Hindawi
Security and Communication Networks
Volume 2021, Article ID 5540670, 11 pages
https://doi.org/10.1155/2021/5540670

Research Article

Secure Real-Time Artificial Intelligence System against Malicious QR Code Links

Mohammed S. Al-Zahrani,[1] **Heider A. M. Wahsheh**,[2] **and Fawaz W. Alsaade**[3]

[1]*Department of Computer Networks and Communications, College of Computer Science and Information Technology, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia*
[2]*Department of Information Systems, College of Computer Science and Information Technology, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia*
[3]*Department of Computer Science, College of Computer Science and Information Technology, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia*

Correspondence should be addressed to Heider A. M. Wahsheh; hwahsheh@kfu.edu.sa

Received 28 January 2021; Revised 27 July 2021; Accepted 28 October 2021; Published 8 December 2021

Academic Editor: Neetesh Saxena

Recently, hackers intend to reproduce malicious links utilizing several ways to mislead users. They try to control victims' machines or get their data remotely by gaining access to private information they use via cyberspace. QR codes are two-dimensional barcodes with the capacity to encode various data types and can be viewed by digital devices, such as smartphones. However, there is no approved protocol in QR code generation; therefore, QR codes might be exposed to several questionable attacks. QR code attacks might be perpetrated using barcodes, and there are some security countermeasures. Some of these solutions are restricted to malicious link detection techniques with knowledge of cryptographic methods. Therefore, this study aims to detect malicious links embedded in 1D (linear) and 2D (QR) codes. A cybercrime attack was proposed based on barcode counterfeiting that can be used to perform online attacks. A dataset of 100000 malicious and benign URLs was created via several resources, and their lexical features were obtained. Analyses were conducted to illustrate how different features and users deal with online barcode content. Several artificial intelligence models were implemented. A decision tree classifier was identified as the most suitable model for identifying malicious URLs. Our outcomes suggested that a secure artificial intelligence barcode scanner (BarAI) is recommended to detect malicious barcode links with an accuracy of 90.243%.

1. Introduction

QR code is a machine-readable code consisting of an array of white and black squares, typically utilized for storing URLs or other information for viewing by several devices such as smartphones [1]. The retrieval of the data encoded in a QR code occurs within few seconds; thanks to the ultrahigh speeds used to verify the validity of the code received from the sensor [2].

Due to the high price of tags and identification devices, some researchers directed their attention to smartphones' cameras as an alternative identification source such as fingerprints and barcodes [3, 4].

When Denso Wave first invented the QR code in 1994, the main objective was to enable quick automobile scanning during manufacturing [1]. QR codes are now widely used in much broader contexts, such as commercial tracking and mobile tagging. A QR code can include collecting data, sensing, and reading parameters from different environments [2].

QR codes were confirmed as an international standard in 2000 [5]. The current standard version was published in 2015 [6]. QR codes can store various information types, for instance, numeric (0–9), alphanumeric (letters and numerals), and binary data (0 and 1), as well as Kanji characters (Japanese writing) [7]. Table 1 briefly describes the capacities of different data types used in QR codes.

Table 1: Description of different data types of maximum capacity in QR code [7].

Data type	Characters size
Numeric	7089
Alphanumeric	4296
Binary	2953

Typically, a QR code image contains two regions: the encoding and function pattern regions [6, 7] (see Figure 1).

QR codes have been used extensively due to the limited technological characteristics of linear (one-dimensional, 1D) barcodes. However, there has been an increasing demand for more information storage than a 1D barcode can provide [7].

QR codes have become widespread in several fields. They can be attached to any screen, poster, or product surface, effectively used in education, transportation, product tracking, ticketing, SmartTags, book returning methods in libraries, payment transfer systems, and tourism promotion [2, 8–14].

Nowadays, QR codes are used as an environment-friendly move toward ensuring a sustainable marketing strategy in various sectors, such as education, fish farming, land management, healthcare services. In this context, for instance, QR codes can improve healthcare services in effective patient identification management. Personal data can be associated with the QR codes on the patients' wristbands [15, 16]. Healthcare services can use a QR code scanner application on their smartphone to access patient information, medication, and medical reports [17–20]. QR codes enable high-speed component scanning in factories [21].

In some cases, secure barcodes can be used in IoT apps to add security, privacy, and management layers, as a free alternative to RFID tags. Barcodes can be a bridge that connects IoT objects to cloud computing, where the cloud can handle big data operations and allow security factors in IoT development [4, 22].

There are various security threats linked with QR codes. Barcodes are unreadable without a particular reader device or apps. However, there is no approved protocol in QR code generation; therefore, QR codes might be exposed to several questionable attacks. QR code attacks might be perpetrated using barcodes, and there are some security countermeasures. Some of these solutions are restricted to malicious link detection techniques with knowledge of cryptographic methods [7, 23, 24].

The main objective of this study is to detect malicious URLs embedded in barcodes (both 1D and QR codes). A cybercrime attack was proposed based on barcode counterfeiting that can be used to perform online attacks, a procedure in which a malicious 1D barcode segment is pasted over a legitimate QR code image to deceive users. In addition, we conducted tests that showed how different features affect barcode scanning. A dataset of 100 000 malicious and benign URLs was created via several resources, and their lexical features were obtained. Furthermore, five classifiers were compared to select the most suitable classifier for detecting malicious URLs. The classifiers were as following: naive Bayes (NB), support vector machine (SVM), logistic regression (LR), K-nearest neighbors (KNN), and decision tree J48 (DT) classifiers.

1.1. Contributions. The contributions of this study are summarized as follows. (i) We explore a type of barcode-in-barcode attack based on QR code counterfeiting that can be used to perform online attacks. (ii) We conducted tests that show how different factors such as size and distance affect barcode scanning. (iii) We built an AI model to detect malicious URLs encoded in barcodes based on the URL lexical properties. (iv) We applied several AI classifiers and compared them. (v) We developed BarAI based on the best model against malicious QR code links and analyzed the comparison results.

1.2. Paper Structure. The structure of this paper is as follows. Section 2 shows the literature review on QR code attacks countermeasures. Section 3 discusses the barcode injection attack, and Section 4 presents our materials and methods. Section 5 explores the experimental techniques and outcome evaluation. Section 6 discusses BarAI and the comparison results. Finally, Section 7 draws the concluding remarks and presents the topics for future work.

2. Literature Review

This section presents a literature review to ascertain the state-of-the-art current research on the available countermeasures and solutions to preserve 2D barcodes.

