing patterns or trends associated with malicious links.

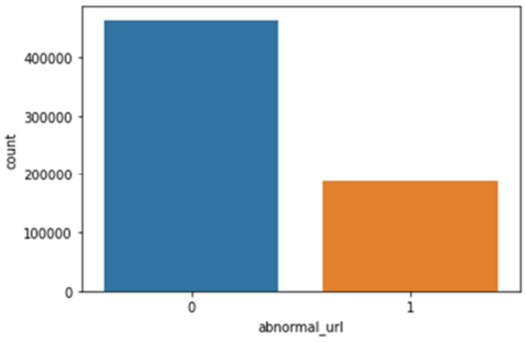

Fig. 3. Feature extraction: abnormal URL identification.

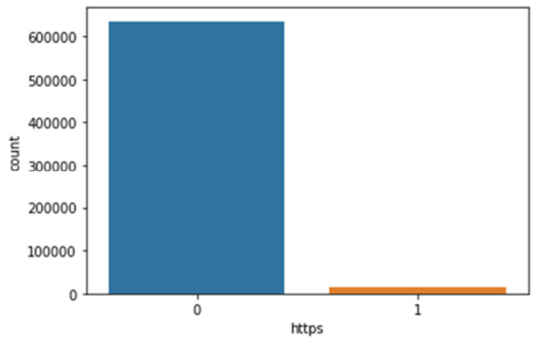

Fig. 4. Feature extraction: character counts

Figure 5 depicts the shortening service check**.** The code investigates whether a URL utilizes a shortening service. Shortened URLs can mask malicious intent and make them appear more trustworthy.

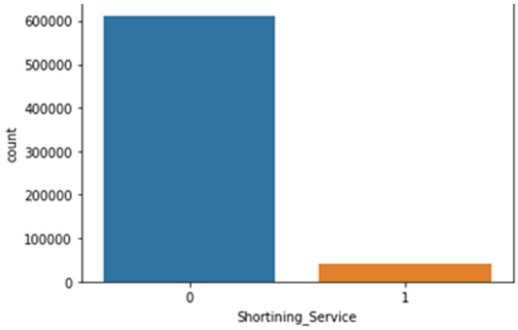

Fig. 5. Feature Analysis: shortening services.

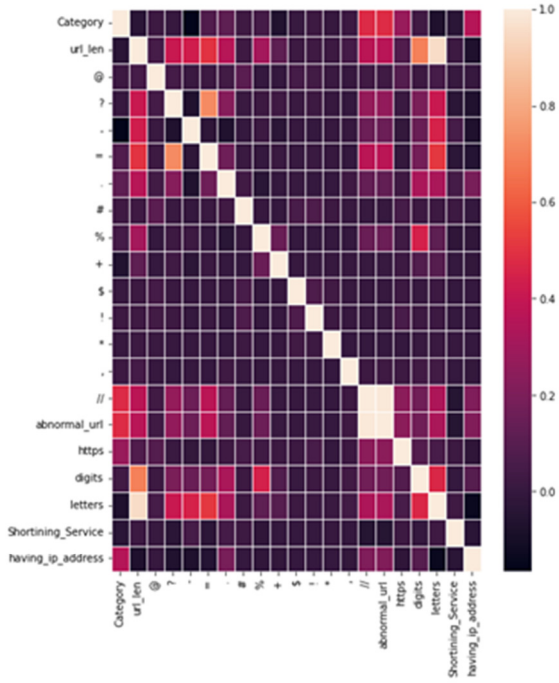

Fig. 6. Correlation heatmap between features.

A function was defined to identify whether a given URL contains an IP address. This function is crucial to determine whether URLs in the dataset have IP addresses embedded within them. Upon being defined, the function is applied to the 'URL' column of the dataset, generating a new binary feature named 'having_ip_address'. This binary feature indicates whether each URL contains an IP address, with 1 denoting presence and 0 denoting absence. Figure 6 manifests a heatmap correlation matrix between various features of the dataset. Darker shades represent stronger correlations, revealing potential relationships between features. This visualization helps identify features that might be particularly relevant for cyberattack detection.

B. Lightweight DL Model

This study implemented a lightweight DL model, using TensorFlow and Keras, based on a Convolutional Neural Network (CNN) architecture. This model employs multiple convolutional layers to extract features from QR code images, followed by dense layers for classification. Dropout layers with a rate of 99% were incorporated to prevent overfitting during training. The model achieved a remarkable test accuracy of 99%. This translates to a precision of 95% for cyberattack detection, signifying the model's exceptional ability to accurately identify malicious QR codes.

