



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

QR Code Phishing Detection Using QR And URL Feature Extraction with Machine Learning

A. Samiha, E Julie Ruth

Department of Computer Applications and Research Centre, Sarah Tucker College(Autonomous), Tirunelveli-7.

ABSTRACT

QR codes provide a convenient way to access websites and online services, but they can also be misused to redirect users to phishing websites. Since the destination URL is not directly visible from the QR code image, users may unknowingly access potentially harmful websites. This paper presents a machine learning-based QR Code Phishing Detection System that analyzes both QR code image characteristics and the URL embedded within the QR code. A dataset containing 5,000 labeled URL samples, consisting of 2,515 legitimate and 2,485 phishing samples, is used for developing and evaluating the system. Five QR code features and fifteen URL-based features are extracted and combined to form a 20-feature representation. The data is preprocessed, including TLD encoding, and divided into training and testing sets using an 80:20 split. Four machine learning algorithms, namely Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine, are trained and evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. Random Forest provides the best overall performance, achieving 94.90% accuracy, 97.71% precision, 92.16% recall, 94.85% F1-score, and 98.61% ROC-AUC. The model also achieves a mean accuracy of 95.06% using 5-fold cross-validation. The final system supports QR code image upload and webcam scanning, followed by QR code detection, decoding, feature extraction, and classification. A Python Tkinter-based graphical user interface provides the decoded URL, prediction result, and model performance comparison. The proposed system demonstrates that combining QR image and URL features can provide an effective approach for identifying potentially phishing QR codes.

Keywords: QR Code Security, Phishing Detection, Machine Learning, URL Feature Extraction, Random Forest

1. INTRODUCTION

QR codes have become an important part of modern digital communication because they provide a convenient way to access websites and online services. Users can scan a QR code using a camera or mobile device to access the information encoded within it. However, QR codes can also be misused by attackers to redirect users to phishing websites that may attempt to obtain sensitive information such as usernames, passwords, account details, or other personal information. Since the visual appearance of a QR code does not directly reveal whether its encoded URL is trustworthy, users may access suspicious websites without recognizing the associated risk. Traditional QR code scanners mainly focus on decoding the information stored in the QR code and displaying the corresponding URL, rather than determining whether the destination is legitimate or potentially malicious. Therefore, an automated mechanism for analyzing QR codes and their embedded URLs can provide an additional layer of security before users access the destination website.

The proposed project, QR Code Phishing Detection Using QR and URL Feature Extraction with Machine Learning, focuses on classifying QR codes as either legitimate or phishing by analyzing both the QR code image and its decoded URL. The system combines five QR image features, including image width, image height, black pixel count, total pixel count, and QR density, with fifteen URL-based features describing characteristics of the embedded URL. These features are combined into a 20-feature representation and processed for machine learning classification. Four supervised machine learning algorithms, namely Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine, are trained and compared using standard classification metrics. Random Forest is selected as the final classification model based on its overall performance. The system accepts QR codes through image upload or webcam scanning and provides QR code detection, perspective correction where required, decoding, feature extraction, and phishing classification. A Python Tkinter-based graphical user interface is developed to allow users to interact with the system and view the decoded URL, prediction result, and model performance comparison.

2. LITERATURE REVIEW

QR code-based phishing, commonly referred to as quishing, has received increasing attention as QR codes are widely used to provide access to online resources. Rafsanjani et al. [2] proposed QsecR, a secure QR code scanner based on malicious URL detection. Their work focused on identifying malicious URLs associated with QR codes and demonstrated the importance of analyzing the destination URL as part of QR code security. The study developed a security-oriented QR scanning framework using multiple categories of features for malicious URL detection.

Trad and Chehab [4] investigated quishing detection using machine learning by directly analyzing QR code structure and pixel patterns without extracting the embedded content. Their study evaluated several machine learning models, including Logistic Regression, Decision Tree, Random Forest, Naive Bayes, LightGBM, and XGBoost. The best-performing model achieved an AUC of 0.9106, which was further improved to 0.9133 after feature refinement. The study demonstrated that QR code visual and structural characteristics can provide useful information for detecting potentially malicious QR codes.

