International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 06 | Jun - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

QR Code Attack Detection for Mobile Security

Hindu S, Milon Sarah A, Shivathmika S, Sri Vibaashini V G, Anitha M
Hindu S[1], Milon Sarah A[2], Shivathmika S[3], Sri Vibaashini V G[4], Anitha M[5] ,

[1],[2],[3],[4],[5] ,Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women

---

Abstract - The widespread adoption of QR codes for mobile transactions has significantly increased user vulnerability to "quishing" (QR phishing) and malicious redirects. Traditional signature-based defenses fail against these visually opaque threats and zero-day exploits. To address this gap, this paper introduces a lightweight, machine-learning-driven detection framework optimized for resource-constrained mobile devices. By analyzing lexical structures and multi-stage redirection patterns in real-time . This research provides a highly scalable, proactive defense mechanism to secure mobile users against evolving QR-based cyber threats.

Key Words: QR Code Security, Quishing, Machine Learning, Mobile Security, Malicious Redirects.

1. INTRODUCTION

Quick Response (QR) codes have transitioned from simple industrial inventory-tracking tools into an indispensable foundation of modern digital communication, mobile transactions, and ubiquitous data sharing frameworks. Due to their rapid readability, high data storage capacity, and unmatched convenience, QR codes are now embedded inside digital online payments, electronic ticketing systems, dynamic physical advertisements, two-factor authentication loops, and contactless information delivery gateways. In an increasingly smartphone-centric world, these matrices serve as a vital physical-to-digital interface, allowing users to interact instantly with complex web applications by simply activating their device's built-in camera hardware.

Despite their extensive utility, the unique architectural nature of QR codes introduces critical cybersecurity vulnerabilities. Because the encoded alphanumeric dataset within a physical QR code matrix is visually opaque to a human observer prior to structural decoding, users cannot instinctively evaluate the destination target or safety parameters of the embedded link. This transparency gap allows malicious actors to execute "quishing" (QR phishing) vectors with high success rates. Attackers seamlessly overlay malicious QR codes onto legitimate public banners, digital payment points, or phishing emails. Once a user scans the compromised code, they are quietly redirected to fraudulent payment pages, credential-harvesting user interfaces, or background malware distribution servers, leading to systemic identity theft, device exposure, and financial losses.

Traditional QR code scanners built into mobile operating systems operate strictly as standard passive decoders. They focus entirely on parsing pixel alignments into text strings without executing proactive security profiling or real-time threat detection. Consequently, users are left exposed to zero-day malicious links and dynamic obfuscation tactics. To systematically resolve this vulnerability, this research paper introduces a standalone, real-time QR Code Attack Detection System for Mobile Security. Operating as a client-focused application layer, the framework parses decoded payloads on-device through an integrated rule-based analytical heuristic checking engine. By evaluating structural parameters, identifying domain anomalies, and serving instant color-coded visual threat warnings prior to browser rendering, the proposed framework hardens mobile environments against quishing variations while maintaining minimal execution latency.

2. LITERATURE SURVEY

To establish a comprehensive understanding of current defensive frameworks against malicious Quick Response (QR) codes, this section analyzes recent advancements and methodologies in the domain of quishing mitigation.

B-PhishQR (2026): Yalda et al. introduced a blockchain-based framework tailored for secure QR code verification. While the decentralized architecture provides high structural protection against malicious manipulation, the system demonstrates limited adaptability and remains fundamentally non-adaptive to novel vector transformations.

Advanced QR Code Attack Mitigation (2025): Jangal et al. leveraged deep learning models paired with Large Language Models (LLMs) to accurately parse embedded structural vectors. However, their architecture remains heavily data-dependent and resource-intensive, resulting in prominent latency constraints and susceptibility to overfitting.

QRiS (2025): Akram et al. designed a preemptive novel approach focused entirely on extracting structural features from QR shapes to detect anomalies prior to execution. Though optimized for immediate preprocessing, it struggles with diminished detection accuracy when processing low-quality or degraded camera captures.

Gen-AI Era Evaluation (2025): Thapa et al. explored the paradigm shift of utilizing quantized LLMs against classical machine learning models. Although the framework delivers deep intelligence metrics, the associated computational overhead introduces significant execution delays on consumer-grade mobile devices.