In this section, we first present a summary of cryptography and information security terms and algorithms. Then, we will discuss the barcodes security solutions.

The main security terminology involves three terms known as the CIA triad: confidentiality, integrity, and availability [7]. Confidentiality means protecting data from being accessed by unauthorized entities. It is commonly achieved by encrypting data so that only authorized users who have the key can decrypt and access contents. Data integrity includes assuring that data were not modified by unauthorized entities and delivered accurately. Moreover, availability indicates that the information system should be available whenever it is needed. In addition, information security includes authentication, which aims to verify the identity of users or entities, and nonrepudiation ensures that an entity cannot deny the sending of a message or sign a document [7].

Public-key cryptography (asymmetric) is a cryptographic method with two keys: public and private. It is extensively used in data encryption and authentication [7, 25]. Besides, the hash function takes a QR code content as input and delivers a fixed-size value called "hash." A hash function is a one-way method; it is hard to get the original content by processing the hash value. It is impossible to have two QR code contents with the same hash value using secure and robust hash functions.

Symmetric-key encryption is a system that uses the same secret key for encryption and decryption. Symmetric-key algorithms are considered more straightforward and faster

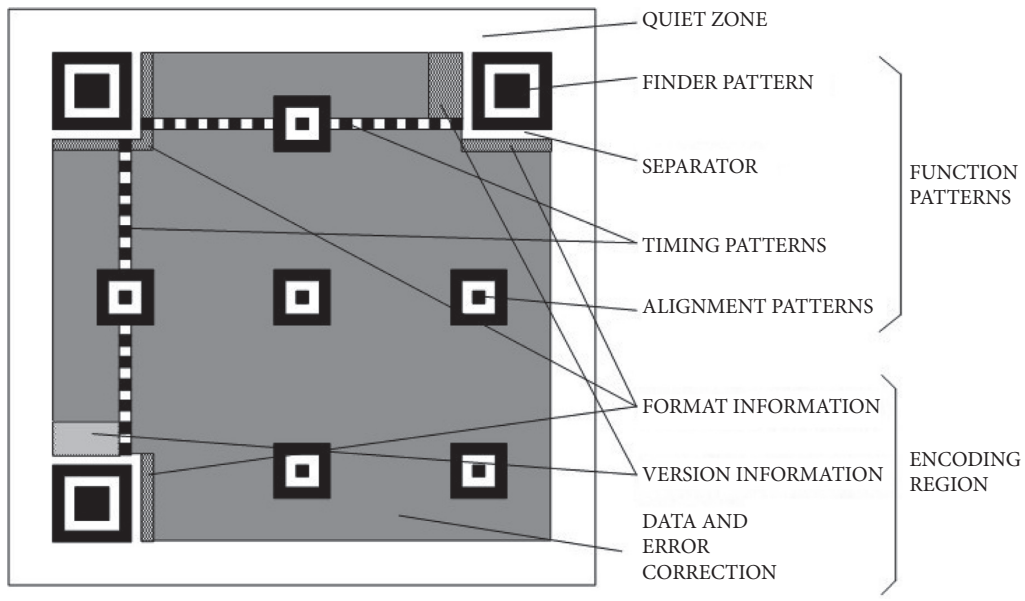


Figure 1: QR code main structure [7].

than asymmetric, but the key exchange should be established securely [7, 25].

Digital signature (DS) is a security method that uses public-key cryptography to confirm authentication, nonrepudiation, and data integrity of the content. It computes the hash and then signed it by the private key [7, 25]. Rivest–Shamir–Adleman (RSA) is a public-key cryptographic algorithm that uses a mathematical approach based on prime numbers. RSA is widely used for data encrypting and digital signatures. In comparison, elliptic curve digital signature algorithm (ECDSA) is a public-key cryptographic algorithm commonly used for digital signatures.

Advanced encryption standard (AES) is a popular symmetric-key algorithm for encrypting electronic data, and it is considered a highly secure algorithm used for confidentiality [7, 25].

References [7, 26, 27] highlighted the gaps in and limitations of the available 2D barcode protection mechanisms. The researchers compared and evaluated 2D barcode security systems using cryptographic characteristics and their security levels. They explored how various usability features affect QR code scanning and assessed several cryptographic techniques concerning QR code usability. Some asymmetric solutions lead to break QR code usability, while the elliptic curve digital signature algorithm (ECDSA) was recommended. The results also showed that symmetric methods were appropriate solutions.

Moreover, according to [28], the authors demonstrated a means of flooding the physical side of the IoT using QR codes by encoding irrelevant content or fake and phishing Web pages. In the experiments, the ECDSA was adopted to guarantee the benign usage of 2D barcodes with physical objects. Different key lengths and hash functions showed various time/space overheads.

Furthermore, in [29], the researchers highlight QR code phishing attacks. They embedded a fake Google Web page inside QR codes and performed a phishing attack. Their results showed the possibility of tricking and successfully skipping the safe browsing service provided by Google. Consequently, the researchers proposed a quick response code secure (QRCS), a comprehensive model that uses a client-server architecture and utilizes the digital signature. The proposed QRCS model adopts the ECDSA with hash function SHA2 or SHA3 (256 bits) to guarantee QR code generator authentication and data integrity. The proposed model analysis demonstrated the flexibility of implementation and efficiency against barcode attacks.

Several studies have been conducted using secret hiding schemes based on hamming code and visual secret sharing schemes to protect QR code content and private information during online transactions [30–36]. The study described in [33] was related to computational security by supposing that the attacker technique was restricted to the QR code scanner.

Reference [37] proposed a stereographic scheme to encode message authentication codes and digital signatures to authenticate data inside QR codes. The main advantage of the proposed method was that any barcode reader application could decode the barcode content. Moreover, a universal message authentication code and ECDSA with a small key length (160 bits) were used in the experiments. The results showed that the performance of the proposed scheme was better than those of the existing methods.

3. Will You Trust This Barcode? A Barcode Injection Attack

A commercial (linear) or 1D barcode is represented by horizontal lines of varying widths and spacing. Commercial barcodes are widely used to encode particular identification values, such as product IDs [38] and prices. Figure 2 shows an example of a 1D barcode that is used to store a specific URL.

Figure 2: 1D barcode example (https://is.gd/TIkkr2).

Figure 3: A readable modified 1D barcode.