Several studies have focused specifically on detecting malicious or phishing URLs using machine learning and lexical features. Joshi et al. [1] proposed a machine learning approach based on static lexical features extracted directly from URL strings. Their approach avoided crawling the destination websites or relying on blacklist lookups, allowing faster detection. The evaluation reported an average accuracy of 92% and an average AUC of 0.98 across multiple testing sets, demonstrating that lexical URL characteristics can provide useful information for malicious URL classification.

Raja et al. [10] presented a lightweight malicious URL detection method using lexical features and machine learning techniques. The study compared different classifiers and reported that Random Forest performed better than the other evaluated classifiers in terms of accuracy. The findings support the use of URL-based feature extraction and machine learning as an effective approach for distinguishing malicious URLs from legitimate URLs.

Rashid et al. [3] investigated the generalization of phishing URL detection using unsupervised domain adaptation. Their study highlights the challenge of maintaining phishing URL detection performance when the characteristics of URLs change across different domains or datasets. This supports the importance of evaluating phishing detection models on diverse data and considering their ability to generalize to changing phishing patterns.

Tamal et al. [6] investigated phishing attack detection using feature vectorization and supervised machine learning techniques. Their work demonstrates the importance of selecting and representing useful features for improving malicious URL classification performance. The study further supports the application of supervised machine learning for identifying suspicious phishing-related characteristics.

Mossano et al. [7] presented SEQR, a usable secure QR code scanner designed to improve QR code security and user interaction. Their work demonstrates the importance of integrating security analysis into QR code scanning systems and highlights usability as an important consideration when developing secure QR code applications.

Charan and Thilak [5] investigated the detection of phishing links and QR codes related to UPI transactions using machine learning. Their work demonstrates the application of machine learning to QR-code-related phishing detection in a transaction-oriented context and further supports the need for automated analysis of QR codes and their associated links.

Reyes-Dorta et al. [9] studied malicious URL detection using machine learning and demonstrated the usefulness of machine learning techniques for identifying malicious URLs. Their work further supports the use of URL characteristics as input for automated malicious URL classification.

Aljabri et al. [8] provided a review of machine learning techniques for detecting malicious URLs and discussed existing approaches, challenges, and research directions in this area. Their review highlights the continued importance of feature-based machine learning methods for malicious URL detection and provides a broader research background for the present study.

The existing literature therefore demonstrates two important directions in phishing detection. QR-focused research has shown that visual and structural characteristics of QR codes can provide useful information for identifying quishing attacks, while URL-focused research has established that lexical and structural characteristics of URLs can be effectively used for malicious URL detection. These approaches provide complementary information: QR image features describe characteristics of the QR code itself, whereas URL features describe properties of the destination encoded within the QR code.

Based on these observations, the present study combines both types of information in a single representation. Five QR image features and fifteen URL-based features are extracted and combined to form a 20-feature dataset. Four machine learning algorithms—Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine—are evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. This approach enables the study to examine whether combining QR image characteristics with embedded URL characteristics provides an effective basis for QR code phishing detection.

3. RESEARCH GAP AND OBJECTIVES

3.1 Research Gap

Existing studies have investigated malicious and phishing URL detection using lexical and structural URL features, while QR-focused studies have examined QR code characteristics and machine learning approaches for detecting potentially malicious QR codes. These studies demonstrate the usefulness of analyzing either the QR code characteristics or the embedded URL for security analysis. However, there is scope to combine information from both the QR code image and its embedded URL within a single feature representation for QR code phishing detection.

The present study addresses this gap by combining five QR image features with fifteen URL-based features to form a 20-feature representation. Multiple machine learning algorithms are trained and compared using standard classification metrics to identify a suitable model for the proposed QR code phishing detection system.

3.2 Objectives

The main objectives of the proposed system are:

1. To generate QR codes from a labeled dataset of legitimate and phishing URLs for developing the detection system.
2. To extract five image-based features from QR codes and fifteen URL-based features from the decoded URLs.
3. To combine the extracted QR and URL features into a 20-feature representation and preprocess the data for machine learning classification.
4. To train and compare Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine models using accuracy, precision, recall, F1-score, and ROC-AUC.
5. To evaluate model consistency using 5-fold cross-validation and select the most suitable model based on overall performance.
6. To develop a Python Tkinter-based graphical user interface that supports QR image upload, webcam scanning, QR decoding, feature processing, and phishing classification.
7. To provide model performance comparison through the graphical user interface and present the QR code analysis and prediction results to the user.

