WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

QR Code Analyser for Phishing Website Detection Using Machine Learning Algorithm

[1] K. Rajalakshmi, [2] Dr N. Subathra, [3] Dr M. Deepa Lakshmi

[1] II - M.Sc Computer Science, Reg. No: 24081206503912004, Sarah Tucker College, Tirunelveli – 7
[2] Assistant Professor, Department of Computer Science, Sarah Tucker College, Tirunelveli - 7
[3] Assistant Professor, Department of Computer Science, Sarah Tucker College, Tirunelveli - 7
[1] rajikumaran28@gmail.com, [2] subathra30@gmail.com, [3] mdeepaacademic@gmail.com

ABSTRACT

QR codes are frequently used to access websites and online services, hackers frequently use them to trick users by inserting phishing links. This project offers a QR Code Analyser that uses key URL-based features to analyse and extract the URL from a scanned QR code. The link is categorised as either malicious or safe using a Random Forest Machine Learning Model. To increase dependability, the system also cross-checks the connection with outside security services. The suggested method lowers the possibility of phishing attacks during routine scanning activities and assists users in identifying dangerous QR codes.

Keywords: Random Forest, Machine Learning, Phishing Detection, QR Code, and URL Features.

1. INTRODUCTION

QR (Quick Response) codes are commonly used today for making payments, opening websites, and sharing information quickly. They have gained popularity because they are fast, easy to scan, and convenient for everyday use. Many shops, apps, and services now rely on QR codes for simple and contactless access.

However, this convenience also brings a significant security issue. Attackers can create fake or harmful QR codes that contain phishing links. When users scan these codes, they might be redirected to unsafe websites that steal personal information, login details, or financial data. Since the link within a QR code is hidden until scanning, users often do not recognize the danger.

To address this problem, this project develops a QR Code Analyser that checks the safety of a QR code before opening the associated website. The system reads the QR code, extracts the URL, and analyses it using various URL-based features. A Random Forest Machine Learning Algorithm determines if the link is safe or malicious. The system also employs external security checks to improve detection accuracy.

This approach helps users avoid harmful QR codes and raises awareness about safe scanning practices in their daily digital lives.

2. LITERATURE SURVEY

Recently, Machine Learning-based techniques have drawn interest for enhancing the identification of phishing or malicious URLs contained within QR codes. In order to prevent users from being sent to dangerous websites, contemporary methods concentrate on feature extraction, anomaly detection, and classification models that can recognise dangerous QR scans. Important contributions in this field are presented in the following studies.

Yuan et al. [1] introduced a joint learning framework for identifying harmful URLs and showed better detection results, especially for phishing links in QR codes. Borkin et al. [2] looked at the effect of data normalization on model accuracy and stressed its importance for improving URL classification efficiency. Aljabri et al. [3] assessed various Machine Learning techniques for malicious URL detection and pointed out the importance of good feature selection and classification strategies.

Some efforts have specifically targeted phishing detection through URL analysis. DR, Patil, and Mohana [4] studied malware URL detection and found that structured URL features can help achieve accurate classification using Machine Learning methods. Mukherjee et al. [5] created a phishing detection model using a Random Forest classifier and reported better predictive performance. Abdulrahman et al. [6] investigated QR codes from a cybersecurity standpoint and discussed how attackers embed harmful redirects, which underscores the need for link verification before access.

Further research has aimed at improving URL analysis and QR-based phishing prevention. Sharma [7] addressed vulnerabilities in QR code usage and reviewed malicious QR attack techniques. Harrison et al. [8] proposed Machine Learning-based methods for phishing URL detection and achieved high

recognition rates. Sahoo et al. [9] conducted a thorough survey on phishing detection using machine learning and outlined challenges in URL-based classification. Mukherjee et al. [10] further examined phishing URL behaviour and showed better detection accuracy using Random Forest techniques.

3. METHODOLOGY

The proposed system aims to find phishing and harmful URLs hidden in QR codes before users click on them. Rather than opening the scanned link directly, the system inspects the QR code through a clear and organized workflow. Each part has a specific role in the detection process, which makes the entire system efficient, easy to understand, and ready for integration with Machine Learning Classification methods. The proposed system consists of **five main modules** that handle end-to-end QR code analysis and phishing URL detection. These modules are listed below:

1. QR Code Scanning Module.
2. URL Pre-Processing Module.
3. Feature Extraction Module.
4. Machine Learning Classification Module.
5. Result and Alert Module.