The data type and length vary according to the standard used, and popular linear (1D) barcodes include universal product code (UPC), European article number (EAN), code 128 [39], and postal numeric encoding technique codes [40]. Both UPC and EAN codes support numeric data with fixed sizes, while code 128 supports variable data lengths and allows the encoding of alphanumeric data (all ASCII characters) [39]. In addition, code 39 type enables the encoding of uppercase letters A–Z, numbers 0–9, and several special characters (spaces, ., $, and %), with variable lengths [38].

Even with the limited size of 1D barcodes, they can still be used maliciously, such as encoding phishing URLs. Here, we explore an attack scenario in which a malicious 1D barcode is injected (pasted) over a legitimate QR code image to deceive users. In our study, we considered this type of attack by hiding a 1D malicious barcode inside a QR code, which is a new form of a barcode-in-barcode attack [41] that hides a malicious 2D barcode inside a QR code. In the barcode injection attack (BIA), an attacker can modify the height of the vertical lines that define the 1D barcode (compare Figures 2 and 3) and paste it over a legitimate QR code image. Note that the 1D barcode cropped image (see Figure 3) will not affect the QR code readability and will be treated as noise. The error correction feature of QR code can recover content damaged by noise; thus, both the 1D barcode and QR code will be readable.

QR codes are adaptable with various environments by using the Reed-Solomon error correction method, which facilitates reading barcodes even if some data blocks have been damaged (i.e., pasted 1D on QR code). The Reed-Solomon process involves presenting a group of redundant bits that attempt to identify, track, and correct errors based on the system itself [6, 7]. QR codes support robust four levels (percentages) of error correction capabilities for restoring the destructed data [7].

Table 2 shows the error correction levels and their tolerances for possible image damage.

Figure 4 shows examples of a BIA (left) and barcode-in-barcode attack (right). In the BIA, the two barcode types are readable: the 1D barcode containing a URL and the QR code containing random data (this code could also be a URL in another example). Thus, while reading the same barcode during different iterations, the same user could obtain two different contents. The ability to hide a 1D barcode in a BIA is visually better than that in a barcode-in-barcode attack [41]. The 1D barcode has a small size and does not have particularly distinct segments.

The QR code generator can select the appropriate error correction level depending on the type and importance of the encoded data. For example, the high error correction level (30%) can be used with severely damaged industrial barcodes distributed in a dirty environment. The low level (7%) is

Table 2: Error correction level and barcode damage tolerance [7].

Error correction level	Percentage of barcode damage tolerance (%)
Level L	7
Level M	15
Level Q	25
Level H	30

preferred with QR codes displayed electronically [41, 42]. The medium level (15%) is the most frequently used level for QR codes [1]. Attackers can utilize error correction levels to perform BIA attacks, making them reliable and dangerous.

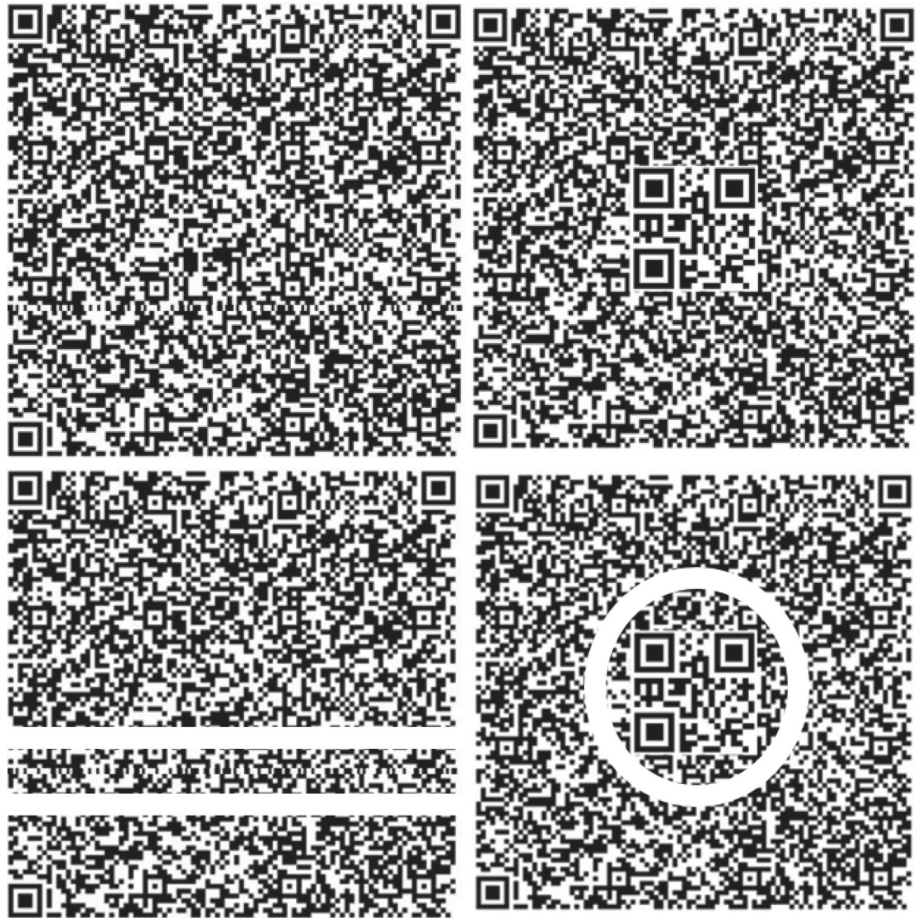
For non-expert users, it will be challenging to identify the inner hidden barcode visually. In contrast, a barcode-in-barcode attack (Figure 4, right) has a larger internal barcode size and distinct QR code finder patterns. According to [24, 41], some barcode readers can read both the outer and inner barcodes; that is, scanners can read several types of 1D and 2D barcodes.

When reading a barcode that contains a URL, almost all barcode scanners display the encoded URL content before redirecting to the Web page, and the user can decide whether to visit the URL [43, 44].

Thus, attackers can trick users by using a URL shortening service such as is.gd URL shortener [45], BL.INK [46], or Shorby [47]. These services reduce the number of characters required for URLs to 12 and display malicious URLs as short URLs to trick users.