4. PROPOSED METHODOLOGY

The proposed methodology consists of a sequence of steps for detecting phishing QR codes by analyzing both the QR code image and the URL encoded within it. A dataset containing 5,000 labeled URL samples is used, consisting of 2,515 legitimate samples and 2,485 phishing samples. Corresponding QR codes are generated from these URLs and used for feature extraction.

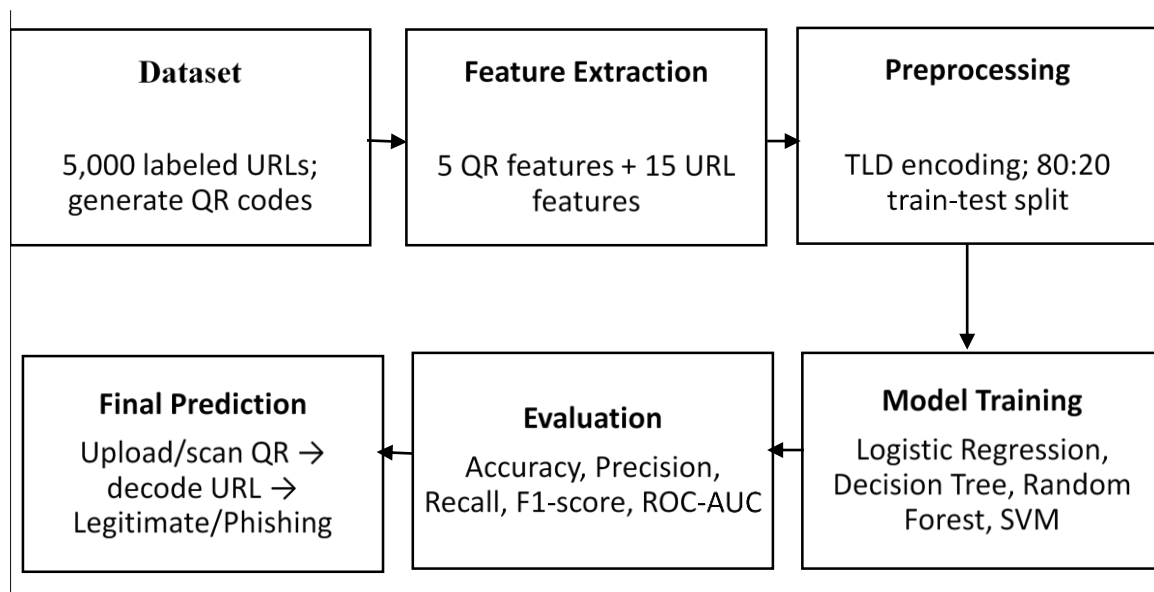
First, QR code images are processed to extract five image-based features: Width, Height, Black Pixels, Total Pixels, and QR Density. The URL encoded within each QR code is then decoded, and fifteen URL-based features are extracted, including URL Length, Domain Length, HTTPS, Number of Dots, Number of Hyphens, Number of Digits, Number of Slashes, Special Characters, Number of Subdomains, Path Length, Query Length, TLD, IP Address, URL Shortener, and Suspicious Keywords.

The five QR features and fifteen URL features are combined to form a 20-feature dataset. During preprocessing, the TLD feature is converted into numerical form using label encoding. The prepared dataset is divided into training and testing sets using an 80:20 split. Four supervised machine learning algorithms—Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine—are trained using the combined feature set.

The trained models are evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. In addition, 5-fold cross-validation is performed to evaluate the consistency of the model performance. Based on the overall evaluation results, Random Forest is selected as the final classification model.

For prediction, the system accepts a QR code through image upload or webcam scanning. For webcam input, the QR code is detected and perspective-corrected before further processing. The QR code is decoded to obtain the embedded URL, after which the same five QR features and fifteen URL features used during training are extracted. The features are combined, preprocessed using the same TLD encoding approach, and provided to the trained Random Forest model for classification as Legitimate or Phishing.