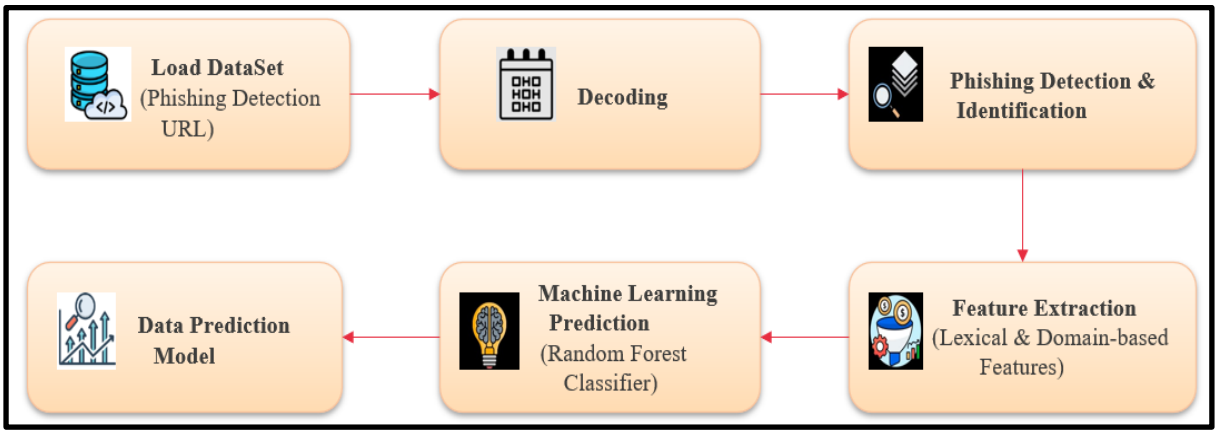

Figure 3.1: System Architecture of the QR Code Phishing Detection Model

The QR code based Phishing Detection Model process is shown in Figure 3.1. The process includes dataset loading, URL decoding, feature extraction, and Machine Learning based classification, which together enables the system to distinguish between legitimate and phishing URLs effectively.

3.1 QR Code Scanning Module

The QR Code Scanning Module reads the QR code submitted by the user. It applies image decoding techniques to extract the URL embedded in the QR code. Once the extraction is successful, the link is sent to the next modules for verification. This method helps users avoid accidentally opening unverified or harmful links.

3.2 URL Preprocessing Module

After extracting the URL, it goes through preprocessing to eliminate unnecessary characters and standardize its format. This module ensures the URL is clean and consistent before further checks. Proper preprocessing improves the detection process's effectiveness and aids in accurate feature extraction.

3.3 Feature Extraction Module

The Feature Extraction Module identifies important characteristics from the pre-processed URL. These characteristics include the URL length, presence of special characters, number of digits, domain structure, and any suspicious keywords. The features captured reflect the behaviour of the URL and serve as input for the classification model.

3.4 Machine Learning Classification Module

In this module, the extracted features are assessed using a Random Forest classification algorithm. The model is trained on labelled datasets that include both legitimate and phishing URLs. Random Forest is chosen for its strength, high detection accuracy, and capability to handle complex feature interactions. Based on the classification results, the URL is marked as either safe or harmful.

3.5 Result and Alert Module

The Result and Alert Module deliver the final decision to the user. If the analysed QR code contains a phishing or harmful URL, it shows a warning message to stop access. If the URL is deemed safe, the user can proceed. This module is essential for improving user awareness and minimizing the risk of phishing attacks.

4. EXPERIMENTAL RESULTS

The proposed QR Code Analyzer for detecting phishing websites was tested using a Kaggle dataset of Benign and Malicious QR codes. This dataset includes both legitimate and phishing URLs gathered from trustworthy sources. In this work, the dataset for phishing URL detection was divided into training and testing phases to evaluate model performance. The dataset was split in a 70:30 ratio where 70% of the data used to train the Random Forest model, while the remaining 30% was used for testing. This data split provides a reliable evaluation and reduces overfitting while keeping enough samples for both phases.

Dataset Description

Table 4.1: Benign and Malicious QR codes Details

Parameter	**Description**
Total No. of URLs	11,430
Legitimate URLs	5,715
Phishing URLs	5,715
Data Type	URL Text Data

The above given Table 4.1 clearly shows that using a balanced dataset helps the model to learn the patterns of phishing and legitimate URLs without bias during classification.

Performance Evaluation Metrics

The performance evaluation metrics were used to measure how effective the proposed phishing URL detection model is. Four standard metrics were used: Accuracy, Precision, Recall, and F1-Score. These metrics help show how well the model classifies legitimate and phishing URLs based on its predictions.