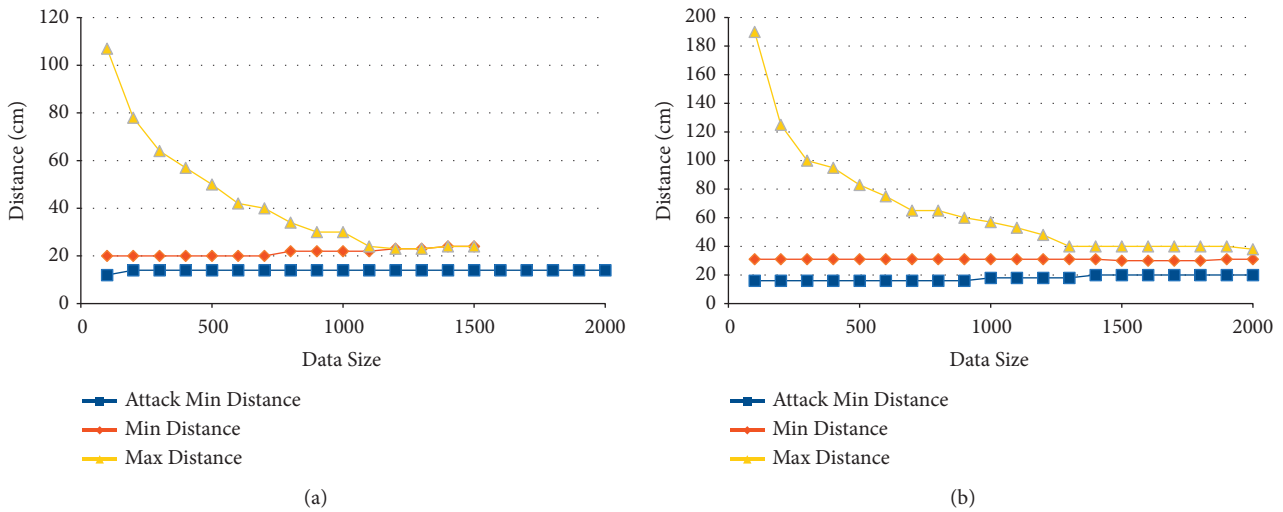
3.1. Barcode Readability Range. The QR code readability range (RR) is defined as the range of distances inside which a barcode is readable [26]. A BIA will violate the reliability of the barcode and may put readers in danger of security risks. We employed the RR experiments described in [26] and measured the RR of BIA content regarding the RR of QR code.

Figure 5 shows RR for 300×300 and 500×500 pixel barcodes, where the X-axis represents the data size in bytes, and the Y-axis represents the distance between the scanning device and barcode image. "Max distance" means the maximum distance at which the legitimate QR code can be read. In contrast, "min distance" represents the minimum distance at which the legitimate QR code can be read and the maximum distance at which the BIA code can be read. "Attack min distance" represents the minimum distance at which the BIA code can be read. The BIA code will be readable between the BIA minimum distance and the minimum distance of the legitimate QR code. The scanning device will comprehensively cover the 1D barcode as an inner barcode.



(a) (b)

Figure 4: BIA (a) and barcode-in-barcode attack [41] (b).



(a) (b)

Figure 5: RR for BIA with regards to RR of QR code. (a) (300 ∗ 300) pixels. (b) (500 ∗ 500) pixels.

For example, a QR code with a data size of 100 bytes is readable from a distance of 31–190 cm (RR = 159 cm) for a 500 × 500 pixel barcode, whereas a BIA code with the same data size is readable from a distance of 15–40 cm (RR = 25 cm). Thus, scanning the QR code without covering the whole QR code finder pattern will lead to retrieval of the BIA content, putting the user at risk. Similarly, for a 300 × 300 pixel barcode, RR for reading the legitimate QR code ranges from 40 to 190 cm, and that for reading the BIA code ranges from 10 to 20 cm. Thus, it can be concluded that

increasing the image size facilitates the implementation of a BIA.

4. Materials and Methods

Since QR codes may include suspicious online content, there are four possible attack scenarios as follows:

(i) Embedding of malicious links inside QR codes

(ii) Embedding of malicious links inside QR and BIA codes

(iii) Embedding of benign links inside QR codes and malicious links inside BIA codes

(iv) Use of several BIA codes inside QR codes and embedding with benign and malicious links to confuse users when reading the content of the same barcode

Figure 6 shows the proposed methodology for the approach adopted in this study to find solutions for these attacking scenarios as the following:

Figure 6: General procedures performed in this study.

4.1. Data Collection. One hundred thousand benign and malicious URLs were collected that might be embedded in QR and BIA codes from various environments. The dataset contained 50000 malicious URLs collected from the most recent phishing [48] and malware domains blacklists [49, 50]. Moreover, we collected 50000 benign URLs of secure websites [51, 52].

4.2. Features' Extraction. To identify malicious URLs with reduced network delay, the link-based features of URLs was analyzed. Figure 7 presents the URL structure [53].

As shown in Figure 7, cybercriminals frequently utilize the second-level domain, generic top-level domain (gTLD), and path directory to conduct cybercrimes. The domain could be a popular website, such as blog, Instagram, and TikTok. The gTLD could be edu, gov, net, and org. Cybercriminals attempt to gimmick their malicious links and bypass blacklists utilizing URL shortening services such as is.gd URL shortener [45], BL.INK [46], and Shorby [47]. Therefore, there is an increasing demand to retrieve the entire link-based features of URLs [48, 49, 54]. Correlation feature selection (CFS) [55] was utilized to assess which lexical URLs properties can be adopted in this study. CFS means that useful features positively correlated with the URL class. All our link-based features get a positive correlation with the class label. Table 3 [54] shows our adopted URL lexical properties.

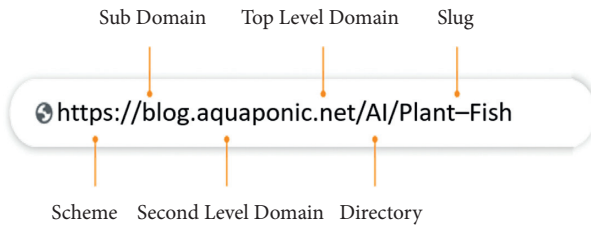


Figure 7: URL structure.

4.3. Artificial Intelligence (AI) Classifiers. In this section, we will describe the AI classifiers that we used in this study.

(1) *Naive Bayes (NB).* The NB classifier is a machine learning probabilistic classifier that utilizes Bayes' theorem and uses conditional independence assumptions between the properties [56].

(2) *Support Vector Machine (SVM).* The SVM classifier is a supervised machine learning classifier suitable for analyzing data for classification and regression. It aims to get the decision boundary that divides the data collection into two classes. The decision boundary decides if the instances are correctly classified or not [57].

