International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

http://dx.doi.org/10.12785/ijcds/XXXXXX

QR Shield: A Dual Machine Learning Approach Towards Securing QR Codes

Hissah Almousa[1], Arwa Almarzoqi*[1], Alaa Alassaf[1], Ghady Alrasheed[1] and Suliman A. Alsuhibany[1]

[1]*Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia*

Received Mon. 20, Revised Mon. 20, Accepted Mon. 20, Published Mon. 20

Abstract: Quick Response (QR) codes are extensively employed due to their compatibility with smartphone technology and the technological advances of QR code scanners. With the ever-increasing adoption and utilization of QR codes in several real-life contexts, finding effective and efficient security mechanisms to maintain their integrity has become crucial. Despite their popularity, QR codes have been exploited as potential attack vectors through which attackers encode malicious URLs. Such attacks have become a critical concern, necessitating effective countermeasures to mitigate them. This research paper proposes a dual machine learning-based model called QR Shield. This QR shield aims to identify and detect the malicious links embedded in QR codes by utilizing a benchmark dataset of URLs. The effectiveness of QR Shield was validated using four evaluation metrics, and experimental outcomes demonstrated an accuracy rate of 96.8%. Based on these findings, the QR Shield exhibits a high potential to detect malicious QR codes, which confirms the ability to generalize the proposed QR Shield to various real-life domains and applications. Additionally, the present study contributes to the broader area of the QR code studies by offering comprehensive insight into the ability and potential of using supervised machine learning models for QR code security and privacy.

Keywords: Cybersecurity; Supervised Learning; Machine Learning Models; QR Code Security; Malicious URL; Experimental Study

1. INTRODUCTION

The employed of Quick Response (QR) codes has become an essential part of our daily modern lives. A QR code resembles a barcode but is composed of small square shapes, which give it a higher data storage capacity than traditional barcodes. Furthermore, when a QR code is scanned, it enables the user to promptly retrieve the stored information, hence the "quick response" name. The use of QR codes continues to expand in several contexts due to the numerous opportunities and advantages they offer to individuals, businesses, and organizations. Multiple real-life domains employ QR codes, such as business applications [1], warehousing and healthcare [2], commercial tracking and mobile tagging [3], advertising, mobile payments, e-ticketing [4], [5], and numerous other contexts. In addition to these uses, QR codes are frequently implemented to encode URLs [4]. Accordingly, the QR code has become the first preference for communicating URLs from billboards to smartphones [5]. The origin of QR codes dates to 1994, with a Japanese company, Denso Wave, playing a pivotal role in their development. Denso Wave sought a way to provide accurate tracking of vehicle and automotive parts during the manufacturing process [3]. To accomplish this goal, Denso Wave focused on enhancing and improving barcode technology to accommodate Japanese characters, which ultimately culminated in the invention of QR codes. Since their inception, QR codes have been refined and have proliferated into extensive implementations owing to the inherent convenience of accessing the information they contain via smartphone cameras, thereby eliminating the necessity for specialist equipment. In conjunction with the widespread utilize and diverse applications of the QR codes, they have emerged as a vector for several kinds of attacks targeting users [4]. Kharraz et al. [6] defined QR code-based attacks as harmful attempts to deceive individuals into scanning QR codes that direct them to harmful content. For example, [3] highlighted many security threats exploiting QR codes, including phishing, malware propagation, barcode tampering and imitation, and Structured Query Language (SQL) and command injections. Such malware propagation is one of the main harmful activities associated with QR codes [7]. The process of malware propagation encompasses the utilization of QR codes by attackers to redirect individuals to malicious URLs that are indistinguishable by the human eye. Subsequently, the attacker installs malware on the targeted device without the user's knowledge by exploiting vulnerabilities in programs [3], [7]. Thus, accelerating the development of malicious URL detection integrated into QR codes has become essential for QR code security. In light of such threats, QR code

E-mail address: (441211996, 441212001, 441212000, 441211998, salsuhibany)@qu.edu.sa

attacks have become a critical concern, necessitating effective countermeasures to mitigate them and to preserve the security and integrity of QR codes. Many techniques have been proposed to safeguard QR codes from malicious embedded URLs, such as employing encryption techniques to protect the data embedded in QR codes [8], [9], [10], [11] or utilizing secure QR code generators and scanners that provide built-in malicious content detection capabilities [12], [13], [14], [15], [16]. The primary objective of this study is to introduce and examine the efficacy of a dual machine learning- based model to recognize the malicious URLs embedded in QR codes. The proposed dual model described in this article referred to as QR Shield. This QR Shield combines two machine learning classifiers, Random Forest (RF) and Extreme Gradient Boosting (XGBoost), by utilizing a benchmark dataset consisting of four different types of URLs [17]. The following portions of this paper are organized as follows: Section 2 presents a brief background, and section 3 reviews the relevant literature. Section 4 provides a comprehensive explanation of the methodology, and section 5 presents an in-depth explanation of the experimental setup and the evaluation of the outcomes. The results are presented and discussed in section 6, followed by an exploration of potential future research directions and the conclusion in section 7.

2. Background

This section aims to introduce the pertinent theoretical terms and concepts regarding barcodes and QR codes. Two-dimensional (2D) barcodes are an improvement of the functionality and features of traditional barcodes (1D), including two key aspects: data capacity and robustness [18]. Such improvements have significantly enhanced the quality of barcodes for industrial and economic purposes. Researchers and businesses have investigated the integration of 2D barcodes to store data with smartphones incorporating built-in cameras to scan and decode data. This integration has resulted in numerous applications, one of which is that the 2D barcodes facilitate easy communication between physical objects and digital realms [3]. Moreover, 2D barcodes operate as portable databases, enabling users' seamless access to information [19]. In the context of 2D barcodes, diversity reigns supreme to suit different needs, and there exist various types, including the QR code, VSCode, and Data Matrix [19]. The most common type of 2D barcode is the QR code, which is a machine-readable code that stores data in a grid of squares and dots organized in a specific pattern. Due to their distinctive qualities, such as the ability to store a large amount of data, encode different kinds of data [5], and rapid scanning capability [20], QR codes are widely used to encode many forms of data, such as symbols, binary data, control codes, and multimedia data [19]. Moreover, URLs can also be encoded and stored within QR codes [4]. Table I displays different data types that can be stored within QR codes, along with their corresponding character size.