The complete methodology is integrated into a Python Tkinter-based graphical user interface. The GUI allows users to upload or scan QR codes, view the decoded URL, obtain the classification result, and view model performance comparisons. Supported form-based QR codes are handled separately, and website phishing detection is not applied to them.



5. DESCRIPTION OF THE SYSTEM

The proposed project, **QR Code Phishing Detection Using QR and URL Feature Extraction with Machine Learning**, is designed to identify whether a QR code is legitimate or potentially phishing by analyzing both the QR code image and the URL encoded within it. The system addresses the security concern that users cannot directly determine the nature of the destination URL from the visual appearance of a QR code.

The project uses a dataset containing **5,000 labeled URL samples**, consisting of **2,515 legitimate samples and 2,485 phishing samples**. QR codes are generated from the collected URLs, and the corresponding QR code images are stored for further processing. The QR images and their embedded URLs are then used to extract the features required for machine learning classification.

For each QR code, **five QR image features** are extracted: Width, Height, Black Pixels, Total Pixels, and QR Density. In addition, **fifteen URL-based features** are extracted from the decoded URL. These include URL Length, Domain Length, HTTPS, Number of Dots, Number of Hyphens, Number of Digits, Number of Slashes, Special Characters, Number of Subdomains, Path Length, Query Length, TLD, IP Address, URL Shortener, and Suspicious Keywords. The extracted features are combined to form a **20-feature representation** for classification.

The combined dataset is preprocessed before machine learning. The TLD feature is converted into numerical form using label encoding, and the dataset is divided into training and testing sets using an **80:20 split**. Four machine learning algorithms—**Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine**—are trained and compared. Their performance is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. Five-fold cross-validation is also performed to assess the consistency of the models. Based on the overall results, **Random Forest** is selected as the final classification model.

The final prediction system allows users to provide a QR code through **image upload or webcam scanning**. For webcam input, the QR code is detected and perspective-corrected where required before decoding. The embedded URL is extracted, and the same QR and URL features used during model training are generated. After preprocessing, the features are provided to the trained Random Forest model, which classifies the QR code as **Legitimate or Phishing**.

A **Python Tkinter-based graphical user interface** is developed to provide an accessible interaction with the system. The interface supports QR code upload, webcam scanning, QR detection and decoding, URL display, phishing classification, and model performance comparison. The system also handles supported form-based QR codes separately, where normal website phishing detection is not applied.

6. EVALUATION METRICS

The proposed system uses four supervised machine learning algorithms to classify QR codes as legitimate or phishing based on the combined QR image and URL feature set. The selected algorithms are Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine. Their performance is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC.

6.1 Logistic Regression

Logistic Regression is a supervised machine learning algorithm used for binary classification. It estimates the probability of a sample belonging to a particular class and assigns the sample to the corresponding class. In this project, Logistic Regression is trained using the 20 combined QR and URL features to classify QR codes as legitimate or phishing.

6.2 Decision Tree

Decision Tree is a supervised machine learning algorithm that performs classification through a sequence of decision rules based on the input features. The data is divided into branches according to feature conditions, and the resulting decision determines the predicted class. In this project, the Decision Tree model is trained using the combined QR and URL feature set.

6.3 Random Forest

Random Forest is an ensemble machine learning algorithm that combines multiple decision trees for classification. The predictions from the individual trees are combined to obtain the final classification. In this project, Random Forest is trained using the 20 extracted QR and URL features. It achieved the best overall performance among the four evaluated models and was selected as the final classification model.

6.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification. It identifies a decision boundary that separates samples belonging to different classes. In this project, SVM is trained using the combined QR and URL features to distinguish between legitimate and phishing QR codes.

6.5 Evaluation Measures

Accuracy measures the proportion of correctly classified samples among all samples.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Precision measures the proportion of correctly identified phishing samples among all samples predicted as phishing.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Recall measures the proportion of actual phishing samples that are correctly identified by the model.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

F1-Score represents the harmonic mean of precision and recall.

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

ROC-AUC measures the ability of the classification model to distinguish between legitimate and phishing samples across different classification thresholds. A higher ROC-AUC value indicates better discrimination between the two classes.

Where TP represents True Positive, TN represents True Negative, FP represents False Positive, and FN represents False Negative.