(3) *Logistic Regression (LR).* The LR classifier is a machine learning classifier that depends on a probability function and is widely used for binary classification to perform predictive analysis [58].

(4) *K-Nearest Neighbors (K-NN).* The KNN classifier is a straightforward machine learning classifier that performs instance-based learning and depends on similarity measurements. In the KNN approach, the instance is assigned to the class through the majority vote of its K neighbors. The KNN method is employed for both classification and regression problems [59].

(5) *Decision Tree (DT).* The DT classifier is a popular supervised machine learning classifier employed for data classification. It works as a decision support tool that uses a flowchart of hierarchical decisions. The

Table 3: Adopted URL lexical properties [54].

ID	Attribute	Description
1	Length of URL	The total number of characters in the URL
2	# of dots	The number of dots in the URL
3	# of @	The number of (@) in the URL
4	# of semicolon	The number of semicolons in the URL
5	# of dashe	The number of dashes in the URL
6	# of slash	The number of slashes in the URL
7	Lexicon of the most popular domains	Check if the most popular domains used in the domain or path directory
8	Most popular domains	Check if the second-level domain (SLD) in the white list of the most popular domain
9	Freehosting domains	Check if the SLD in the free hosting domains
10	Backlists domains	Check if the SLD in the blacklist domains
11	Lexicon of the phishing tokens	Check if the phishing tokens used in the domain or path directory
12	Lexicon of the malware tokens	Check if the malware tokens used in the domain or path directory
13	# of tokens	The number of tokens split by (.), (/), (?), (), (%), (=), (-), (@), (\)
14	# of and	The number of (?) in the URL
15	# of percent	The number of (%) in the URL
16	# of equal	The number of (=) in the URL
17	# of questMark	The number of (?) in the URL

paths from the roots to leaves and branches represent the classification rules that indicate class labels [60].

5. Experimental Techniques and Outcome Evaluation

This section shows the outcomes of the applied five AI classifiers (NB, SVM, LR, KNN, and DT). They were evaluated using 10-fold cross-validation [61] in which the set of instances inside the data collection was split into ten parts, among which nine were used for training and one for testing. The cross-validation process is repeated ten times to test all ten parts. We compared the results using the confusion matrix, a table design employed to visualize a classifier performance (Figure 8). It includes the following prediction quality measures: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

Besides, we computed the accuracy, true positive rate (TPR), false positive rate (FPR) precision (P), recall (R), and F-measure (F-M), as expressed in (1)–(5) [61].

$$\text{Accuracy}_i = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{1}$$

$$FPR = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{2}$$

$$\text{Recall}_i = TPR = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

$$\text{Precision}_i = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$F - \text{m}easure = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{5}$$

The NB experiments yielded an accuracy of 73.928% with an error rate of 26.072%. Table 4 shows the detailed accuracy of the NB classifier.

Actual Class	Predicted Class	
	Benign	Malicious
Benign	TP	FN
Malicious	FP	TN

Figure 8: General form of the confusion matrix.

Table 4: Detailed accuracy results of the NB classifier.

Class	TP rate	FP rate	Precision	Recall	F-measure
Benign	0.964	0.485	0.665	0.964	0.787
Malicious	0.515	0.036	0.934	0.515	0.664
Weighted avg.	0.739	0.261	0.800	0.739	0.725

Besides, the NB classifier successfully classified 48179 benign URLs and 25749 malicious URLs, as shown in Table 5.

As shown in Tables 4 and 5, the NB classifier successfully obtained optimal prediction results for benign URLs (0.964), whereas it recorded a lower detection percentage for malicious URLs (0.515). Moreover, the NB classifier failed in predicting malicious URLs and classifying them as benign URLs with a false positive rate of 0.485. In comparison, NB successfully classified the malicious URLs as benign URLs within a low false positive rate (0.036).

The SVM classifier yielded an accuracy of 84.671% for the weighted average of two classes with an error rate of 15.329%. Table 6 presents the detailed results for the SVM classifier.

Table 6 demonstrates that the SVM classifier obtained close accuracy results as the NB classifier results for the benign URLs. In contrast, the SVM results are slightly improved compared with the NB results for the malicious URLs. Table 7 shows that the SVM classifier was able to assign 84671 URLs correctly among the entire dataset. These results still require enhancement to make this approach useful for malicious link detection.

Table 5: Confusion matrix of the NB classifier.

Actual class	Predicted class	
	Benign	Malicious
Benign	48179	1821
Malicious	24251	25749

Table 6: Detailed accuracy results of the SVM classifier.

Class	TP rate	FP rate	Precision	Recall	F-measure
Benign	0.912	0.219	0.807	0.912	0.856
Malicious	0.781	0.088	0.899	0.781	0.836
Weighted avg.	0.847	0.153	0.853	0.847	0.846

Table 7: Confusion matrix of the SVM classifier.

Actual class	Predicted class	
	Benign	Malicious
Benign	45613	4387
Malicious	10942	39058

The overall accuracy achieved using the LR classifier was 85.726%. The detailed results reveal enhancement compared with those of the NB and SVM classifiers for benign and malicious URL detection, as shown in Table 8.

The LR classifier was able to classify 85726 URLs correctly. In particular, it predicted 41193 malicious URLs correctly. Table 9 presents the detailed confusion matrix of the LR model.

The weighted average results of the KNN classifier, when $k = 1$, exhibit an accuracy of 89.614% with an error rate of 10.386%. Highly accurate detection was achieved for both the benign and malicious classes, with accuracies of 91.4% and 87.8%, respectively, as shown in Table 10.

By comparing the results in Tables 8 and 10, it is shown that the KNN classifier could detect the benign class better than the LR classifier. The KNN classifier detected malicious instances better than the LR approach, so it is recognized both benign and malicious URLs with the high accuracy. However, the main target of this study was to find the most suitable model for identifying the malicious URLs. Table 11 presents the confusion matrix of the KNN classifier.

As shown in Table 11, the KNN classifier could predict 43920 of 50000 malicious URLs correctly. The last classifier we applied in our experiments was the DT classifier, which yielded an overall accuracy of 90.243%, and in particular, 90.5% and 90% for detecting benign and malicious URLs, respectively. The detailed information about DT results is shown in Table 12.

The DT results exhibit slight enhancement compared to the accuracy percentage of the KNN classifier for detecting malicious URLs. The DT method correctly classified over 1094 malicious URLs more than the KNN classifier did. More details are shown in Table 13.