Structurally, a QR code consists of square modules organized in an array. These are surrounded by quiet area borders, which, in turn, provide accurate barcode reading [21]. Additionally, the structural characteristics of QR codes comprise two primary components: function patterns and encoding regions [3]. Function patterns, located in specific places within the QR code, assist the QR code scanners in correct recognition and orientation for decoding [21]. There are four function patterns: finder, separator, timing, and alignment. The encoding region stores version information, format information, and data and error-correcting codewords. Figure 1 shows a visual representation of the primary structural features of a QR code.

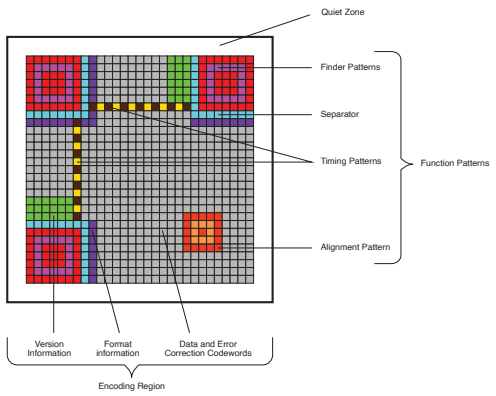


Figure 1. The structure of QR codes [21]

There are 40 unique versions of QR codes, each encompassing a range of variations. Version 1 has dimensions of 21x21 modules, from which 133 units can be used to store encoded data, while version 40 releases the largest QR code, with dimensions of 177x177 modules and 4296 units that can be used for encoding alphanumeric characters [21]. Despite such variation, all these versions have the same structural attributes, although they are used for different purposes. For example, smartphone applications exclusively utilize QR code versions 1 through 10 due to camera restrictions [19], such as resolution, optical zoom, and autofocus [22]. According to [23], the QR codes are readable from various angles, allowing for successful data decoding even in the presence of partial coverage or damage to the code. This capability is due to the use of powerful error correction based on Reed–Solomon codes [5]. There are four error-correction levels: Low (L 7%), Medium (M 15%), Quartile (Q 25%), and High (H 30%), the percentages indicate the data restoration rates for total codewords [19]. If the error correction level is high, the area reserved for error correction codewords will increase, whereas the area reserved for actual data will decrease; thus, error correction level L is usually preferred [4]. Due to their ability to encode, store, and provide easy retrieval of a large amount of data, QR codes are widespread in numerous fields. Despite the numerous benefits of QR codes in our daily lives, their popularity has made them a popular target for

TABLE I. Data types and sizes storable in QR codes

Data type	Character Size
Numeric data	7,089
Alphanumeric data	4,296
Binary data	2,953
Japanese Kanji and Kana data	1,817

attacks [4], [24]. Attackers often employ the malicious QR codes that direct users to malicious links [4]. There are two key attack vectors to exploit QR codes:

- The attacker substitutes the legitimate QR code with a malicious one by superimposing it on top of the original QR code.
- The attacker alters specific modules of the QR code. This approach involves modifying the encoded content by changing the color of a certain module in the QR code, which redirects the user to malicious content upon scanning [24].

The increasing popularity of QR codes and of user attacks that exploit such popularity necessitate the enhancement of QR code security by designing and implementing effective mechanisms to mitigate QR code threats and counter potential attacks.

3. Literature Review

In response to the growing number of potential attacks exploiting and targeting QR codes, numerous studies have been published offering several security measures to improve QR code security. This section discusses current advanced research on countermeasures and approaches used to protect and preserve the QR codes. Several QR code scanners provide scanning services; some also include security measures to protect users from potential URL threats, whereas others lack any security safeguards. Building on the analysis conducted by [11] on QR code security services, Section 3-A presents cryptographic-based approaches, while section 3-B presents URL-based approaches.

A. Cryptographic-Based Approaches

Cryptographic-based approaches are used in QR code scanners to encrypt, sign, and manage access to the QR content, guaranteeing the confidentiality and privacy of the QR codes [11]. Bani-Hani et al. [8] proposed a secure QR code system that generates and reads QR codes. The system ensures data integrity by utilizing a Digital Signature Algorithm (DSA) and hashing algorithm, as well as by generating a public key and a private key. Additionally, the QR code undergoes a verification process. If the verification is unsuccessful, a warning message alerts the users about a malicious QR code, and they can then decide whether to use the QR code contents. Along similar lines, Mavroeidis and Nicho [9] proposed a cryptographic secure anti-phishing tool for QR code attacks, referred to as a Secure QR Code Solution (QRCS). The QRSC is a client-server-based cryptographic system that utilizes hash functions and digital signatures to guarantee the integrity and authenticity of QR codes. Each QR code scanned undergoes verification using its public key and digital signature. Upon unsuccessful verification, the QR code is classified as malicious and, hence, is blocked and thwarted at the initial scanning phase. An Anti-Malware Phishing Scanner (AMPS) was proposed by Niranjan Hegde et al. [10] that provides a QR code scanner with built-in capabilities to detect malicious content. The AMPS scanner also incorporates encryption and decryption features for QR codes, employing the Advanced Encryption Standard (AES) mechanism. Wahsheh and Luccio [11] introduced BarSec, which is an extensive barcode scanner that uses both symmetric and asymmetric cryptographic techniques to create and scan secure barcodes. Furthermore, BarSec provides a range of security functionalities, encompassing barcode authentication, data integrity, confidentiality measures, and access control; consequently, it is capable of generating and reading QR codes securely. Despite such possibilities, adding security features to QR code scanners can be challenging for several reasons, including the use of inadequate key lengths and weak encryption and decryption algorithms. The most significant challenge is that the cryptographic QR codes necessitate users to generate and read QR codes using the same application [25]. Therefore, the aforementioned challenges might clarify the reasons for the limited adoption of cryptographic-based methodologies.