V. RESULTS AND DISCUSSION

A. Performance Evaluation Across QR Code Categories

The evaluation of the proposed cyberattack detection model for QR codes revealed nuanced insights across different classes. In particular, for normal (Class 0) and phishing (Class 1) QR codes, the model demonstrated high precision and recall scores, indicating accurate identification and effective capture of these instances. However, challenges arise with malware (Class 2), as detected in a lower recall for this category. The recall value suggests difficulty in adequately capturing instances of malware QR codes, potentially leading to missed detections. Despite these challenges, the model achieved an impressive overall accuracy of 99%, emphasizing its effectiveness in distinguishing normal and phishing QR codes. To enhance the model's performance, further analysis and potential refinements are recommended, with a particular focus on addressing the imbalances observed in precision and recall metrics for malware. Table II presents the model performance across different QR code types. It reveals high precision (0.95-0.97) for both normal (Class 0) and phishing (Class 1) codes, indicating accurate identification. However, the recall for malware (Class 2) is slightly lower (0.94), suggesting potential missed detections in this category.

TABLE II. CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.95	0.99	0.92
1 Phishing	0.95	0.96	0.90
2 Malware	0.97	0.94	0.94

B. Discussion

The model's performance is influenced by factors, such as the training dataset's quality and size, the chosen lightweight DL architecture, and the effectiveness of preprocessing techniques. Analyzing these aspects provides a comprehensive understanding of the model's strengths and weaknesses in real-world applications.

1) Limitations

Although promising, the proposed approach faces limitations. Lower recall of malware QR codes (Class 2) suggests challenges in differentiating them from normal or phishing ones. This could be due to limited malware training data, insufficient feature extraction techniques, or class imbalance in training data that favors more frequent classes. Addressing these limitations through data augmentation, exploring more effective feature extraction methods, and handling class imbalances is crucial for future work.

2) Additional Considerations

Beyond these specific challenges, further consideration is required on the generalizability to real-world scenarios, with diverse conditions and QR code designs, data privacy concerns, the trade-off between false positives and negatives, the dynamic nature of cyber threats, and the model's dependency on QR code usage. Additionally, limitations in interpretability and explainability of the lightweight DL model along with its deployment cost and scalability require further exploration.

VI. CONCLUSIONS AND FUTURE WORK

This study aimed to address the critical issue of cyberattacks using QR codes by employing a lightweight DL model. A dataset, amassed from different sources, was augmented and used to train a lighweitght deep-learning model. The results were promising in terms of overall accuracy, especially in detecting normal instances. The classification report highlighted areas of success and identified challenges, such as lower recall detecting malware. This information is crucial for understanding the model's strengths and limitations.

The proposed lightweight deep-learning model needs further refinement. To enhance its robustness to a variety of attack scenarios, hyperparameters must be optimized, different architectures must be explored, and advanced techniques must be investigated. Class imbalances may be contributing to challenges in detecting certain attack classes. To improve the performance of the model on minority classes, future research could explore techniques, such as oversampling, undersampling, and advanced loss functions. Furthermore, it is important to incorporate explainability and interpretability techniques to enhance trustworthiness and applicability, making the model's decision-making process more transparent and understandable for end-users and cybersecurity experts. Taking the research from a controlled environment to real-world deployment is a critical step and involves testing the model on diverse datasets, considering various QR code variations and potential adversarial scenarios.

ACKNOWLEDGMENT

The authors express their deepest gratitude to the Deanship of Scientific Research at Majmaah University for their consistent support and assistance with this project. Their

commitment to promoting a culture of research and innovation had a significant impact on academic endeavors.

REFERENCES

[1] N. A. Abd Rahman, A. Bahaj, H. A. Abdul Halim Sithiq, I. Farhana Kamsin, and N. K. Zainal, "Secure Parking and Reservation System Integrated with Car Plate Recognition and QR Code," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Apr. 2022, pp. 1–7, https://doi.org/10.1109/ICDCECE53908.2022.9792692.

[2] F. Abdullayeva, "Cyber resilience and cyber security issues of intelligent cloud computing systems," *Results in Control and Optimization*, vol. 12, Sep. 2023, Art. no. 100268, https://doi.org/10.1016/j.rico.2023.100268.