When comparing the classifiers, we look for the highest values of TP, precision, recall, and F-measure. On the other hand, the FP rate should be minimized. According to this, the results of Tables 4, 6, 8, and 10 show clearly that the DT classifier recorded the best results for both classes and the weighted average. It recorded more than 0.9 for TP, precision, recall, and F-measure and less than 0.1 for the FP rate.

Table 8: Detailed accuracy results of the LR classifier.

Class	TP rate	FP rate	Precision	Recall	F-measure
Benign	0.891	0.176	0.835	0.891	0.862
Malicious	0.824	0.109	0.883	0.824	0.852
Weighted avg.	0.857	0.143	0.859	0.857	0.857

Table 9: Confusion matrix of the LR classifier.

Actual class	Predicted class	
	Benign	Malicious
Benign	44533	5467
Malicious	8807	41193

Table 10: Detailed accuracy results of the KNN classifier.

Class	TP rate	FP rate	Precision	Recall	F-measure
Benign	0.914	0.122	0.883	0.914	0.898
Malicious	0.878	0.086	0.911	0.878	0.894
Weighted avg.	0.896	0.104	0.897	0.896	0.896

Table 11: Confusion matrix of the KNN classifier.

Actual class	Predicted class	
	Benign	Malicious
Benign	45694	4306
Malicious	6080	43920

Table 12: Detailed accuracy results of the DT classifier.

Class	TP rate	FP rate	Precision	Recall	F-measure
Benign	0.905	0.100	0.901	0.905	0.903
Malicious	0.900	0.095	0.904	0.900	0.902
Weighted avg.	0.902	0.098	0.902	0.902	0.902

Table 13: Confusion matrix of the DT classifier.

Actual class	Predicted class	
	Benign	Malicious
Benign	45229	4771
Malicious	4986	45014

6. Discussion

Based on the outcomes presented in Section 5, we utilized the DT classifier since it yielded the most accurate detection and prediction results for malicious links. BarAI was

Table 14: Summary of the security features of crypto barcode apps with respect to BarAI.

App	Independent	Check BIA	Key management	Crypto details	Size overhead	LP	Environment
[62]	√	√	—	—	X	√	Open
[64]	X[ab]	√	√[c]	√	√[d]	√	Open[e]
[65]	X[a]	X	√	√	√[d]	X	Close
[66]	X[a]	X	√[c]	N/A	√[d]	X	Close
[67]	X[a]	X	√[c]	√	√[d]	X	Close
[68]	X[a]	X	√[c]	N/A	√[d]	X	Close
[69]	X[a]	X	√[c]	N/A	√[d]	X	Close
[70]	X[a]	X	√[c]	N/A	√[d]	X	Close
[71]	X[a]	X	√[c]	N/A	√[d]	X	Close
[72]	X[a]	X	√[c]	N/A	√[d]	X	Close
[73]	X[a]	X	√[c]	N/A	√[d]	√	Close

[a]Depending on particular barcode generating tool. [b]Depends on particular web service (to check URLs). [c]Needing a secure way to exchange symmetrical keys. [d]To encode cryptographic data. [e]When using digital signature or checking legacy URLs. N/A means not available.

consequently implemented [62] based on the guidelines recommended by [24]. The following features describe our implementation.

(i) *Self-Supporting*. BarAI uses a DT classifier and does not require any external web service.

(ii) *Inspect BIA*. Besides the ability to detect QR code malicious links, our proposed approach can detect the malicious usage of the 1D barcode.

(iii) *Camera-Only Privilege Functionality*. This approach minimizes the levels of access to the camera (to scan the barcode image).

(iv) *Interoperability in an Open Application Environment*. Therefore, no supplementary key management or cryptographic specifications are required.

(v) *Applicable*. Neither storing nor retrieving data is required (signatures or certificates); therefore, there is no size overhead.

Few cryptographic QR codes applications offer generation and scanning services [63]. The BarSec Droid app [64] provides various symmetric and asymmetric cryptographic algorithms to secure barcodes and uses the standard JavaScript Object Notation as the formal structure with QR codes [27]. BarSec Droid [64] decodes cryptographic barcodes if its own generation app produced them. Other QR code cryptographic apps include no formal way of encoding cryptographic information inside QR codes. Each app uses its own structure. Thus, to retrieve cryptographic QR code content, the user must have the same generation tool [65–73]. Some of these apps employ unsecured encryption algorithms [65, 66]. We could not evaluate the strength of the remaining apps [68–73] because of missing cryptographic information, that is, the algorithm used or key length. These apps [65–73] use base 64 strings to represent ciphertexts that require size overhead.

Table 14 summarizes the features of applying the QR code cryptographic apps with regards to BarAI.

All apps [64–73] require a secure means of exchanging their keys, and they use encryption methods that require size overhead. Hence, these apps cannot prevent BIAs, owing to the limited size of 1D barcodes. Although digital signature techniques cannot avoid BIAs for the same reason for size overhead, BarSec Droid can still deal with and check for BIAs using a particular web service [64].

All apps [64–73] can work in closed/controlled environments, except [64] which can work also in an open environment when using a digital signature or checking URLs. Users should be aware of the potential threats when using digital signature certificates such as expired certificates, self-signed certificates, hostname mismatch, and chain of certificates issues [74, 75].

Our BarAI represents a comprehensive solution to the limitations of cryptographic apps. It works in both open and closed environments, checking all the possible suspicious online content in all barcodes types, and does not require size overhead with least privilege permissions.

7. Conclusions

This study demonstrates some QR code cybercrime attacks of phishing and malware propagation. Our work explores barcode-counterfeiting attacks and describes experiments performed to determine the effects of size and distance on barcode reading. A dataset containing 100000 URLs categorized as benign and malicious and their features were extracted for further analysis. Besides, five AI classifiers were applied, NB, SVM, LR, K-NN, and DT. The outcomes showed that the DT classifier is the most suitable model for recognizing QR code malicious links. Based on that, the BarAI was developed and later proposed as a swift warning management tool for identifying QR code malicious links among the available apps with an accuracy of 90.243%. Consequently, the proposed tool is under evaluation for a possible used to improve agroecosystems sustainability by using secure QR codes technology for durable development.

Data Availability

The dataset used in the experiments is available at https://tinyurl.com/5779wdw2.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors acknowledge the Deanship of Scientific Research at King Faisal University for their support under grant number 17122016.