B. URL-Based Approaches

One of the most prevalent tactics targeting users' devices is the incorporation of malicious URLs into QR codes. The existing literature proposes various technologies to detect malicious URLs, including artificial intelligence (AI) approaches and black- and whitelists [11]. The use of black- and whitelists, also known as the blacklist approach, is a security URL-based approach for QR code scanners. The blacklist approach refers to the method by which URLs are compared against a database of known phishing URLs [26]. Yao and Shin [12] proposed SafeQR, a QR code scanner that can detect both phishing attacks and malware attacks. Additionally, it attempts to find malicious URLs by utilizing two blacklist datasets from Google Safe Browsing [27] and PhishTank [28] through the application programming interface (API). It also augments user perceptions of security by providing a preemptive notification prior to accessing the website associated with the URL, enabling users to make informed security decisions when scanning QR codes. As suggested by an empirical study,

warning messages serve as a proficient security indicator to safeguard users against phishing URLs [29]. Similarly, the work on AMPS presented by [10] was also based on blacklist use, employing blacklist datasets from Virustotal Service [30] via an API. Ohigashi et al. [31] proposed detection methods for fake QR codes and also accounted for environmental changes. Their approach works to classify whether the QR code is fake or legitimate by analyzing the information acquired from the error-correcting process. Three detection methods were proposed, each handling different error usages; consequently, the combination of the proposed methods showed more robust results than utilizing a single detection method. Several researchers have demonstrated the significant efficiency of AI techniques in the detection of malicious URLs embedded in QR codes. As part of AI, the adoption of machine learning techniques with generalization and resilience against actual attacks has become the dominant detection method for malicious URLs [32]. The study conducted by Al-Zahrani et al. [13] confirmed the efficacy and capability of AI in detecting suspicious content and malicious URLs embedded in all barcode types. The researchers introduced a robust and secure AI-based barcode scanner called BarAI. Of the five AI classifiers used, the DT classifier achieved the highest performance in identifying and detecting malicious barcode links, with an accuracy rate of 90.24%. Similarly, a comprehensive investigation was conducted by Maheshwari et al. [14] to assess the performance of seven machine learning classifiers in the detection of malicious URLs; their RF classifier demonstrated superior performance compared to the other classifiers, achieving an accuracy rate of 92.65%. Another study employed four machine learning classifiers to implement a secure QR code scanner to detect malicious URLs; the bidirectional Long Short-Term Memory (LSTM) classifier achieved the highest performance, with an accuracy rate of 83.79% [15]. Rafsanjani et al. [16] proposed QsecR, an Android application designed to offer QR code scanning functionalities that prioritize security and privacy based on malicious URL detection. QsecR analyzes the URL associated with each scanned QR code, focusing on its feature values. The overall feature values are aggregated to determine a final score, which is then compared to a predefined threshold value. If the final score is below the threshold, the URL is classified as benign; otherwise, the URL is classified as malicious. According to their experimental outcomes, the QsecR demonstrated a detection accuracy rate of 93.50%.

4. Methodology

The identification of malicious URLs embedded in replicated QR codes involves a classification process in which supervised machine learning techniques are employed to categorize raw URLs into various classes. As illustrated in Fig. 2, the workflow of the proposed dual model, QR Shield, begins with the acquisition of a QR code image. Subsequently, the image is decoded to reveal the embedded URL, which is then extracted. QR Shield, a meticulously designed dual model that integrates RF and XGBoost algorithms, plays a central role in safety assessment. It thoroughly analyzes the URL for safety attributes. If the URL is determined to be benign or safe, then the user is seamlessly redirected to the URL. If the URL fails the safety assessment, a warning message is promptly displayed. This structured workflow ensures a systematic and secure user experience while leveraging the novelty of the QR Shield model.

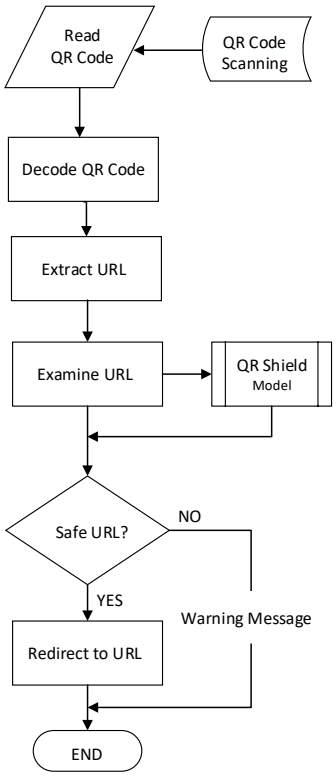


Figure 2. A flowchart of QR Shield

The methodology section comprises several critical subsections, each contributing significantly to the development and implementation of the QR Shield model for QR code safety assessment. These subsections encompass dataset interpretation, data preprocessing and feature extraction, the selection of machine learning algorithms, and the execution of the QR Shield model.

A. Dataset

The proposed approach relies on a benchmark dataset called Malicious URLs [17], which contains a total of 651,191 URLs. This extensive dataset encompasses various URL categories, including 428,103 benign URLs, 96,457 defacement URLs, 94,111 phishing URLs, and 32,520 malware URLs. The dataset consists of two primary columns: 'URL' and 'Type', where 'Type' indicates the class of maliciousness.

