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

TrustLens: A Survey of Multi-Layer URL Threat Detection for Secure QR Code Scanning

Rohan Londhe[1], Kunal Mahamuni[2], Anand Shinde[3], Ajaysinh Mulmule[4]
[1,2,3,4] Department of Computer Engineering, SVPM's College of Engineering, Malegaon Bk., Pune, India

Abstract - A QR code conceals what it encodes from the bearer of the scanning device: the encoded address is invisible until decoded. Attackers exploit this by means of QR-code phishing (quishing) and most QR-code scanners do not offer any protection against such attacks: they simply decode the QR code and open the decoded URL, without any security checks. In this paper we review research on QR-code and malicious-URL threat detection, classified according to whether the analysis is performed on the QR image or the decoded URL, the type of analysis (blacklists and reputation lookup, lexical machine learning, content and host inspection, cryptographic verification), and how the results are presented to the user. We compare twenty studies published between 2022 and 2026 according to the problem addressed, the approach followed and the gap addressed. Based on this review we propose TrustLens, a multi-layer secure QR-code scanner that decodes the payload, normalizes the URL and combines reputation, lexical machine-learning and redirect analysis evidence into a single risk score that is mapped to Safe, Suspicious or Malicious along with supporting arguments for the user to consider. TrustLens is a design proposal, not yet implemented and evaluated. The paper concludes with evaluation guidelines, open challenges and directions for future work

Key Words: QR code, quishing, phishing detection, malicious URL detection, risk scoring, mobile security.

1. INTRODUCTION

In modern times, QR codes are increasingly used to open web pages, install applications, make payments to the merchant, and connect to various online services. At the same time, QR-code contains a string of data that is invisible to humans, namely, the Uniform Resource Locator (URL). However, this characteristic of QR codes is exploited by criminals to mislead users, divert them to phishing resources or fraudulent sites [1], [2]. The first researchers managed to demonstrate that users of smartphones are at risk of being deceived through QR codes [3]. Later, scientists have confirmed this thesis with an experiment that demonstrated the successful use of malicious QR codes in real-life scenarios [4]. Moreover, it was revealed that URLs hidden in QR-codes pose a significant security threat to users [5].

Typical scanner apps simply decode a QR code, and open the decoded link, with little or no security checks. Scanners that do perform link vetting typically rely on blacklists, which are unsatisfactory for newly created websites [5], [6] A user study with 556 participants demonstrated that users of default iOS and Samsung scanners got correct answers 75.24% and 65.11% of the time respectively, whereas the security focused SEQR scanner achieved 93.35% [1], showing that default scanners do not solve the problem.

Research has been done on multiple fronts: scanners targeting QR codes specifically [1], [5], [7]; detectors operating on the QR image or structure [8]–[10]; cryptographic and physical level verification [11]–[13]; and machine-learning (ML) based URL classifiers, which are not specific to QR codes [14]–[16]. However, each front typically examines only a single source of evidence. A link not present in any blacklist may still be malicious, and a seemingly innocuous string may redirect

\

through multiple URLs to a dangerous endpoint. to a dangerous endpoint. What is needed to a dangerous endpoint. What is required is a scanner that blends multiple independent signals and explains to the user in simple terms how dangerous a link is and why.

This paper presents the following contributions:

•A structured review of approaches to QR-code and hostile-URL detection, ranging from blacklists and lexical ML to QR image inspection and cryptographic checks.

•A paired comparison of twenty studies (2022-2026) on this topic, covering the addressed problems, solutions, and remaining gaps.

•TrustLens, a proposed design that merges reputation, lexical ML and redirect evidence, with three classes of verdict and guidelines for evaluation and open challenges.

The paper is structured as follows. Sec. 2 presents some background and requirements, while Sec. 3 covers URL detection methods, and Sec. 4 QR scanners. Sec. 5 discusses risk fusion and warning design, while Sec. 6 compares related works. The TrustLens framework is presented in Sec. 7, its evaluation, open challenges, and future directions in Secs. 8-10, and finally concluding remarks in Sec. 11.