References

[1] D. Wave, "QRcode.co | DENSO WAVE," *Qrcode.com*, https://www.qrcode.com/en/index.html, 2021.

[2] C. Aktaş, *The Evolution and Emergence of QR Codes*, Cambridge Scholars Publishing, Newcastle upon Tyne, UK, 2017.

[3] W. Berchtold, M. Sütter, and M. Steinebach, "Fingerprinting blank paper and printed material by smartphones," *Electronic Imaging*, vol. 4, pp. 298-1–298-7, 2021.

[4] S. Bkheet and J. Agbinya, "A review of identity methods of internet of things (IOT)," *Advances in Internet of Things*, vol. 11, no. 21, pp. 153–174, 2021.

[5] ISO/IEC 18004:2000, "ISO," 2021, https://www.iso.org/standard/30789.html.

[6] ISO/IEC 18004:2015, "ISO," 2021, https://www.iso.org/standard/62021.html.

[7] H. Wahsheh, *Secure and Usable QR Codes*, Ph.D, Universita Ca Foscari Venezia, Venice, Italy, 2019.

[8] J. Palazón and A. Giráldez, "QR codes for instrumental performance in the music classroom," *International Journal of Music Education*, vol. 36, no. 3, pp. 447–459, 2018.

[9] M. Pérez-Sanagustín, D. Parra, R. Verdugo, G. García-Galleguillos, and M. Nussbaum, "Using QR codes to increase user engagement in museum-like spaces," *Computers in Human Behavior*, vol. 60, pp. 73–85, 2016.

[10] X. Yu, Z. Fan, H. Wan et al., "Positioning, navigation, and book accessing/returning in an autonomous library robot using integrated binocular vision and qr code identification systems," *Sensors*, vol. 19, no. 4, p. 783, 2019.

[11] R. Chen, Y. Yu, X. Xu, L. Wang, H. Zhao, and H.-Z. Tan, "Adaptive binarization of QR code images for fast automatic sorting in warehouse systems," *Sensors*, vol. 19, no. 24, p. 5466, 2019.

[12] N. Gligoric, S. Krco, L. Hakola et al., "SmartTags: iot product passport for circular economy based on printed sensors and unique item-level identifiers," *Sensors*, vol. 19, no. 3, p. 586, 2019.

[13] A. Alhothaily, A. Alrawais, T. Song, B. Lin, and X. Cheng, "QuickCash: secure transfer payment systems," *Sensors*, vol. 17, no. 6, p. 1376, 2017.

[14] S.-F. Lien, C.-C. Wang, J.-P. Su, H.-M. Chen, and C.-H. Wu, "Android platform based smartphones for a logistical remote association repair framework," *Sensors*, vol. 14, no. 7, pp. 11278–11292, 2014.

[15] S. Sharara and S. Radia, "Quick Response (QR) codes for patient information delivery: a digital innovation during the coronavirus pandemic," *Journal of Orthodontics*, vol. 48, no. 2, pp. 1–9, 2021.

[16] H. Wahsheh and M. Al-Zahrani, "Secure and usable QR codes for healthcare systems: the case of covid-19 pandemic," in *Proceedings of the 12th International Conference on Information and Communication Systems (ICICS)*, IEEE, Valencia, Spain, May 2021.

[17] J. J. Mira, M. Guilabert, I. Carrillo et al., "Use of QR and EAN-13 codes by older patients taking multiple medications for a safer use of medication," *International Journal of Medical Informatics*, vol. 84, no. 6, pp. 406–412, 2015.

[18] Uzun and S. Bilgin, "Evaluation and implementation of QR code identity tag system for healthcare in Turkey," *SpringerPlus*, vol. 5, no. 1, pp. 1454–1524, 2016.

[19] M. S. Zahrani, "New trends in wireless communication: a comparative analysis and study on Li-Fi and Wi-Fi technology (strength, security, privacy and the future)," *Asian Journal of Applied Sciences*, vol. 11, no. 1, pp. 1–8, 2017.

[20] H. Wahsheh and M. Al-Zahrani, "Secure real-time computational intelligence system against malicious QR code links," *International Journal of Computers, Communications & Control*, vol. 16, no. 3, pp. 1–9, 2021.

[21] C. E. H. Ventura, R. V. Aroca, A. Í. S. Antonialli, A. M. Abrão, J. C. C. Rubio, and M. A. Câmara, "Towards part lifetime traceability using machined quick response codes," *Procedia Technology*, vol. 26, pp. 89–96, 2016.

[22] Y.-J. Tu, W. Zhou, and S. Piramuthu, "Critical risk considerations in auto-ID security: barcode vs. RFID," *Decision Support Systems*, vol. 142, Article ID 113471, 2021.

[23] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh, "Security threats and solutions for two-dimensional barcodes: a comparative study," *Computer and Network Security Essentials*, pp. 207–219, 2017.

[24] 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, p. 217, 2020.

[25] I. Alsmadi, R. Burdwell, A. Aleroud, A. Wahbeh, M. Al-Qudah, and A. Al-Omari, *Practical Information Security: A Competency-Based Education Course*, Springer, Berlin, Germany, 2018.

[26] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh, "Usable cryptographic QR codes," in *Proceedings of the 2018 IEEE International Conference on Industrial Technology (ICIT)*, Lyon, France, February 2018.

[27] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh, "Usable security for QR code," *Journal of Information Security and Applications*, vol. 48, Article ID 102369, 2019.

[28] F. Razzak, "Spamming the internet of things: a possibility and its probable solution," *Procedia Computer Science*, vol. 10, pp. 658–665, 2012.

[29] V. Mavroeidis and M. Nicho, "Quick response code secure: a cryptographically secure anti-phishing tool for QR code attacks," *Lecture Notes in Computer Science*, pp. 313–324, 2017.

[30] P.-C. Huang, C.-C. Chang, Y.-H. Li, and Y. Liu, "Efficient QR code secret embedding mechanism based on hamming code," *IEEE Access*, vol. 8, pp. 86706–86714, 2020.

[31] S. Liu, Z. Fu, and B. Yu, "Rich QR codes with three-layer information using hamming code," *IEEE Access*, vol. 7, pp. 78640–78651, 2019.

[32] Z. Fu, Y. Cheng, and B. Yu, "Visual cryptography scheme with meaningful shares based on QR codes," *IEEE Access*, vol. 6, pp. 59567–59574, 2018.