B. Data Preprocessing and Feature Extraction

Prior to model development, the URL dataset undergoes a comprehensive data preprocessing procedure to achieve data normalization. This process involves applying the MinMaxScaler technique, which transforms numerical data into a specific range, typically between 0 and 1. The primary objective of MinMax scaling is to standardize the scales of various features within the dataset, thus enabling direct comparisons and mitigating the influence of features with larger scales on the analysis [33]. Eq. 1 provides the representation of the MinMaxScaler formula.

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

Feature extraction techniques are then applied to transform the URLs into a structured format that enables effective analysis by the machine learning models, thereby enhancing overall performance. In a research study [34], the primary attribute groups for identifying malicious URLs were defined as follows:

- Lexical Features: These encompass path average, average token count within the domain, maximum token length, URL length, and main domain attributes. Lexical features provide valuable insights into the structure of URLs.
- Network-Based Characteristics: Derived from URL host information, these attributes reveal details about server behavior, identity, and their contribution to the classification of malicious URLs.
- Content-Based Characteristics: These features are obtained from web page content upon downloading, thus raising security concerns and requiring extensive data extraction due to their high data load.

In this research, lexical features are extracted from raw URLs. Table II provides a description of the key features that serve as input attributes for the model. The choice of lexical features in this research is guided by their effectiveness in capturing essential information from raw URLs, which makes them highly valuable for the machine learning model. These features function as a key factor in distinguishing between benign and malicious URLs by highlighting patterns and anomalies indicative of malicious intent. By incorporating lexical features, the proposed approach aims to enhance the machine learning model's capability to make well-informed decisions by considering URL characteristics. These features simplify the model's interpretation of raw text data, thereby contributing to the accurate classification of URLs as benign, defacement, phishing, or malware.

1) Exploratory Data Analysis

This section examines the feature distributions across the four URL classes. A notable observation from the analysis of the 'checking_ip_address' feature is that only malware URLs exhibit the presence of IP addresses, as shown in Fig. 3. Furthermore, the examination of the 'abnormal_url' feature reveals that defacement URLs display a notably higher distribution, as presented in Fig. 4.

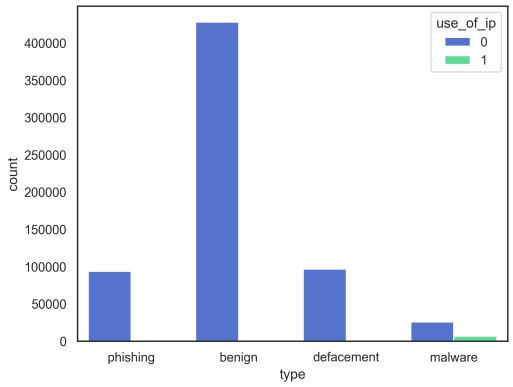


Figure 3. Distribution of the use of IP

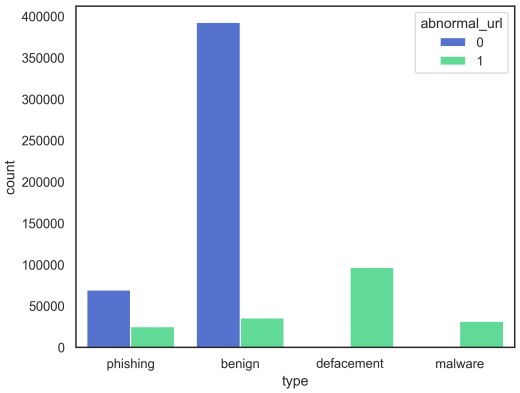


Figure 4. Distribution of abnormal URLs

Fig. 5 demonstrates that, for the 'suspicious_urls' feature, benign URLs show the highest distribution, followed by phishing URLs. This trend is attributed to the composition of suspicious URLs, which often include transaction and payment-related keywords such as PayPal, login, bank, account, or free. The prevalence of such keywords in benign URLs, typically corresponding to genuine banking and payment-related websites, results in the highest distribution within this category.

C. Machine Learning: Algorithms Selection

The supervised machine learning method of classification entails the input of a labeled dataset into a model. The model is equipped with prior knowledge regarding the training dataset, which may consist of both structured and unstructured data. The procedure consists of several key steps, including data preprocessing, model training, and data classification. Notably, in the context of this research, feature engineering plays a pivotal role as a mandatory step. Feature engineering is employed to extract relevant features

TABLE II. Features extraction for malicious URLs dataset

Lexical Group			
ID	Feature	Data Type	Description
1	checking_ip_address	Boolean	Checks if an IP address replaces the domain name
2	abnormal_url	Boolean	Identifies irregular URL patterns
3	google_idx	Boolean	Verifies if the URL is indexed by Google Search Console
4	count_dot	Numeric	Number of dots in URL
5	find_www	Numeric	Counts 'www' occurrences in the URL to identify anomalies
6	count_at	Numeric	Counts the '@' symbol in the URL
7	find_dir	Numeric	Counts the number of directories in the URL
8	no_of_embed	Numeric	Counts the occurrence of '//' in the URL
9	shortening_service	Boolean	Detects the use of URL shortening services
10	count_https	Numeric	Detects the presence or absence of HTTPS in the URL
11	count_http	Numeric	Counts the 'HTTP' occurrences in the URL
12	count_per	Numeric	Counts the occurrence of the '%' symbol in the URL
13	count_ques	Numeric	Counts the occurrence of the '?' symbol in the URL
14	count_dash	Numeric	Counts the occurrence of the '-' symbol in the URL
15	count_equal	Numeric	Counts the occurrence of the '=' symbol in the URL
16	url_length	Numeric	Measures the length of the URL
17	hostname_length	Numeric	Evaluates the length of the hostname
18	suspicious_words	Boolean	Checks for the presence of suspicious words
19	digit_count	Numeric	Counts the number of digits in the URL
20	letter_count	Numeric	Counts the number of letters in the URL
21	count_special_chars	Numeric	Counts the occurrence of special characters in the URL
22	fd_length	Numeric	Determines the length of the first directory in the URL
23	tld_length	Numeric	Measures the length of the toplevel domain (TLD) in the URL
24	secure_http	Boolean	Indicates whether the URL uses a HTTPS protocol

