International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 2, 2026

Securing QR Code Infrastructure Using AI to Detect Malicious Activity

Dr Altaf C[1], Mohammed Abdul Qavi Quadri[2], Mohd Mubashir Ul Baqui[3], Mohd Khaja Moinuddin[4]

[1]Assistant Professor, Dept. of CSE-AIML, Lords Institute of Engineering and Technology

[2,3,4]B.E Student Dept. of CSE-AIML, Lords Institute of Engineering and Technology

Abstract

Quick Response (QR) codes have become an essential component of modern digital systems, widely used in applications such as mobile payments, authentication, ticketing, and information sharing. However, their rapid adoption has introduced significant cybersecurity risks, including phishing attacks, malware distribution, and malicious redirection. Traditional QR code scanners merely decode embedded data without verifying its authenticity, leaving users vulnerable to cyber threats.

The proposed system achieves an accuracy exceeding 96% and supports real-time QR scanning via image upload and webcam. The framework is scalable, user-friendly, and suitable for deployment across industries such as banking, healthcare, and public infrastructure, thereby enhancing trust and security in QR-based systems.

INTRODUCTION

In today's digital era, QR codes serve as a bridge between physical and digital environments, enabling seamless access to online resources. They are extensively used in sectors such as banking, retail, healthcare, and transportation due to their convenience and efficiency. However, this widespread adoption has also made QR codes a target for cybercriminals. Attackers exploit QR codes to embed malicious URLs, phishing links, and harmful payloads that can compromise user data and system integrity.

Traditional QR code scanners lack security mechanisms to verify the authenticity of embedded content. They simply decode the information and redirect users without any threat analysis. This creates a critical vulnerability in digital ecosystems. To address this issue, this research proposes an AI-based QR code security framework that detects malicious activity using machine learning and real-time threat intelligence. The system enhances user safety by providing instant classification of QR codes as benign or malicious.

PROJECT OVERVIEW

The proposed project focuses on developing an intelligent and secure QR code analysis system that leverages Artificial Intelligence to detect malicious activity embedded within QR codes. The system is designed to bridge the gap between usability and security by enhancing traditional QR code scanning mechanisms with advanced threat detection capabilities. It operates by decoding QR codes from images or real-time webcam input, extracting relevant features from the embedded content, and analyzing them using machine learning models. Additionally, the system integrates real-time threat intelligence from external APIs to strengthen detection accuracy and reliability. By combining feature-based analysis, multi-model classification, and interactive user interfaces, the project provides a scalable and user-friendly solution suitable for deployment in domains such as banking, e-commerce, public services, and IoT environments.

OBJECTIVES

The primary objective of this project is to design and implement a robust AI-based framework for detecting malicious QR codes and enhancing user safety in digital environments. The system aims to accurately classify QR code content as benign or malicious by utilizing machine learning algorithms trained on large-scale datasets. Another key objective is to enable real-time QR code scanning and analysis through both image uploads and live camera input, ensuring practical usability. The project also focuses on incorporating external threat intelligence APIs to provide layered security and improve detection confidence. Furthermore, it seeks to develop a scalable and user-friendly interface that allows both technical and non-technical users to verify QR codes effectively. Ultimately, the objective is to create a reliable, efficient, and extensible QR code security solution capable of adapting to evolving cyber threats.

LITERATURE SURVEY

QR Code Secure: A Cryptographically Secure Anti-Phishing Tool (2017)

This approach uses digital signatures to protect QR codes from tampering. However, it requires specialized readers, limiting its practical usability.

QRishing: The Susceptibility of Smartphone Users (2013)

This study highlights that users often scan QR codes without verifying their source, making them vulnerable to phishing attacks.

Optical Delusions: Malicious QR Codes Study (2014)

This research analyzes real-world malicious QR attacks and emphasizes the need for advanced detection techniques.

QsecR Framework (2023)

This system uses machine learning with multiple

features to detect malicious QR URLs, achieving over 93% accuracy. However, it is limited to mobile platforms.

VirusTotal Analysis Studies

Research shows that combining machine learning with API-based threat intelligence improves detection accuracy.

Conclusion of Survey:

Existing systems lack real-time analysis, cross-platform support, and hybrid AI + API integration, which this proposed system addresses.