- **True Positive (TP):** It represents the count of correctly classified malicious URLs.
- **True Negative (TN):** It denotes the number of benign URLs correctly identified.
- **False Positive (FP):** It signifies benign URLs incorrectly classified as malicious.
- **False Negative (FN):** It indicates malicious URLs mistakenly classified as benign.

Accuracy

Accuracy, denoted in Eq. 1, shows the number of correctly classified URLs compared to the total number of samples. It reflects how correct the model is overall.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision

Precision, denoted in Eq. 2, refers to the number of correctly detected phishing URLs out of all URLs predicted as phishing. It indicates how the phishing predictions are trustworthy.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall

Recall, denoted in Eq. 3, measures the number of correctly identified phishing URLs compared to all actual phishing URLs in the dataset. It shows the model's ability to find the phishing attempts.

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

F1-Score

F1-Score, denoted in Eq. 4, shows the average of Precision and Recall. It offers a balanced measure of performance, especially when the distribution of classes is uneven.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Table 4.2: Performance Metrics

Metric	**Value (%)**
Accuracy	96.8
Precision	95.9
Recall	96.2
F1-Score	96.0

The results of the Performance Metrics in Table 4.2 show that the system achieves high accuracy in detecting phishing URLs within QR codes. The balanced precision and recall values indicate that the model successfully identifies phishing websites while reducing false classifications.

Comparative Analysis

To further evaluate the performance, the proposed model was compared with existing URL detection techniques. The comparative analysis results are summarized in Table 4.3, where the proposed Random Forest-based model is compared against traditional classifiers such as Naive Bayes, Support Vector Machine, and Decision Tree.

Table 4.3: Comparative analysis

Method	**Accuracy (%)**
Naive Bayes	89.4
Support Vector Machine (SVM)	92.1
Decision Tree	93.6
Proposed Random Forest Model	96.8

This comparison shows that the Random Forest-based QR phishing detection model performs better than traditional classifiers in terms of accuracy.