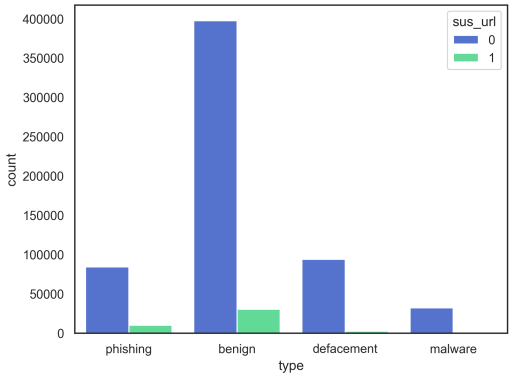


Figure 5. Distribution of suspicious URLs

or labels that facilitate a more effective understanding of the underlying data patterns, subsequently improving the overall classification process, as described in section 4-B. While labels are relevant to each dataset, the classes collectively encompass the entirety of the dataset. Numerous studies and applications have investigated employing machine learning methods to identify malicious URLs [13], [16], [34], [35], [36], [37]. In the machine learning model selection domain, XGBoost is the primary choice due to its accelerated execution and competitive performance compared to leading algorithms [38]. An extensive evaluation of various classifiers, including Decision Tree (DT), Logistic Regression (LR), Naïve Bayes (NB), RF, and K-Nearest Neighbor (KNN), was conducted to rigorously assess their classification accuracy. Table III illustrates the overall performance of the designated classifiers, utilizing weighted evaluation across all four classes.

Based on this analysis, RF emerged as the most accurate classifier, leading to its selection as a complementary component to the XGBoost classifier in the development of the QR Shield model. This research utilizes RF and XGBoost, which are well-known supervised machine learning techniques [39], [40].

1) Random Forest Algorithm

Random Forest is a group learning technique regularly employed in supervised classification and regression tasks. This approach involves constructing a diverse collection of DTs during training and returning the class label or mean prediction (in regression) derived from the individual trees. The RF classifier, a specific form of ensemble learning, fits numerous DTs on different data subsets, resulting in a composite model composed of these trained DTs. Due to its parallel architecture, the RF classifier not only outperforms standalone DTs and effectively mitigates overfitting issues in most scenarios [41] but also excels in terms of speed in contrast to alternative state-of-the-art classifiers [42]. Nonetheless, it is important to recognize that the training process for the RF classifier can be computationally demanding, rendering it less suitable for real-world applications [39].

2) XGBoost Algorithm

Extreme Gradient Boosting, or XGBoost, is an advanced software system that leverages the power of Gradient Tree Boosting [43], thereby enabling the efficient management of large-scale machine learning tasks. Its remarkable predictive capabilities and streamlined training procedure have led to its consistent success across various applications and domains. The core concept of this algorithm is its continuous integration of new trees and the dynamic partitioning of features to facilitate tree expansion. A new function is learned to efficiently model the most recent anticipated residual when each tree is added, thus contributing to its

TABLE III. Classifiers performance overview

Classifier	Accuracy	Precision	Recall	F1 Score
DT	0.958	0.96	0.96	0.96
LR	0.841	0.82	0.84	0.82
NB	0.815	0.82	0.82	0.81
RF	**0.966**	**0.97**	**0.97**	**0.97**
KNN	0.944	0.94	0.94	0.94

remarkable performance in predictive analytics and data-driven decision-making [40].

D. Implementation of Proposed Model

In the pursuit of an improved and resilient QR code security classification system, QR Shield employs a dual model that consolidates the strengths of two distinct classifiers through ensemble stacking techniques. This collaboration aims to enhance the overall accuracy and robustness of QR Shield.

In the context of stacking, the algorithm combines predictions from various machine learning algorithms. Initially, all other algorithms undergo training with the available dataset. Subsequently, the combinator algorithm is trained to produce a conclusive prediction by incorporating the predictions of these algorithms as supplementary inputs [44].

The stacking classifier within QR Shield encapsulates the collaborative efforts of the base models and the meta-model. During the training phase, predictions from the base models serve as inputs for the meta-model, which intelligently learns to weigh and combine these predictions by leveraging the distinctive strengths of each base model. This stacking technique significantly enhances the predictive accuracy of QR Shield compared to traditional ensemble boosting and bagging methods. The meta-model adeptly adjusts to the complexity and subtlety of the base model outputs. The resulting ensemble, consisting of both the base models and the meta-model, encapsulates a holistic understanding of QR code security characteristics.

The model composition of QR Shield is fortified by two machine learning classifiers: RF and XGBoost algorithms. This ensemble forms the foundation of QR Shield. Additionally, a meta-model, specifically an XGBoost classifier, is introduced to integrate the predictions produced by the base models. The role of the meta-model is to refine individual predictions, thereby enhancing the overall classification accuracy. For further insights into the implementation of both the base and meta-models, refer to Listing 1 displaying the corresponding code snippet.

Listing 1. QRShield classifier initialization

```
from sklearn.ensemble import
    RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import
    StackingClassifier
base_models = [
    ('random_forest', RandomForestClassifier(
    n_estimators=100, eta=0.05)),
    ('xgboost', XGBClassifier(n_estimators
    =100))]
meta_model = XGBClassifier(n_estimators=100,
    eta=0.05)
```

The code segments in Listing 2 instantiate, train, and employ the stacking classifier, enabling QR Shield to harness the strengths of diverse base models cohesively and adaptively for QR code security classification.