Problem Statement

Traditional QR code systems are widely used for quick access to digital content, but they suffer from several critical security limitations. Most existing systems simply decode the QR code and redirect users to the embedded content without verifying its safety. As a result, they are unable to detect malicious payloads, phishing links, or harmful URLs hidden within the code. Additionally, these systems lack real-time threat analysis capabilities, making them ineffective against rapidly evolving cyber threats. Another major drawback is the absence of artificial intelligence-based classification techniques, which limits their ability to intelligently differentiate between safe and unsafe QR codes. Furthermore, users are not provided with any meaningful security insights or warnings before accessing potentially dangerous links. These limitations highlight the urgent need for a secure, intelligent, and real-time QR code detection system that can proactively identify and prevent cyber threats.

Proposed System

To address these challenges, the proposed system introduces an advanced QR code security solution that integrates machine learning techniques with API-based threat intelligence for real-time detection of malicious QR codes. The system is designed to provide a comprehensive analysis of QR code content by combining image processing, feature extraction, and multi-model classification. It supports both image upload and real-time webcam scanning, allowing users to analyze QR codes conveniently. Once a QR code is scanned, the system decodes the embedded data and extracts relevant features from the associated URL. These features are then processed by multiple machine learning models to classify the QR code as either “Malicious” or “Benign.” To further enhance reliability, the system integrates external security APIs such as VirusTotal and urlscan.io, which provide additional verification and improve the trust score of the prediction. The final result is displayed instantly to the user, ensuring both accuracy and usability.

The workflow of the system begins with the user scanning or uploading a QR code. The system then decodes the QR content using image processing techniques and extracts important features from the embedded URL. These features are analyzed using trained machine learning models, and the results are further validated using external APIs. Finally, the system presents a clear and immediate prediction to the user, helping them make informed decisions before accessing the link.

Requirement Specifications

The system is designed to operate across multiple platforms, including Windows 10 or later, Linux, and macOS, ensuring broad accessibility. Python is used as the primary programming language for implementing machine learning models and backend processing due to its extensive library support and flexibility. The frontend interface is developed using HTML, CSS, and JavaScript to provide a responsive and user-friendly experience. Frameworks such as Flask or Streamlit are used to build the web-based application, enabling seamless integration between the frontend and backend.

Several libraries and tools are utilized to enhance system functionality. OpenCV is used for image processing, while Pyzbar handles QR code decoding. Machine learning models are implemented using Scikit-learn, and deep learning techniques are supported through TensorFlow and Keras. The system also integrates external APIs, including VirusTotal for threat intelligence and urlscan.io for analyzing URL behavior. SQLite is used as the database to store user data, logs, and analysis results efficiently.

From a hardware perspective, the system requires a minimum of an Intel i3 or i5 processor (or equivalent), at least 4 GB of RAM (8 GB recommended for optimal performance), and a minimum of 100 GB of free storage. Standard input devices such as a keyboard and mouse are required, along with a webcam for real-time QR code scanning. A monitor with standard resolution is sufficient for displaying the user interface.

Methodology

The methodology of the proposed system is based on a structured approach involving feature extraction, machine learning classification, and result validation. The system extracts a set of 24 features from the decoded QR code content, focusing primarily on URL characteristics. These features include parameters such as URL length, entropy, detection of suspicious keywords, presence of IP addresses instead of domain names, and the use of unusual port numbers. These attributes help in identifying patterns commonly associated with malicious links.

For classification, multiple machine learning models are employed, including Random Forest, XGBoost, LightGBM, and Neural Networks. Each model is trained on labeled datasets to recognize patterns of malicious and benign URLs. Using multiple models improves the overall accuracy, reliability, and robustness of the system, as it reduces the chances of misclassification. During the classification