Self-Authenticating Modulation (2024): Barron & Sharma introduced a dual-modulation mechanism that embeds cryptographic authentication paths directly within the structural modules of the QR design. While robust, the system introduces deployment friction due to its dependence on universal system updates and image resolution baselines.

AI Exploration Strategies (2024): Vaithilingam et al. utilized artificial intelligence networks to categorize prospective web threats. The approach suffers from a direct dependency on centralized cloud lookups, leading to communication delays and processing bottlenecks.

QSecR Scanner (2023): Rafsanjani & Kamaruddin engineered a static feature-based classification scanner. While effective against historically logged malicious links, its rigid rule bounds render it incapable of adapting to dynamic or obfuscated zero-day URLs.

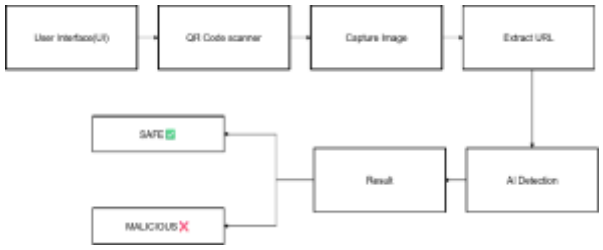

Fig -1: Overall Structural architecture and modular flow pipeline of the proposed system

3. PROPOSED METHODOLOGY

The proposed QR Code Attack Detection System is engineered as a lightweight client-side defense framework. It is designed to perform localized analysis without relying on heavy cloud server dependencies. The overall layout of the system is divided into four functional core modules.

A. System Architecture & Modular Breakdown

Module 1 : Input Interaction & Matrix Decoding:

This gateway layer manages primary data acquisition through dual ingestion paths. The system interfaces directly with mobile camera hardware configurations using the html5-qrcode JavaScript engine to process live video capture streams. Concurrently, it hosts an asynchronous static file-upload pipeline that leverages the jsQR parser library to analyze static images from local storage. Both routines scan raw matrix pixel grids, isolate alignment targets, perform binarization filters, and decode the geometric modules into an absolute plain text URL representation string.

Fig-2: User Interface modules for live camera tracking and file-upload data ingestion paths.

Module 2: URL Preprocessing & Heuristic Rule Analysis Engine:

Once the hidden destination string is extracted, it undergoes an automated structural cleaning sequence. This steps strips unnecessary tracking tracking blocks and handles redirection resolution to expose hidden paths. The cleaned string is passed to ai_engine.py, where it is systematically run through a 5-rule heuristic scoring engine to compute its security profile before any request hits the browser layer.

Module 3: Visual Security Dashboard & Administrative Controls:

This module operates as the central metrics workspace. It fetches background log tables from the database to render responsive statistical indicators, total scanning volume metrics, threat category count tracking models, and chronological activity grids via a dark-mode optimized canvas interface.

Fig-3: Visual Dashboard Interface for monitoring and threat category tracking

Module 4: Mitigation & Secure Result Presentation Layer:

This terminal module executes defensive workflow overrides. If the rule execution logs a safe profile, the application safe-loads the target site natively. If any rule flags a hazard condition, the routing engine stops automated browser redirection and displays a prominent visual notification card detailing the exact heuristic parameters triggered.

Fig-4: Secure Result Presentation Interface for displaying defensive workflow and visual hazard notifications.

B. Heuristic Rules Framework

The structural evaluation engine applies five custom classification rules to determine the threat profile of an incoming destination link:

Rule 1 (Insecure Protocol): If the URL protocol uses http:// instead of an encrypted layer (https://), it is classified as Malicious due to clear data-transit vulnerabilities.

Rule 2 (Lexical Volume Length): If the total string length is greater than 100 characters, it is classified as Malicious. Lengths between 60 and 100 characters are flagged as Suspicious, as long URLs are often used to conceal bad sub-domains.

Rule 3 (Literal Address Mapping): If the destination host string consists of a raw numerical IP address instead of a registered top-level domain name, it is classified as Suspicious.

Rule 4 (Token Match Detection): The engine searches the string array for known threat tokens (login, verify, phishing, virus, update). If matched, it triggers a Malicious classification.