Listing 2. QRShield classifier implementation

```
QRShield = StackingClassifier(estimators =
    base_models,
                    final_estimator =
    meta_model)
QRShield.fit(X_train, y_train)
y_pred_QRS = QRShield.predict(X_test)
```

In QR Shield, the safety evaluation of a given QR code is designed to provide users with critical insights into the nature of the QR code's content. The process begins by extracting various features from the QR code, which offers valuable information about its composition and structure. These features include elements such as the presence of certain URL components, character counts, and other relevant attributes outlined earlier in section 4-B. The central purpose of this feature extraction is to gain a comprehensive understanding of the QR code's characteristics. This information serves as the foundation for evaluating the QR code's safety and reliability. The core of this safety assessment is facilitated by a pretrained QR Shield model, which combines RF and XGBoost classifiers. This model has been trained on a vast dataset of QR codes, allowing it to make informed classifications based on the extracted features. Based on the consolidation of features, the QR Shield model assigns one of four possible classes to the QR code: 'benign', 'defacement', 'phishing', or 'malware'. Each class represents a distinct level of safety. A 'benign' classification indicates a safe QR code that poses no risks to the user. In such cases, the user is promptly redirected to the associated URL, ensuring a secure experience. However, when the QR code falls into the 'defacement', 'phishing', or 'malware' categories, it is considered potentially unsafe. In these instances, the model issues a warning message to the user. This warning serves as a critical alert that highlights potential security threats associated with the QR code. It urges users to exercise caution and be aware of their interactions with such codes.

Ultimately, the QR Shield safety assessment mechanism is designed to empower users with information, hence enabling them to make informed decisions about the QR

codes they encounter. It acts as a protective shield, thereby enhancing security and trust in the use of QR technology.

5. Experimental Setup and Outcome Ecaluation

In this study, the 80-20 splitting rule was applied to partition the dataset of malicious URLs. All machine learning models employed in this study, including RF, XGBoost, and QR Shield, were tested using a 20% data subset that was not seen during training on the remaining 80% of the data.

For the experimental setup, we used Python version 3.11 on a macOS operating system. The hardware configuration included a 2.7 GHz Dual-Core Intel Core i5 processor with 8 GB of RAM to ensure the efficient and reliable execution of our machine learning experiments. To assess model performance, four distinct metrics were utilized that each incorporated the following variables:

- **True Positives (TP):** Represents the count of correctly classified malicious URLs.
- **True Negatives (TN):** Denotes the number of benign URLs correctly identified.
- **False Positives (FP):** Signifies benign URLs incorrectly classified as malicious.
- **False Negatives (FN):** Indicates malicious URLs mistakenly classified as benign.

Accuracy, denoted in Eq. 2, serves as an overall measure of the method's success in performing predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

Precision, as represented by Eq. 3, reflects the proportion of accurate positive predictions. A low precision value suggests a tendency for the method to categorize even benign URLs as malicious.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall, as expressed in Eq. 4, quantifies the ratio of actual positive cases that are correctly identified.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F1 Score, represented by Eq. 5, is employed to provide a comprehensive evaluation of model performance. The F1 Score serves as the harmonic mean of precision and recall, thus effectively balancing the trade-off between them.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

Precision and recall are integral in addressing imbalanced datasets, in which benign URLs outnumber malicious ones. A low recall indicates that the method is failing to detect malicious URLs. In scenarios where the cost of missing a malicious URL is higher than rejecting a benign one, high recall becomes pivotal. Conversely, high recall coupled with low precision suggests that an excessive number of URLs are being incorrectly classified as malicious. Ideally, both precision and recall should be maximized. More importantly, they should maintain a balance to indicate unbiased and well-rounded predictions [15].

6. Results and Discussion

The effectiveness of the QR Shield model in enhancing QR code classification accuracy is evidenced by the experimental results. For individual performance, the RF classifier achieved an impressive overall accuracy of 96.6%, while XGBoost was not far behind, with an accuracy of 96.2%. Further detailed accuracy metrics for RF and XGBoost are presented in Tables **??** and **??**, respectively.

QR Shield, which integrates these two algorithms using the stacking technique, attained a slightly higher accuracy of 96.8%, as demonstrated in Table VI. This outcome underscores the successful fusion of RF and XGBoost to improve QR code classification's overall performance.

To obtain deeper insights into the model's performance, confusion matrices were employed. These matrices, displayed in Tables VII, VIII, and IX for RF, XGBoost, and QR Shield, respectively, provide a detailed breakdown of TP, TN, FP, and FN. They reinforce the notion that QR Shield is highly promising in classifying both benign and malicious URLs, including defacement, phishing, and malware classes.

A comparison of our QR Shield classifier with previous studies, such as [13] and [16], is enlightening. These previous works achieved accuracy rates of 90.2% using DTs and 93.5% using their proposed model of QsecR, as illustrated in Table X, which were considered commendable at the time. However, the exceptional performance demonstrated by QR Shield sets new benchmarks and indicates it as one of the leading models. It not only achieved the highest values for TP but also excelled in precision, recall, and F1-measure while effectively mitigating FP rates. QR Shield is a testament to the evolving landscape of QR code classification and represents a significant leap forward in this domain.