Machine Learning Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
Logistic Regression	87.90	88.98	87.06	88.01	92.52
Decision Tree	92.80	93.11	92.75	92.93	92.80
Random Forest	94.90	97.71	92.16	94.85	98.61
Support Vector Machine	89.10	90.18	88.24	89.20	92.27

7. SYSTEM IMPLEMENTATION

The proposed QR Code Phishing Detection System is implemented using Python and provides a graphical interface for QR code analysis and phishing classification. The implementation integrates QR code processing, feature extraction, machine learning classification, and result presentation into a single system.

7.1 QR Code Input and Detection

The system accepts QR codes through two input methods: image upload and webcam scanning. For image input, the user can select a QR code image through the graphical user interface. For webcam input, the system captures the QR code using the camera and detects the QR code from the captured frame. Where required, perspective correction is applied to the detected QR code before decoding.

7.2 QR Code Decoding and URL Extraction

After the QR code is detected, the system decodes the QR code to obtain the embedded URL. The decoded URL is displayed in the graphical user interface and is used for extracting the URL-based features required by the classification model.

7.3 Feature Extraction

The system extracts five features from the QR code image: Width, Height, Black Pixels, Total Pixels, and QR Density. Fifteen additional features are extracted from the decoded URL: URL Length, Domain Length, HTTPS, Number of Dots, Number of Hyphens, Number of Digits, Number of Slashes, Special Characters, Number of Subdomains, Path Length, Query Length, TLD, IP Address, URL Shortener, and Suspicious Keywords.

The extracted QR and URL features are combined to form the 20-feature input required by the trained machine learning model. The TLD feature is encoded using label encoding before the feature set is provided to the classifier.

7.4 Phishing Classification

The processed 20-feature representation is provided to the trained Random Forest model. The model classifies the QR code as either **Legitimate** or **Phishing**. The prediction result is displayed through the graphical user interface.

7.5 Graphical User Interface

A Python Tkinter-based graphical user interface is developed to provide user interaction with the system. The interface supports QR code image upload, webcam scanning, QR code detection, URL decoding, feature processing, phishing prediction, and model performance comparison. The interface also displays the decoded URL and the final classification result.

The system provides separate handling for supported form-based QR codes. For these QR codes, the system displays the corresponding form-related message and does not apply the normal website phishing detection process.

8. RESULTS AND DISCUSSIONS

The proposed QR Code Phishing Detection System was evaluated using the extracted QR image and URL features with four machine learning algorithms: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine. The models were compared using accuracy, precision, recall, F1-score, and ROC-AUC to determine their classification performance.

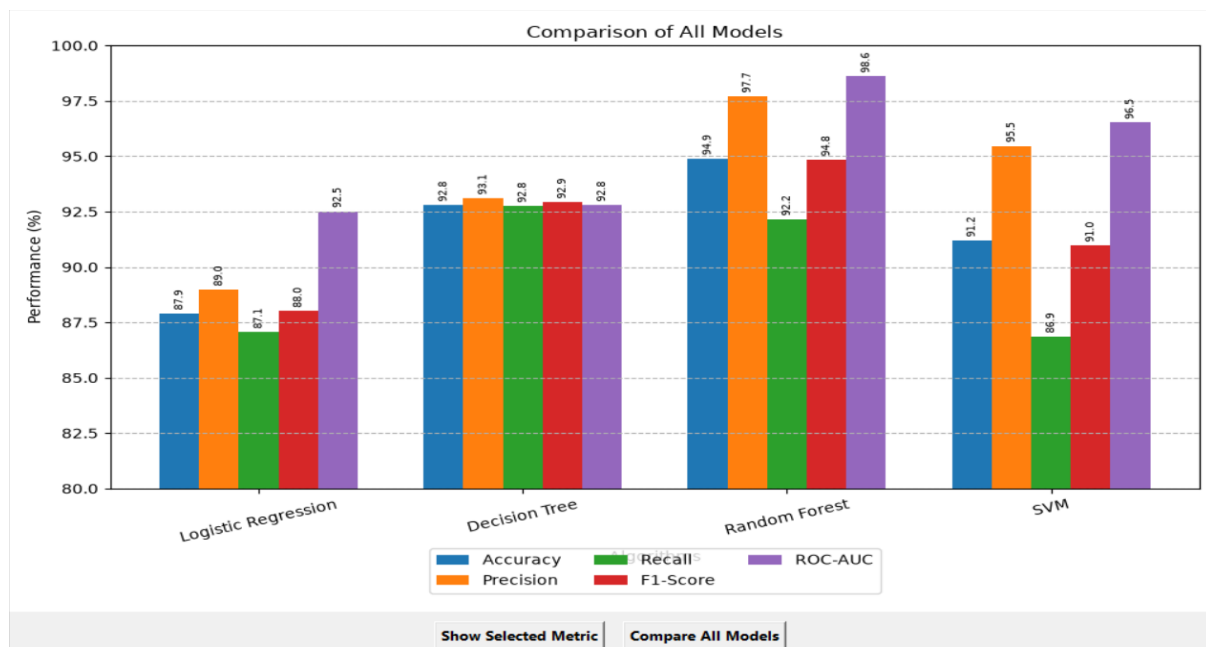
The experimental results show that Random Forest achieved the best overall performance among the four evaluated models. The Random Forest model achieved an accuracy of **94.90%**, precision of **97.71%**, recall of **92.16%**, F1-score of **94.85%**, and ROC-AUC of **98.61%**. Based on these results, Random Forest was selected as the final classification model for the proposed system.