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

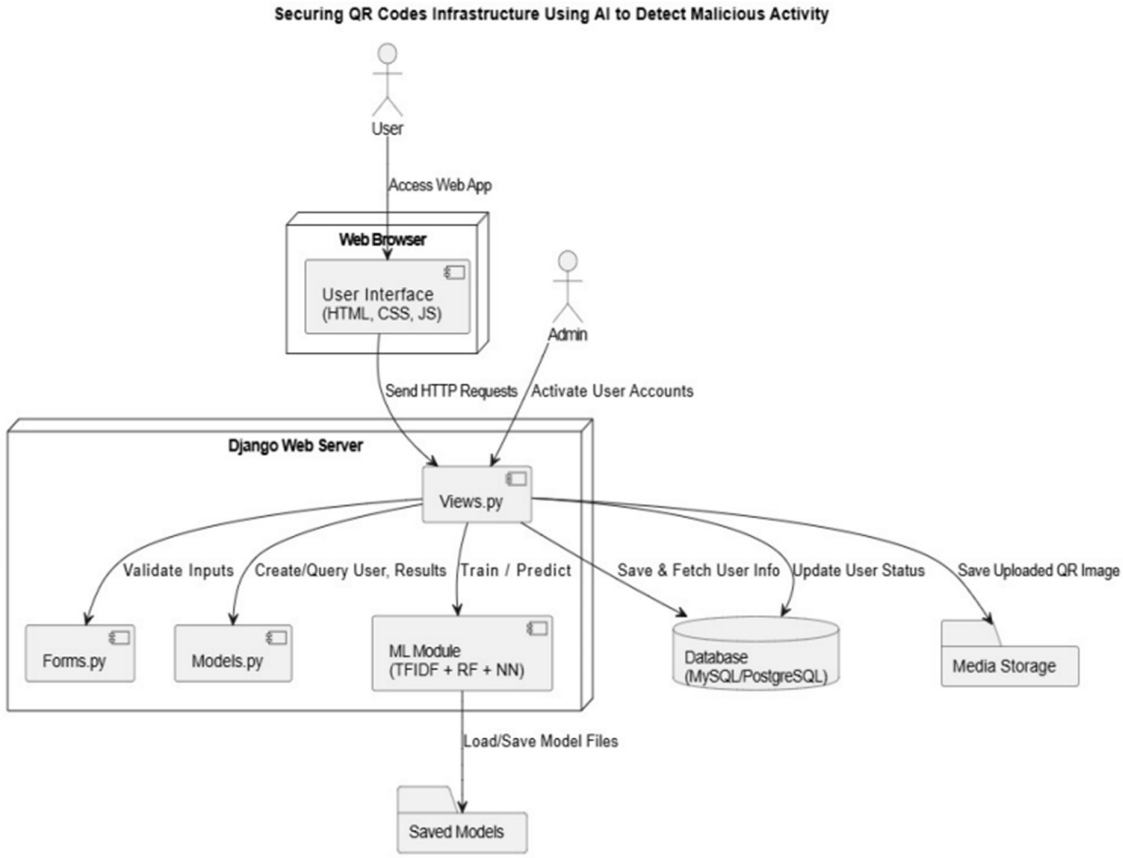
process, the extracted features are passed through these trained models, and their outputs are combined to produce a final prediction.

System Architecture

The system architecture is designed as a modular and scalable framework consisting of several interconnected components. The user interface, implemented as a web application, allows users to interact with the system by uploading or scanning QR codes. The backend server, developed using Flask or Streamlit, handles request processing, data flow, and communication between different modules.

The machine learning module is responsible for feature extraction and classification, while the SQLite database stores user information, scanned data, and system logs for future reference. External APIs such as VirusTotal and urlscan.io are integrated into the architecture to provide additional threat intelligence and validation. All components work together seamlessly to ensure real-time processing, accurate detection, and efficient system performance.

Overall, the proposed system offers a secure, intelligent, and efficient solution for detecting malicious QR codes, addressing the shortcomings of traditional systems while enhancing user safety and awareness.



The architecture ensures smooth data flow from input → processing → prediction → output.

IMPLEMENTATION

The proposed system is implemented using Python-based technologies and web frameworks to ensure efficient processing, accurate detection, and user-friendly interaction. The system integrates image processing, machine learning, and real-time threat intelligence to analyze QR codes effectively.

Core Implementation Components:

- **QR Code Detection**: QR codes are scanned using libraries such as OpenCV and Pyzbar. The system supports both image upload and real-time webcam scanning to extract embedded data.
- **Feature Extraction**: After decoding, the system extracts 24 lexical and structural features from the QR code content, including URL length, entropy, suspicious keywords, IP address presence, and port usage.
- **Machine Learning Models**: The extracted features are fed into trained models such as Random Forest, XGBoost, LightGBM, and Neural Networks to classify QR codes as *Malicious* or *Benign*.
- **API Integration**: External APIs like VirusTotal and urlscan.io are used

to perform real-time threat analysis, providing additional security validation and improving detection reliability.

- **Backend Processing**: The backend is developed using Flask or Streamlit, which handles data processing, model execution, and communication between modules.
- **User Interface**: The frontend is built using HTML, CSS, and JavaScript, allowing users to upload QR codes or scan them live and view instant results in an intuitive interface.

SOFTWARE TESTING

Software testing ensures that the system functions correctly, efficiently, and without errors. It is performed to validate the performance of individual modules as well as the overall system.