[33] T. Liu, B. Yan, and J.-S. Pan, "Color visual secret sharing for QR code with perfect module reconstruction," *Applied Sciences*, vol. 9, no. 21, p. 4670, 2019.

[34] Y. Cheng, Z. Fu, and B. Yu, "Improved visual secret sharing scheme for QR code applications," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2393–2403, 2018.

[35] M. Alajmi, I. Elashry, H. S. El-Sayed, and O. S. Farag Allah, "Steganography of encrypted messages inside valid QR codes," *IEEE Access*, vol. 8, pp. 27861–27873, 2020.

[36] B. Yu, Z. Fu, and S. Liu, "A novel three-layer QR code based on secret sharing scheme and liner code," *Security and*

Communication Networks, vol. 2019, Article ID 7937816, 13 pages, 2019.

[37] C. Chen, "QR code authentication with embedded message authentication code," *Mobile Networks and Applications*, vol. 22, no. 3, pp. 383–394, 2016.

[38] barcode, "Gtin INFO," 2017, https://www.gtin.info/barcode-101/.

[39] Dynamsoft, https://www.dynamsoft.com/blog/barcode-reader/the-comprehensive-guide-to1d-and-2d-barcodes/, 2020.

[40] "TechnoRiver - barcode software, components, and font," *Technoriversoft.com*, https://www.technoriversoft.com/, 2018.

[41] A. Dabrowski, K. Krombholz, J. Ullrich, and E. Weippl, "QR inception," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, USA, 2014.

[42] "QR codes - what is A QR code," *Qrcode.meetheed.com*, http://qrcode.meetheed.com, 2017.

[43] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. F. Cranor, and N. Christin, "QRishing: the susceptibility of smartphone users to QR code phishing attacks," *Financial Cryptography and Data Security*, pp. 52–69, 2013.

[44] H. Yao and D. Shin, "Towards preventing QR code based attacks on android phone using security warnings," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13*, Hangzhou, China, May 2013.

[45] "is.gd - a URL shortener. Mmmm, tasty URLs!" *Is.gd*, https://is.gd, 2017.

[46] "BLINK | enterprise URL shortener, branded URLs, link management," *BL.INK*, https://www.bl.ink/, 2020.

[47] "SHORBY Custom URL shortener," *Shorby.com*, https://shorby.com/retargeting/, 2020.

[48] Phishtank, https://www.phishtank.com, 2020.

[49] 2020 https://Malware-domains.com/files.

[50] A. Faizan, "Using machine learning to detect malicious URLs," 2016, https://www.kdnuggets.com/2016/10/machine-learning-detect-malicious-urls.html.

[51] Crowdflower, https://data.world/crowdflower/urlcategorization, 2020.

[52] M. N. Al-Kabi, I. M. Alsmadi, and H. A. Wahsheh, "Evaluation of spam impact on Arabic websites popularity," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 2, pp. 222–229, 2015.

[53] J. McAlpin, "What is website taxonomy (a.k.a. URL Taxonomy)," *Search Engine Journal*, https://www.searchenginejournal.com/%20website-taxonomy/361348, 2020.

[54] A. Joshi, L. Lloyd, P. Westin, and S. Seethapathy, "Using lexical features for malicious URL detection -- a machine learning approach," *arXiv.org*, 2019.

[55] S.-H. Moon and Y.-H. Kim, "An improved forecast of precipitation type using correlation-based feature selection and multinomial logistic regression," *Atmospheric Research*, vol. 240, Article ID 104928, 2020.

[56] "Naive Bayes classifier. What is a classifier?" *Data Science*, https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c, 2020.

[57] N. Bambrick, "Support vector machines for dummies; a simple explanation - AYLIEN news API," *Aylien.com*, https://aylien.com/blog/support-vector-machines-for-dummies-a-simple-explanation, 2020.

[58] "What is logistic regression? - statistics solutions," *Statistics Solutions*, https://www.statisticssolutions.com/what-is-logistic-regression/, 2020.

[59] O. Harrison, "Machine learning basics with the K-nearest neighbors algorithm," *The Medium*, https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761, 2018.

[60] T. Plapinger, "What is a decision tree?" *The Medium*, https://towardsdatascience.com/what-is-a-decision-tree-22975f00f3e1, 2017.

[61] I. Witten and E. Frank, *Data Mining*, Morgan Kaufmann, San Francisco, CA, USA, 2016.

[62] "BarAI QR code scanner," 2021, https://sites.google.com/site/heiderawahsheh/apps.

[63] "Google play store," *Play.google.com*, https://play.google.com/store?hl=en, 2020.

[64] H. Wahsheh and F. Luccio, "Evaluating security, privacy and usability features of QR code readers," in *Proceedings of the 5th International Conference on Information Systems Security and Privacy*, Czech Republic, 2019.

[65] "QR droid private," *DroidLa*, http://qrdroid.com, 2016.

[66] "QR & barcode security," *Madiff Net*, https://play.google.com/store/apps/details?id=com.trustbookin.qrcodebarcodesecurity, 2017.

[67] D. Tengler, "Crypto message," 2018, https://play.google.com/store/apps/details?id=cz.crypto_message_free.apk.

[68] Ec Qr, *Ecrubit Consultancy Service*, http://www.ecrubit.co, 2018.

[69] "Fastest QR barcode reader: scanner and generator," *I-Plex Technology*, https://play.google.com/store/apps/details?id=com.iplextech.barcode.scanner, 2018.

[70] "QR code secret," *SOLEZERO.COM*, https://play.google.com/store/apps/details?id=com.solezero.android.qrcodesecret, 2019.

[71] enQRCode, "My encrypted MSG-QR code," *Liliandroid*, https://play.google.com/store/apps/details?id=com.liliandroid.enqrccmyencryptedmsg, 2019.

[72] S. QrCode, *SaiFinTex*, https://apkpure.com/secret-qrcode/org.saifintex.qrcode, 2019.

[73] Global Input App, *Iterative Solution Limited*, https://play.google.com/store/apps/details?id=uk.co.globalinput, 2018.

[74] C. D'Orazio and K. Choo, "A technique to circumvent SSL/TLS validations on iOS devices," *Future Generation Computer Systems*, vol. 74, pp. 366–374, 2017.

[75] M. Ukrop, L. Kraus, V. Matyas, and H. Wahsheh, "Will you trust this tls certificate? perceptions of people working in it," in *Proceedings of the 35th Annual Computer Security Applications Conference*, San Juan, PR, USA, December 2019.