The Random Forest model was further evaluated using 5-fold cross-validation. The fold accuracies obtained were **94.20%**, **95.20%**, **95.30%**, **95.90%**, and **94.70%**, resulting in a mean accuracy of **95.06%**. The standard deviation was approximately **0.00575**, indicating consistent performance across the validation folds.

The model comparison is presented in a table containing the accuracy, precision, recall, F1-score, and ROC-AUC values of all four machine learning models. A grouped bar chart provides a visual comparison of the models across the five evaluation measures. The comparison helps identify the model that provides the most suitable overall performance for QR code phishing classification.

The final Random Forest model is integrated into the prediction system. When a QR code is uploaded or scanned using the webcam, the system detects and decodes the QR code, extracts the same QR and URL features used during training, performs the required preprocessing, and provides the features to the trained model. The prediction is then displayed as either **Legitimate** or **Phishing** through the Tkinter-based graphical user interface.

The results demonstrate that combining QR image characteristics with URL-based characteristics provides an effective feature representation for QR code phishing detection. The obtained Random Forest performance also demonstrates the suitability of the selected machine learning approach for the implemented system.



9. CONCLUSION

The proposed QR Code Phishing Detection System demonstrates the application of machine learning for identifying phishing QR codes by analyzing both the QR code image and the URL embedded within it. The system addresses the security problem of QR codes hiding their destination URLs, which can make it difficult for users to determine whether a QR code is safe before accessing the linked website.

The system uses a Kaggle dataset containing URLs and their corresponding labels. QR codes are generated from the collected URLs, and features are extracted from both the QR images and the URLs. Five QR code features, namely Width, Height, Black Pixels, Total Pixels, and QR Density, along with fifteen URL-based features are extracted. These features are combined to form a 20-feature dataset. The data is preprocessed, including TLD encoding, and divided using an 80:20 train-test split. Four machine learning algorithms—Decision Tree, Random Forest, Support Vector Machine, and Logistic Regression—are trained and compared.

Among the four models, Random Forest achieved the best overall performance and was selected as the final classification model. It achieved an accuracy of **94.90%**, precision of **97.71%**, recall of **92.16%**, F1-score of **94.85%**, and ROC-AUC of **98.61%**. The model was also evaluated using 5-fold cross-validation, with a mean accuracy of **95.06%**, demonstrating consistent classification performance.

The final prediction system accepts a QR code through image upload or webcam scanning. For webcam input, the QR code is detected and perspective-corrected before further processing. The QR code is then decoded to obtain the embedded URL, and the same QR and URL features used during training are extracted. These features are combined and provided to the trained Random Forest model, which classifies the QR code as either Legitimate or Phishing.

A Python Tkinter-based graphical user interface was developed to provide a simple and user-friendly interaction with the system. The GUI supports QR image upload, webcam scanning, URL display, phishing prediction, and model performance comparison. This allows users to perform QR code analysis without manually executing the individual processing steps.