Rule 5 (Micro-URL Shrinkage): If the link structure contains fewer than 15 total characters, it marks a redirect or URL shortener tool and is flagged as Suspicious.

C. System Implementation Parameters

To ensure low computing overhead on mobile devices, the system uses the specialized technology stack detailed in Table I.

Table -1: Technical Specifications Matrix

System Layer	Selected Technology	Operational Scope
Live Stream Capture	html5-qrcode	Processes real-time video feeds via mobile camera streams
Static File Decoding	jsQR Module	Parses pixels from local uploaded image files to fetch text data
Heuristic Evaluation	Python core script	Runs the 5-rule checking pipeline and safety filtering
User Interface Layout	HTML5, CSS3, JS	Renders data charts, user dashboards, and warning windows
Back-end Engine	Flask Core Framework	Routes API transactions and
		handles database storage pipelines
Data Layer	SQLite, SQLAlchemy	Stores operational scan history tables and system parameters

4. RESULTS AND DISCUSSION

The proposed framework was systematically validated across all three specified input options: live mobile camera streams, file image uploads, and manual URL text checks.

A. Threat Profiling Evaluation

During live environment execution, the system parsed data arrays consisting of safe, suspicious, and malicious target parameters. The custom heuristic classification core demonstrated accurate categorization, returning instant color-coded status states to user screens:

- Green (Safe Label): Generated for verified HTTPS routes displaying normal character allocations and no signature matches.
- Orange (Suspicious Label): Triggered by active link shorteners, raw numerical IP locations, or domain blocks containing character arrays longer than 60 indices.
- Red (Malicious Label): Prompted automatically by unencrypted standard HTTP endpoints or clear text matches with malicious system keywords.

Fig-5: Real-time threat alert validation interface isolating unencrypted malicious payloads and insecure network connections

B. Local Dashboard Analytics

Scanned records pass directly to transactional storage sheets managed by the SQLite infrastructure. The analytics application framework processes historical traffic patterns to update chart modules and security tracking statistics. Users can review threat categories, manage activity lists, and customize operational parameters through a dedicated settings tab.

The dashboard uses these stored transaction logs to update its visual components dynamically. It tracks overall scanning volume, updates threat

distribution charts, and lists chronological history logs. This background analytics system runs smoothly alongside active scanning, ensuring a responsive user experience. Users can also fine-tune the system via a settings panel, which includes toggling a dark mode layout and adjusting the sensitivity thresholds of the heuristic engine.

5. FUTURE SCOPE

While the current standalone web application framework delivers high detection accuracy and exceptionally low execution latency, several critical technological trajectories can be explored in subsequent development phases to significantly broaden its defensive coverage and harden consumer mobile environments globally:

1. Cross-Platform Client Integration and Universal Web Ecosystem Expansion:

   The core on-device heuristic checking architecture can be abstracted and recompiled into a lightweight, platform-agnostic background service module. Developing dedicated, native browser extension plugins for leading desktop and mobile web environments—such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari—would allow the system to inspect dynamic links globally across all browser navigation requests. Furthermore, by optimizing the underlying parsing logic, the system can be integrated across enterprise-scale Internet of Things (IoT) terminal displays, smart automated retail Point-of-Sale (PoS) registers, digital ticketing gates, and public digital information kiosks. Expanding coverage to these devices will create a secure infrastructure at the physical points where users scan codes.

2. Edge-AI Integration via Machine Learning Model Quantization:

   To move beyond fixed static rule checks, future versions will integrate quantized machine learning classifiers (such as highly compact Random Forest or Support Vector Machine models). By compressing these models using edge-AI optimization pipelines, the system can run them directly on-device without network dependence. This will enable the application to recognize complex, evolving phishing URL patterns based on semantic characteristics rather than strict string rules.

3. Advanced Dynamic Risk Scoring Engines:

   The binary and ternary classification flags can be replaced with a highly granular, multi-factor numeric hazard metric (e.g., an absolute risk factor percentage scaling from 0% to 100%). This dynamic score will be computed in real time by an automated weighted matrix engine that simultaneously evaluates multiple parameters: active domain registration age records, SSL certificate verification pathways, hosting infrastructure integrity, global domain trust indexes, and structural token anomalies.