7. Conclusion and Future Work

As the utilization of QR codes has become pervasive in our daily lives, understanding and addressing QR attacks has become crucial to safeguarding the integrity and security of the data encoded within QR codes as well as to ensuring the safety of users. This paper sheds light on malicious URLs embedded in QR codes as one kind of QR code attack. Although the investigation into the detection of malicious QR codes continues to expand in various directions, this paper employed machine learning techniques to contribute to the literature. Therefore, this

TABLE IV. Performance analysis of RF

class	Precision	Recall	F1 Score
Benign	0.97	0.98	0.98
Defacement	0.98	0.99	0.99
Phishing	0.99	0.94	0.96
Malware	0.91	0.86	0.88
Weighted avg	0.97	0.97	0.97

TABLE V. Performance analysis of XGBoost

Class	Precision	Recall	F1 Score
Benign	0.97	0.99	0.98
Defacement	0.97	0.99	0.98
Phishing	0.97	0.92	0.95
Malware	0.91	0.83	0.87
Weighted avg	0.96	0.96	0.96

TABLE VI. Performance analysis of QRShield

Class	Precision	Recall	F1 Score
Benign	0.97	0.99	0.98
Defacement	0.99	0.99	0.99
Phishing	0.98	0.95	0.96
Malware	0.91	0.86	0.89
Weighted avg	0.97	0.97	0.97

TABLE VII. Confusion matrix of RF classifier

	Predicted Benign	Predicted Defacement	Predicted Phishing	Predicted Malware
Actual Benign	84305	30	31	2214
Actual Defacement	9	19162	67	330
Actual Phishing	8	13	6140	65
Actual Malware	1299	87	266	16213

paper introduced QR Shield, a dual machine learning-based model, as a contribution to enhancing the security of QR codes. QR Shield utilizes two primary machine learning classifiers, RF and XGBoost, on a benchmark dataset of URLs. QR Shield was evaluated in terms of accuracy, precision, recall, and F1 score. According to the experimental outcomes, the proposed model achieved an accuracy rate of 96.8%, thereby demonstrating its strong potential for detecting malicious QR codes. However, this study is subject to some limitations imposed by dataset size, as the high volume of the dataset requires more powerful computational resources for efficient and fast implementation. Based on these findings, future research directions could involve enhancing QR Shield's accuracy through the integration of deep learning techniques for a more secure QR code environment. Additionally, QR Shield could be integrated into smart applications to promote accessibility and usability.

array

A. Abbreviations and Acronyms

All abbreviations in the paper are annotated and described in Table XI.

B. Authors and Affiliations

All authors are affiliated with the Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia

Acknowledgment

Researchers would like to thank the Deanship of Scientific Research, Qassim University, for funding the publication of this project.

References

[1] V. S. Bhamidipati and R. S. Wvs, "A novel approach to ensure security and privacy while using qr code scanning in business applications," in *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE, 2022, pp. 198–203.

[2] K. Saranya, R. Reminaa, and S. Subhitsha, "Modern applications of qr-code for security," in *2016 IEEE International Conference on Engineering and Technology (ICETECH)*. IEEE, 2016, pp. 173–177.

[3] H. A. M. Wahsheh, "Secure and usable qr codes," 2019.

[4] K. Krombholz, P. Frühwirt, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl, "Qr code security: A survey of attacks and challenges for usable security," in *Human Aspects of Information Security, Privacy, and Trust: Second International Conference, HAS 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings 2*. Springer, 2014, pp. 79–90.

[5] A. Dabrowski, K. Krombholz, J. Ullrich, and E. R. Weippl, "Qr inception: Barcode-in-barcode attacks," in *Proceedings of the 4th ACM workshop on security and privacy in smartphones & mobile devices*, 2014, pp. 3–10.

[6] A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, and A. Francillon, "Optical delusions: A study of malicious qr codes in the

TABLE VIII. Confusion matrix of XGBoost classifier

	Predicted Benign	Predicted Defacement	Predicted Phishing	Predicted Malware
Actual Benign	84497	59	59	2531
Actual Defacement	15	19114	153	455
Actual Phishing	14	20	5970	120
Actual Malware	1095	99	322	15716

TABLE IX. Confusion matrix of QR Shield classifier

	Predicted Benign	Predicted Defacement	Predicted Phishing	Predicted Malware
Actual Benign	84452	16	31	2239
Actual Defacement	10	19121	35	249
Actual Phishing	11	22	6162	84
Actual Malware	1148	133	276	16250

TABLE X. Comparison with existing work

Study Reference	Classifier	Accuracy
[13]	Decision Trees	90.2%
[16]	QsecR	93.5%
This work	QR Shield	96.8%

TABLE XI. List of Abbreviation

Abbreviation	Meaning
QR	Quick Response Code
QR Shield	Quick Response Code Shield
URL	Uniform Resource Locator
API	Application Programming Interface
RF	Random Forest
XGBoost	Extreme Gradient Boosting
DT	Decision Tree
LR	Logistic Regression
NB	Naïve Bayes
KNN	K-Nearest Neighbor
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

wild," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2014, pp. 192–203.

[7] V. Sharma, "A study of malicious qr codes," *International Journal of Computational Intelligence and Information Security*, vol. 3, no. 5, pp. 21–26, 2012.

[8] R. M. Bani-Hani, Y. A. Wahsheh, and M. B. Al-Sarhan, "Secure qr code system," in *2014 10th International Conference on Innovations in Information Technology (IIT)*, 2014, pp. 1–6.

[9] V. Mavroeidis and M. Nicho, "Quick response code secure: a cryptographically secure anti-phishing tool for qr code attacks," in *Computer Network Security: 7th International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2017, Warsaw, Poland, August 28-30, 2017, Proceedings 7*. Springer, 2017, pp. 313–324.

[10] P. Hemavathi, N. Hegde, R. Bharti, R. Sur, and S. Priyanka, "Anti-malware phishing qr scanner," *International Journal of Innovative Science and Research Technology*, vol. 3, no. 5, 2018.

[11] H. A. 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.

[12] 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*, 2013, pp. 341–346.

[13] M. S. Al-Zahrani, H. A. Wahsheh, and F. W. Alsaade, "Secure real-time artificial intelligence system against malicious qr code links," *Security and Communication Networks*, vol. 2021, pp. 1–11, 2021.