In conclusion, the proposed system demonstrates that combining QR image features with URL features can provide an effective approach for QR code phishing detection. The system provides an additional security layer by analyzing the QR code and its embedded URL before the user proceeds to the destination website.

10. FUTURE WORK

The proposed QR Code Phishing Detection System provides an effective machine learning-based approach for identifying phishing QR codes by combining QR code image features and URL features. However, several enhancements can be introduced in the future to further improve the accuracy, security, usability, adaptability, and practical application of the system.

One important enhancement is to expand and continuously update the dataset with a larger and more diverse collection of legitimate and phishing URLs. Including newly emerging phishing URLs can help the model recognize evolving phishing patterns. The feature extraction process can also be enhanced by incorporating additional QR code and URL characteristics to provide more useful information for classification. Future studies can further evaluate advanced machine learning techniques such as Gradient Boosting, XGBoost, neural networks, and ensemble methods and compare their performance with the existing models.

Another useful enhancement is the integration of real-time URL security information. The decoded URL could be checked against trusted threat-intelligence databases, phishing databases, or website reputation services. Combining machine learning predictions with external security information could provide an additional layer of protection against newly created malicious websites. The system can also be enhanced with explainable detection, where users are provided with information about suspicious URL characteristics and the factors that influenced the prediction.

Finally, the system can be extended from the current desktop-based Tkinter application to a web or mobile application for easier access on smartphones and other devices. Future versions can also support periodic model retraining using newly collected data and evaluate the model on different datasets to improve its adaptability and generalization. These enhancements can make the system more robust, adaptive, and practical for detecting evolving QR-based phishing attacks.

REFERENCES

- [1] Joshi, L. Lloyd, P. Westin, and S. Seethapathy, "Using Lexical Features for Malicious URL Detection—A Machine Learning Approach," *arXiv preprint arXiv:1910.06277*, 2019.
- [2] 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*, vol. 11, pp. 92523–92539, 2023, doi: 10.1109/ACCESS.2023.3291811.
- [3] F. Rashid, B. Doyle, S. C. Han, and S. Seneviratne, "Phishing URL Detection Generalisation Using Unsupervised Domain Adaptation," *Computer Networks*, vol. 245, Art. no. 110398, 2024, doi: 10.1016/j.comnet.2024.110398.
- [4] F. Trad and A. Chehab, "Detecting Quishing Attacks with Machine Learning Techniques Through QR Code Analysis," *arXiv preprint arXiv:2505.03451*, 2025.
- [5] G. R. Charan and K. D. Thilak, "Detection of Phishing Link and QR Code of UPI Transaction Using Machine Learning," in *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2023, doi: 10.1109/ICIMIA60377.2023.10426613.

-
- [6] M. A. Tamal, M. K. Islam, T. Bhuiyan, A. Sattar, and N. U. Prince, "Unveiling Suspicious Phishing Attacks: Enhancing Detection with an Optimal Feature Vectorization Algorithm and Supervised Machine Learning," *Frontiers in Computer Science*, vol. 6, Art. no. 1428013, 2024, doi: 10.3389/fcomp.2024.1428013.
- [7] M. Mossano, M. F. Veit, T. Lange, B. M. Berens, F. Sharevski, and M. Volkamer, "Development, Evaluation, and Implementation of SEQR—A Usable Secure QR Code Scanner," in *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*, Art. no. 1306, pp. 1–33, 2026, doi: 10.1145/3772318.3793213.
- [8] M. S. 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. 121395–121417, 2022, doi: 10.1109/ACCESS.2022.3222307.
- [9] N. Reyes-Dorta, P. Caballero-Gil, and C. Rosa-Remedios, "Detection of Malicious URLs Using Machine Learning," *Wireless Networks*, vol. 30, pp. 7543–7560, 2024, doi: 10.1007/s11276-024-03700-w.
- [10] S. Raja, R. Vinodini, and A. Kavitha, "Lexical Features Based Malicious URL Detection Using Machine Learning Techniques," *Materials Today: Proceedings*, vol. 47, no. 1, pp. 163–166, 2021, doi: 10.1016/j.matpr.2021.04.041.