4. Immersive User Awareness Modules and Explanatory Visual Hazard Warnings:

   To improve user security awareness during scanning, future updates will enhance the threat alert interfaces with intuitive, educational visual components. Instead of displaying a basic warning notice, the alert window can highlight the specific structural red flags found (e.g., showing a hidden sub-domain or showing that a link uses an insecure connection). This gives users clear, actionable insights to make safer choices during day-to-day digital transactions.

6. CONCLUSION

This research project has successfully engineered, implemented, and validated a lightweight, highly responsive, real-time QR Code Attack Detection System specifically designed to reinforce client-side mobile security configurations against modern social engineering threats. By shifting the primary analytical defensive mechanisms from conventional, slow cloud-dependent lookup databases directly to an optimized on-device heuristic evaluation script layer, the developed platform completely eliminates network routing overheads, multi-hop lookup delays, and data transmission latency bottlenecks. This ensures that defensive profiling occurs instantly upon data acquisition, making it exceptionally well-suited for deployment on consumer-grade mobile devices with limited hardware resources.

The successful integration of the custom 5-rule heuristic checking engine allows the application to accurately categorize multi-vector QR code inputs into structured, color-coded risk profiles—Green (Safe), Orange (Suspicious), and Red (Malicious)—prior to browser interaction. This proactive interception pipeline fundamentally halts the execution of hidden zero-day quishing links, automated drive-by malware downloads, and manipulative web redirection pathways. Furthermore, the platform achieves robust operational sustainability by preserving detailed historical records within a local SQLite database infrastructure, which seamlessly feeds an interactive administrative dashboard equipped with responsive data tracking charts and configuration panels.

Ultimately, this project demonstrates that robust, modern anti-phishing defense mechanisms can be deployed natively within mobile client environments without compromising processing efficiency or user convenience. The resulting application provides an accessible, reliable, and user-centric security framework that effectively hardens mobile devices against the rapidly growing threat of QR-code-based cyberattacks.

ACKNOWLEDGEMENT

We express our deepest gratitude to our mentor, Mrs.Anitha.M, Assistant Professor, Department of Computer Science and Engineering , Avinashilingam Institute for Home Science and Higher Education for Women for the invaluable guidance and insightful feedback throughout the course of this research project. We are also highly thankful to the Department

of Computer Science and Engineering at Avinashilingam Institute for Home Science and Higher Education for Women for providing the essential infrastructure, institutional resources, and academic facilities required to carry out this work successfully.

REFERENCES

1. Ahmad Sahban Rafsanjani , Norshaliza B .Kamaruddin(2023). QSecR:Secure QR code scanner According to a novel malicious URL detection framework, IEEE Access :https://ieeexplore.ieee.org/document/10172018.

2. Aayush Trivedi, Krishnappa Jangal, Rashi Gupta(2025) Phishing Detection in Advanced QR Code Attacks: Challenges and AI- Driven Solutions, International Journal for Research in Applied Science & Engineering Technology(URASET) :https://www.ijraset.com/research-paper/phishing-detection-in-advanced-qr-code-attacks.

3. Rouwa Yalda, Arshad Khan, Narayan Nepal(2026).B-PhishQR – A BlockChain- based framework for secure QR code Verification against Phishing attacks :https://www.sciencedirect.com/science/article/pii/S2772503025001033.

4. Irving Barron, Gaurav Sharma (2025). Quashing Quishing Attacks Using Self- Authenticating Dual-Modulated QR Codes, IEEE Security & Privacy. DOI: https://ieeexplore.ieee.org/document/10877412

5. Muhammad Wahid Akram, Keshav Sood, Muneeb Ul Hassan (2025). QRïS: A Preemptive Novel Method for Quishing Detection Through Structural Features of QR Codes, arXiv : https://arxiv.org/abs/2510.17175

6. Jikesh Thapa, Gurrehmat Chahal, Şerban Voinea Gabreanu, Yazan Otoum (2025). Phishing Detection in the Gen-AI Era: Quantized LLMs vs Classical Models, arXiv : https://arxiv.org/abs/2507.07406

7. Saranya Vaithilingam, Santhosh Aradhya Mohan Shankar (2024). Enhancing Security in QR Code Technology Using AI: Exploration and Mitigation Strategies, International Journal of Intelligence Science https://doi.org/10.4236/ijis.2024.142003