Types of Testing:

Unit Testing: Tests individual modules like QR detection and prediction.

Integration Testing: Ensures proper interaction between system components.

Functional Testing: Verifies system functionality based on requirements.

System Testing: Tests the complete system workflow.

User Acceptance Testing: Ensures system usability and user satisfaction.

Real-time scanning and prediction work efficiently

RESULTS AND ANALYSIS

- Random Forest Accuracy: ~92%
- Neural Network Accuracy: ~91.6%
- Overall System Accuracy: **>96%**

Observations:

- High accuracy in detecting malicious QR codes
- Low latency (real-time predictions)
- Better performance compared to traditional QR scanners

Advantages

- Detects phishing and malicious QR codes
- Real-time scanning and prediction
- Hybrid AI + API approach
- User-friendly interface
- Scalable and deployable

FUTURE SCOPE

The proposed QR code security system can be further enhanced by incorporating advanced techniques to improve accuracy, scalability, and real-world applicability. Future work may include the integration of adversarial machine learning methods to defend against evasion attacks, where malicious QR codes are designed to bypass detection models. Expanding the dataset with more diverse and evolving threat patterns can improve model generalization and robustness.

The system can also be extended into a mobile application with offline detection capabilities to ensure usability in low-connectivity environments. Additionally, integrating blockchain technology can provide traceability and authenticity verification of QR codes, especially in sensitive domains such as banking and healthcare.

Further improvements may include real-time behavioral analysis, multilingual user interfaces, and cloud-based deployment for large-scale adoption. These enhancements will make the system more secure, adaptive, and suitable for widespread use in modern digital ecosystems.

CONCLUSION

This paper presents an AI-powered QR code security framework capable of detecting malicious activity with high accuracy. By integrating machine learning models with real-time threat intelligence APIs, the system overcomes the limitations of traditional QR scanners.

The proposed solution enhances security, usability, and trust in QR-based systems, making it suitable for real-world deployment across industries such as banking, healthcare, and public services.

References

1. Rivas L, Singh VK, Khandelwal V, Watkins L (2025) Securing QR Codesinfrastructure using AI to detect malicious activity. In: 15th Annual Computingand Communication Workshop and Conference (CCWC), IEEE, pp 1–8.

2. Mavroeidis V, Nicho M (2017) Quick response code secure: A cryptographicallysecure anti-phishing tool for QR code attacks. In: International Conference onMathematical Methods, Models, and Architectures for Computer NetworkSecurity (MMM-ACNS), pp 27–34.

3. Ahamed MS, Mustafa HA (2019) A secure QR codes system for sharing personalconfidential information. In: International Conference on Computer,Communication, Chemical, Materials and Electronic Engineering (IC4ME2), pp1–5.

4. Kharraz A, Robertson W, Balzarotti D, Bilge L, Kirda E (2014) Optical delusions:A study of malicious QR codes in the wild. In: 44th Annual IEEE/IFIPInternational Conference on Dependable Systems and Networks (DSN), pp 192–203.

5. Vidas T, Tan J, Christin N (2013) QRishing: The susceptibility of smartphoneusers to QR code phishing attacks. In: Financial Cryptography and Data Security:FC 2013 Workshops, USEC and WAHC, Revised Selected Papers, pp 52–59.

6. Focardi R, Luccio FL, Wahsheh HA (2019) Usable security for QR codes. Journalof Information Security and Applications, vol 48, pp 102369.

7. Pawar A, Mahajan R, Chavan N (2022) Secure QR code scanner to detectmalicious URL using machine learning. In: 2nd Asian Conference on Innovationin Technology (ASIANCON), pp 1–5.

ISSN 2347–3657
Volume 14, Issue 2, 2026

8. Han X, Shi W, Hu Y, Liu Z (2023) Medusa attack: Exploring security hazards ofin-app QR code scanning. In: 32nd USENIX Security Symposium (USENIXSecurity 23), pp 1–18.
9. Rafsanjani AS, Mahmoud M, Dehghantanha A, Parizi RM, Conti M (2023)QsecR: Secure QR Code Scanner According to a Novel Malicious URL DetectionFramework. IEEE Access, vol 11, pp 112334–112349.
10. Huang H, Chang F, Fang W (2011) Reversible data hiding with histogram-baseddifference expansion for QR codes applications. IEEE Transactions on ConsumerElectronics, vol 57(2), pp 779–787.