2. BACKGROUND AND REQUIREMENTS

2.1 QR Codes and Quishing

A QR code encodes a payload, typically a URL but possibly also plain text or other structured data. The way the module positions are defined (version, error-correction level and mask pattern) follows an international standard, and doubts about the security of this coding format have been expressed early [17]. QR codes are also being extended to new applications, such as executable variants for the Internet of Things [18]. Being decoded automatically, whatever the next action is performed by the scanner decides on the security. A malicious code can be submitted anywhere where a genuine one is expected, and the potential damage ranges from stealing credentials to distributing malware. A systematic review from 2025 on 50 studies catalogues the attack types, detection methods and countermeasures so far reported [2], while older surveys address attacks and the challenges of usable security [19]–[22].

2.2 Attack Landscape

Reported attacks include substitution or overlay of genuine codes, hiding one barcode within another [23], taking over codes invisibly with active light sources [24], and leading victims to browser-in-the-browser phishing pages via a scanned code [25]. Work centred on users indicates that people have difficulty assessing a code before they scan it, and the evidence comes from susceptibility experiments [3], work with malicious codes [26], a field study of real phishing [27] and an inquiry into how people use QR codes on a university campus [28]. This is in line with the wider phishing literature, in which convincing

pages often deceive users [29] and phishing remains a persistent danger [30].

2.3 Requirements for a Secure Scanner

Drawing on the emphasis on usability in [1], the privacy-friendly design of [5] and the practical limits of mobile deployment, we think the following requirements are essential to a secure scanner:

•Safety before access: every link is verified ahead of opening, and a suspicious link is never opened without user action.
•Low latency: the verdict has to arrive quickly enough to allow the natural scanning gesture.
•Low false-alarm rate: wrongly blocked safe links erode trust and teach us not to trust warnings.
•Clear, actionable communication: a small set of risk classes with accompanying reasons, instead of a raw score [1].
•Resilience: a verdict is always produced even if external services are down or throttled.
•Privacy and cost: as little data as possible leaves the device, and the scanner is affordable on everyday smartphones.

3. MALICIOUS URL DETECTION APPROACHES

After decoding a QR code, the problem reduces to determining whether the decoded Uniform Resource Locator (URL) is malicious. Table -1 summarizes the main families of approaches, which are detailed next. For general surveys of malicious-URL and phishing detection, see [31]–[35].

Table -1: Approaches to malicious URL and QR-code detection

Approach	Strengths	Limitations
Blacklist / reputation	It handles familiar threats quickly and accurately	It might miss New URLs if the isn't updated or hit service limits
Lexical machine learning	Fast and Efficient because it skips heavy page downloads to keep latency to minimum	If a URL is crafted well enough it can totally look harmless depending on the data you're comparing it against
Content / host analysis	It looks at more than the URL text to find proof	Because it needs to check page or registration details, it's a bit slower and exposes more user privacy
Link redirection	Shows actual destination of link	Requires extra network requests so limit will be applied
QR-image deep learning	Flags image without decoding	Expensive computationally ignores reputation of destination
Cryptographic verification	Strong integrity and revocation	Requires infrastructure and key mangement

3.1 Blacklist and Reputation Checking

Blacklists and reputation services such as PhishTank [37] and VirusTotal [38] provide instant answers on the safety of a given domain or URL but are ineffective at flagging URLs that are not in the database. PhishTank's database of phishing URLs is slow to be updated after new phishing attacks are created [6]. A broad study on the lifecycle of phishing URLs found that it can take a long time between a URL's creation and its addition to a phishing list [39]. Consequently, web site scanners relying on reputation lists or blacklists have poor performance when it comes to newly created web sites [5]. These scanners depend on a timely feed of data, and online web services can provide this in varying degrees based on the availability and response time of the services. DNS data can also provide a basis for building reputation systems [40, 41] [42] and on survey work covering DNS-based domain detection [43] reach beyond static lists.