[3] Y. Alaca and Y. Çelik, "Cyber attack detection with QR code images using lightweight DL models," *Computers & Security*, vol. 126, Mar. 2023, Art. no. 103065, https://doi.org/10.1016/j.cose.2022. 103065.

[4] M. Alanazi, A. Mahmood, and M. J. M. Chowdhury, "SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues," *Computers & Security*, vol. 125, Feb. 2023, Art. no. 103028, https://doi.org/10.1016/j.cose.2022.103028.

[5] B. Al-Fuhaidi, W. Al-Sorori, N. Maqtary, A. Al-Hashedi, and S. Al-Taweel, "Literature Review on Cyber Attacks Detection and Prevention Schemes," in *2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, Sana'a, Yemen, Nov. 2021, pp. 1–6, https://doi.org/10.1109/ITSS-IoE53029.2021.9615288.

[6] G. A. Amoah and J.B. Hayfron-Acquah, "QR Code Security: Mitigating the Issue of Quishing (QR Code Phishing)," *International Journal of Computer Applications*, vol. 184, no. 33, pp. 34–39, Oct. 2022, https://doi.org/10.5120/ijca2022922425.

[7] D. Benalcazar, J. E. Tapia, S. Gonzalez, and C. Busch, "Synthetic ID Card Image Generation for Improving Presentation Attack Detection," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1814–1824, 2023, https://doi.org/10.1109/TIFS.2023.3255585.

[8] J. Brandman, L. Sturm, J. White, and C. Williams, "A physical hash for preventing and detecting cyber-physical attacks in additive manufacturing systems," *Journal of Manufacturing Systems*, vol. 56, pp. 202–212, Jul. 2020, https://doi.org/10.1016/j.jmsy.2020.05.014.

[9] K. Cargrill, T. Abegaz, L. C. Parra, and R. DaSouza, "Scan Me: QR Codes as Emerging Malware Delivery Mechanism," in *Proceedings of the Future Technologies Conference (FTC) 2023, Volume 2*, 2023, pp. 611–617, https://doi.org/10.1007/978-3-031-47451-4_44.

[10] R. Chen *et al.*, "Rapid Detection of Multi-QR Codes Based on Multistage Stepwise Discrimination and a Compressed MobileNet," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 15966–15979, Apr. 2023, https://doi.org/10.1109/JIOT.2023.3268636.

[11] R. Chen, Z. Zheng, Y. Yu, H. Zhao, J. Ren, and H.-Z. Tan, "Fast Restoration for Out-of-Focus Blurred Images of QR Code With Edge Prior Information via Image Sensing," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 18222–18236, Dec. 2021, https://doi.org/10.1109/JSEN. 2021.3085568.

[12] Y. W. Chow *et al.*, "Utilizing QR codes to verify the visual fidelity of image datasets for machine learning," *Journal of Network and Computer Applications*, vol. 173, Jan. 2021, Art. no. 102834, https://doi.org/ 10.1016/j.jnca.2020.102834.

[13] Z. Guo, H. Zheng, C. You, T. Wang, and C. Liu, "DMF-Net: Dual-Branch Multi-Scale Feature Fusion Network for copy forgery identification of anti-counterfeiting QR code." arXiv, Jan. 19, 2022, https://doi.org/10.48550/arXiv.2201.07583.

[14] A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, and A. Francillon, "Optical Delusions: A Study of Malicious QR Codes in the Wild," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Jun. 2014, pp. 192–203, https://doi.org/10.1109/ DSN.2014.103.

[15] N. Kumar, S. Jain, M. Shukla, and S. Lodha, "Investigating Users' Perception, Security Awareness and Cyber-Hygiene Behaviour Concerning QR Code as an Attack Vector," in *HCI International 2022 Posters*, 2022, pp. 506–513, https://doi.org/10.1007/978-3-031-06394-7_64.

[16] V. Kumar and D. Sinha, "A robust intelligent zero-day cyber-attack detection technique," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2211–2234, Oct. 2021, https://doi.org/10.1007/s40747-021-00396-9.

[17] D. O. Do Rosario Lourenco, M. V. H. Sai Sriraj, K. K. Thambi, and V. Ranjan, "Malicious URLs and QR Code Classification Using Machine Learning and DL Techniques," in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Aug. 2023, pp. 1–10, https://doi.org/10.1109/ASIANCON58793.2023.10270125.

[18] P. Mathivanan and A. B. Ganesh, "QR code based color image stego-crypto technique using dynamic bit replacement and logistic map," *Optik*, vol. 225, Jan. 2021, Art. no. 165838, https://doi.org/10.1016/ j.ijleo.2020.165838.