[14] B. Janet, R. J. A. Kumar *et al.*, "Malicious url detection: a comparative study," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 1147–1151.

[15] 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)*. IEEE, 2022, pp. 1–8.

[16] 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," *IEEE Access*, 2023.

[17] "Malicious URLs dataset." [Online]. Available: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

[18] J. Z. Gao, L. Prakash, and R. Jagatesan, "Understanding 2d-barcode technology and applications in m-commerce-design and implementation of a 2d barcode processing solution," in *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, vol. 2. IEEE, 2007, pp. 49–56.

[19] H. Kato and K. T. Tan, "Pervasive 2d barcodes for camera phone applications," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 76–85, 2007.

[20] R. Focardi, F. L. Luccio, and H. A. Wahsheh, "Usable security for qr code," *Journal of Information Security and Applications*, vol. 48, p. 102369, 2019.

[21] S. Tiwari, "An introduction to qr code technology," in *2016 international conference on information technology (ICIT)*. IEEE, 2016, pp. 39–44.

[22] J. Tierno and C. Campo, "Smart camera phones: limits and applications," *IEEE Pervasive Computing*, vol. 4, no. 02, pp. 84–87, 2005.

[23] "QRcode.comDENSO WAVE." [Online]. Available: https://www.qrcode.com/en/

[24] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, and E. Weippl, "Qr code security," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, 2010, pp. 430–435.

[25] R. Focardi, F. L. Luccio, and H. A. Wahsheh, "Security threats and solutions for two-dimensional barcodes: a comparative study," *Computer and network security essentials*, pp. 207–219, 2018.

[26] K. A. Latif, B. Sugiantoro, and Y. Prayudi, "Anti-qrishing real-time technique on the qr code using the address bar-based and domain-based approach on smartphone," *International Journal of Cyber-Security and Digital Forensics*, vol. 8, no. 2, pp. 134–144, 2019.

[27] "Safe Browsing – Google Safe Browsing." [Online]. Available: https://safebrowsing.google.com/

[28] "PhishTank | Join the fight against phishing." [Online]. Available: https://www.phishtank.com/

[29] S. Egelman, L. F. Cranor, and J. Hong, "You've been warned: an empirical study of the effectiveness of web browser phishing warnings," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 1065–1074.

[30] "VirusTotal - Home." [Online]. Available: https://www.virustotal.com

[31] T. Ohigashi, S. Kawaguchi, K. Kobayashi, H. Kimura, T. Suzuki, D. Okabe, T. Ishibashi, H. Yamamoto, M. Inui, R. Miyamoto *et al.*, "Detecting fake qr codes using information from error-correction," *Journal of Information Processing*, vol. 29, pp. 548–558, 2021.

[32] J. Yuan, Y. Liu, and L. Yu, "A novel approach for malicious url detection based on the joint model," *Security and Communication Networks*, vol. 2021, pp. 1–12, 2021.

[33] D. Borkin, A. Némethová, G. Michal'čonok, and K. Maiorov, "Impact of data normalization on classification model accuracy," *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, vol. 27, no. 45, pp. 79–84, 2019.

[34] M. Aljabri, H. S. Altamimi, S. A. Albelali, M. Al-Harbi, H. T. Alhuraib, N. K. Alotaibi, A. A. Alahmadi, F. Alhaidari, R. M. A. Mohammad, and K. Salah, "Detecting malicious urls using machine learning techniques: review and research directions," *IEEE Access*, vol. 10, pp. 121 395–121 417, 2022.

[35] S. R. Abdul Samad, S. Balasubaramanian, A. S. Al-Kaabi, B. Sharma, S. Chowdhury, A. Mehbodniya, J. L. Webber, and A. Bostani, "Analysis of the performance impact of fine-tuned machine learning model for phishing url detection," *Electronics*, vol. 12, no. 7, p. 1642, 2023.

[36] U. S. DR, A. Patil, and M. Mohana, "Malicious url detection and classification analysis using machine learning models," in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE, 2023, pp. 470–476.

[37] M. Mehndiratta, N. Jain, A. Malhotra, I. Gupta, and R. Narula, "Malicious url: Analysis and detection using machine learning," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2023, pp. 1461–1465.

[38] J. Gu and H. Xu, "An ensemble method for phishing websites detection based on xgboost," in *2022 14th international conference on computer research and development (ICCRD)*. IEEE, 2022, pp. 214–219.

[39] O. Baines, A. Chung, and R. Raval, "Random forest classification algorithm," *Leicester Undergraduate Mathematical Journal*, vol. 2, 2020.

[40] S. He, B. Li, H. Peng, J. Xin, and E. Zhang, "An effective cost-sensitive xgboost method for malicious urls detection in imbalanced dataset," *IEEE Access*, vol. 9, pp. 93 089–93 096, 2021.

[41] K. K. Dutta, A. Victor, A. G. Nathu, M. A. Habib, D. Parashar *et al.*, "Kannada alphabets recognition using decision tree and random forest models," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2020, pp. 534–541.

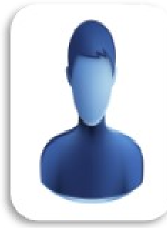
[42] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

[43] K. Sadaf, "Phishing website detection using xgboost and catboost classifiers," in *2023 International Conference on Smart Computing and Application (ICSCA)*. IEEE, 2023, pp. 1–6.

[44] Y. Khukalenko, I. Stopochkina, and M. Ilin, "Machine learning models stacking in the malicious links detecting," *Theoretical and Applied Cybersecurity: scientific journal, Vol. 5, No. 1*, 2023.

Author 1 Name short biography . . . . . .

Author 2 Name short biography ….



Author 3 Name short biography ….