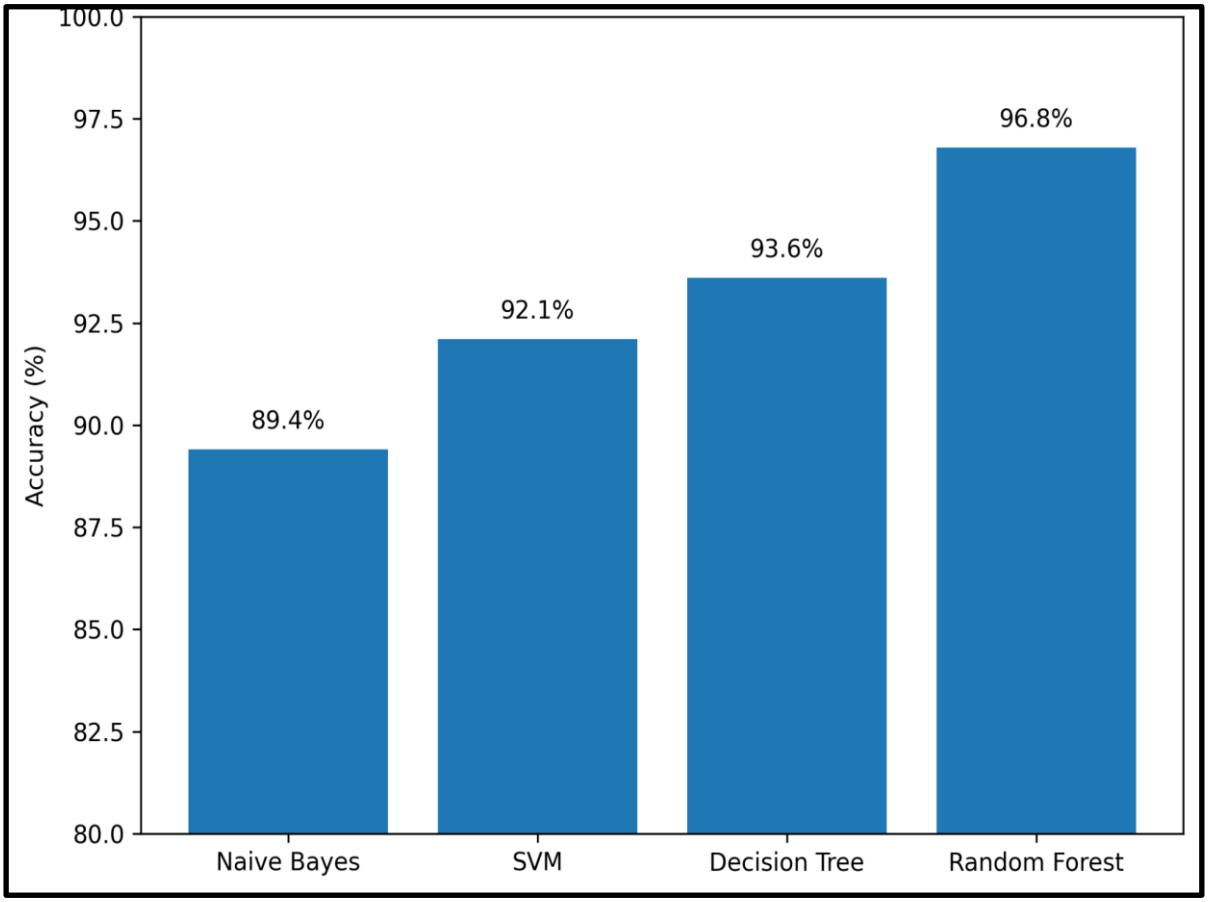

Figure 4.1: Comparative analysis of Phishing Detection Model

The graphical representation of the comparative analysis of the Phishing Detection Model is shown in Figure 4.1. The results indicate that the proposed model achieves consistently high performance across all evaluation metrics, demonstrating its effectiveness in detecting phishing websites through QR codes.

Overall, the experimental results demonstrate that the proposed QR Code Analyzer is a reliable solution for detecting phishing websites. The system effectively identifies malicious QR code URLs and improves security during QR code scanning.

5. CONCLUSION

This work presented a QR Code Analyzer for detecting phishing websites using a machine learning approach. The proposed system extracts URLs from QR codes and analyses them using lexical and domain-based features. A Random Forest classifier identifies whether the embedded URL is legitimate or phishing. Experimental results show that the proposed system achieves high detection accuracy with low misclassification rates. The combined use of feature extraction and ensemble learning allows for effective identification of phishing websites before users access them. This system offers a reliable and practical solution to improve security during QR code scanning. Overall, the proposed approach reduces phishing attacks from malicious QR codes and boosts user awareness and safety in real-world digital environments.

References

[1] Y. Yuan, Y. Liu, and Y. Lu, "A novel approach for malicious URL detection based on joint model," *Security and Communication Networks*, vol. 2021, pp. 1–12, 2021.

[2] D. Borkin, N. Nemtseva, G. Michal'čák, and K. Maiorov, "Impact of data normalization on classification model accuracy," *Res. Papers Fac. Mater. Sci. Technol.*, vol. 27, no. 45, pp. 79–84, 2019.

[3] K. Aljabri, N. Alotaibi, A. Alnufayli, M. Alharbi, and M. A. Mohammed, "Detecting malicious URLs using machine learning techniques: Review and research directions," *IEEE Access*, vol. 9, pp. 121395–121417, 2021.

[4] U. S. DR, R. Patil, and M. Mohana, "Malicious URL detection and classification analysis using machine learning models," *Proc. Int. Conf. Intelligent Data Communication Technologies and IoT*, pp. 470–476, 2023.

[5] P. D. Mukherjee, P. Das, A. Gangopadhyay, and A. R. Chinta, "Random forest-based phishing URL detection," *Int. J. Comput. Appl.*, vol. 172, no. 3, pp. 1–7, 2020.

[6] A. Abdulrahman, A. Khider, and A. Hamad, "Quick response code and cyber security," *Int. J. Eng. Res. Technol.*, vol. 4, no. 7, pp. 1–5, 2017.

[7] M. Sharma, "Malicious QR code system design using vulnerable code structure," *Int. J. Comput. Appl.*, vol. 59, no. 11, pp. 1–6, 2017.

[8] W. Harrison, A. Siu, and Y. Xu, "Phishing URL detection with machine learning techniques," *Procedia Comput. Sci.*, vol. 195, pp. 114–123, 2021.

[9] D. Sahoo, C. Liu, and S. Hoi, "Phishing detection using machine learning approaches: A survey," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018.

[10] P. D. Mukherjee, A. Gangopadhyay, and A. R. Chinta, "Behavioural analysis of phishing URLs using random forest," *Int. J. Cyber Res. Educ.*, vol. 4, no. 2, pp. 50–60, 2020.