[19] S. A. Nawaz, J. Li, U. A. Bhatti, M. U. Shoukat, and R. M. Ahmad, "DL Applications in Digital Image Security: Latest Methods and Techniques," in *DL for Multimedia Processing Applications*, CRC Press, 2024.

[20] A. Pawar, C. Fatnani, R. Sonavane, R. Waghmare, and S. Saoji, "Secure QR Code Scanner to Detect Malicious URL using Machine Learning," in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, Ravet, India, Dec. 2022, pp. 1–8, https://doi.org/10.1109/ASIANCON55314.2022.9908759.

[21] D. Rathee and S. Mann, "Detection of E-Mail Phishing Attacks – using Machine Learning and DL," *International Journal of Computer Applications*, vol. 183, no. 47, Jan. 2022, https://doi.org/10.5120/ ijca2022921868.

[22] L. Ren and D. Zhang, "A QR code-based user-friendly visual cryptography scheme," *Scientific Reports*, vol. 12, no. 1, May 2022, Art. no. 7667, https://doi.org/10.1038/s41598-022-11871-9.

[23] C. Shaik, "Preventing Counterfeit Products Using Cryptography, QR Code and Webservice," *Computer Science & Engineering: An International Journal (CSEIJ)*, vol. 11, no. 1, Feb. 2021.

[24] H. Sultana, A. H. M. Kamal, G. Hossain, and M. A. Kabir, "A Novel Hybrid Edge Detection and LBP Code-Based Robust Image Steganography Method," *Future Internet*, vol. 15, no. 3, Mar. 2023, Art. no. 108, https://doi.org/10.3390/fi15030108.

[25] M. J. Tsai and S. L. Peng, "QR code beautification by instance segmentation (IS-QR)," *Digital Signal Processing*, vol. 133, Mar. 2023, Art. no. 103887, https://doi.org/10.1016/j.dsp.2022.103887.

[26] A. Darem, "Anti-Phishing Awareness Delivery Methods," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7944–7949, Dec. 2021, https://doi.org/10.48084/etasr.4600.

[27] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," *Computers & Security*, vol. 137, Feb. 2024, Art. no. 103587, https://doi.org/10.1016/j.cose.2023.103587.

[28] G. Varshney, R. Kumawat, V. Varadharajan, U. Tupakula, and C. Gupta, "Anti-phishing: A comprehensive perspective," *Expert Systems with Applications*, vol. 238, Mar. 2024, Art. no. 122199, https://doi.org/ 10.1016/j.eswa.2023.122199.

[29] H. A. M. Wahsheh and F. L. Luccio, "Security and Privacy of QR Code Applications: A Comprehensive Study, General Guidelines and Solutions," *Information*, vol. 11, no. 4, Apr. 2020, Art. no. 217, https://doi.org/10.3390/info11040217.

[30] A. Al-Marghilani, "Comprehensive Analysis of IoT Malware Evasion Techniques," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7495–7500, Aug. 2021, https://doi.org/10.48084/ etasr.4296.

[31] H. A. M. Wahsheh and M. S. Al-Zahrani, "Secure Real-Time Computational Intelligence System Against Malicious QR Code Links," *International Journal of Computers Communications & Control*, vol. 16, no. 3, May 2021.

[32] H. S. Wdhayeh, R. A. Azeez, and A. J. Mohammed, "A Proposed Algorithm for Hiding a Text in an Image Using QR Code," *Iraqi Journal of Computers, Communications, Control, and Systems Engineering*, vol. 23, no. 1, pp. 1–9, Mar. 2023, https://doi.org/10.33103/uot.ijccce.23.1.1.

[33] B. Zhang, D. Wu, Z. Lan, Z. Cui, and L. Xie, "Malicious code detection based on many-objective transfer model," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 22, 2023, Art. no. e7728, https://doi.org/10.1002/cpe.7728.

[34] D. Zhang, M. Shafiq, G. Srivastava, T. R. Gadekallu, L. Wang, and Z. Gu, "STBCIoT: Securing the Transmission of Biometric Images in Customer IoT," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16279–16288, Feb. 2024, https://doi.org/10.1109/JIOT.2024.3351988.

[35] M. Anwer, S. M. Khan, M. U. Farooq, and Waseemullah, "Attack Detection in IoT using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, Jun. 2021, https://doi.org/10.48084/etasr.4202.