3.2 Lexical Machine Learning

Lexical methods are entirely based on the characters comprising the URL and are not related to its contents, thus never downloading the page. Common attributes are URL and domain length, number of dots, hyphens, subdomains, an IP address or "@" in the address, HTTPS utilization, number of special characters, suspicious keywords and shortener use. Studies in the early period revealed that a combination of lexical and host-based attributes sufficed to distinguish malicious URLs from benign ones without the need for a blacklist [44]. An online learning on lexical features [45] and a statistical survey on phishing URL writing [46] then demonstrated that the string alone was highly informative, and streaming analytics brought URL features to large scale [47]. Feature-driven frameworks followed in [48], [49], and subsequent surveys have listed the feature sets and algorithms that appeared [31].
Deep models, in turn, learn their own representations from the URL itself: character- and word-level CNNs [50], [51], recurrent networks [52], multidimensional deep-learning features [53], a real-time deep-learning detection system [54], a framework based on a similarity index and incremental learning [55] and, latest of all, a transformer-based model [56]. Tree ensembles still hold their ground on tabular features: Random Forest [57], XGBoost [58] and LightGBM [59] are popular choices, and a wide benchmark concluded that, on ordinary tabular data, tree-based models continue to beat deep learning [60]. Current work focusses on feature engineering; PhishOFE presents an optimized feature-engineering framework for real-time phishing URL detection [14], and a comparative study evaluates URL, HTML and derived features on ten ML models across 101,063 URLs [15]. Lexical models are lightweight, yet a well-crafted URL can appear innocuous, and a model trained on one dataset may not generalize to the URLs that actually circulate via QR codes.

3.3 Content and Host Analysis

Further evidence can be taken from thecontents of a page (its HTML) or from domain-registration and hosting records. A static analysis of HTML,JavaScript, URL and host attributes can serve as a fast first-line filter for malicious pages [61]; extraction of identity keywords can reveal the brand that a page is impersonating [62]; and ML running on the client can assess pages independently of thirde-party services [63]. According to the comparison in [15] some approaches require fetching the page itself, which entails time andexposure for the analysing system. Early-detection research focuses on phishing URLs hosted on freshly-registered or parked domains [16], where reactive blacklists usually lag.

3.4 Link redirection

Shorteners and redirect chains can obscure the true destination of a link, meaning that a shortened, seemingly innocuous URL may actually point to a malicious resource. This can be detected by resolving the entire chain, analyzing its length and the properties of the final URL, and repeating the other checks on the final destination. In the studies surveyed here, no QR

scanner pairs redirect analysis with lexical ML and reputation checks.

3.5 QR-image deep learning

A different family of methods classifies the QR code as an image and ignores the URL it carries. A deep-learning classifier of QR images reported high accuracy together with a real-time prototype [8]; lightweight CNNs combined with feature QR code detectors (e.g., QRscanner) [67] are generally weak when it comes to detecting QR codes, even though they have been extensively studied. QR codes that have been attacked have gone through several rounds of image processing [64]; CNNs that have been fed with histogram-density features perform well on a small dataset [65]; and custom CNN, ResNet and VGG models were trained on synthetic codes built from URL lists [66]. ML applied to QR pixel patterns reaches an AUC near 0.91 [9]. QRïS, by contrast, extracts 24 interpretable structural features from the code layout and achieves up to 83.18% accuracy, accepting lower accuracy in exchange for transparency [10]. Such methods can raise a flag without decoding the code, but they do not look at the decoded destination's reputation or its redirects, and image-level deep models require more computation [8], which is important for resource-constrained devices [67].

3.6 Cryptographic and Physical-Layer Verification

B-PhishQR recommends verifying QR codes on a blockchain using cryptographic proof of authenticity and immediate revocation [11]. Dual-modulated self-authenticating codes contain embedded signatures attesting to the payload [12]; ScreenID authenticates displayed codes by fingerprints unique to each screen [13]; and design-oriented proposals seek to encode integrity directly into the code [68]. They offer good protection, but focus on authenticity, not on lightweight URL-risk analysis, and require additional infrastructure, key management or dedicated scanners, leaving many ordinary codes already deployed vulnerable.

4. QR-SPECIFIC SCANNER SYSTEMS

Some scanner applications and frameworks have specifically been developed for QR codes.

QsecR [5] is an Android QR scanner that was built upon a malicious-URL detection framework with 39 feature classes. Evaluated on 4,000 URLs, it is said to attain 93.50% accuracy and 93.80% precision. Its strengths are its being light weight and respecting privacy; its weaknesses are having fixed predefined features, a relatively small dataset and only limited explanation of its decisions.

SEQR [1] is a secure scanner aimed towards usability that raises phishing, suspicious or safe warnings. It was developed out of QRishing and usability research and was tested with 556 participants. Because it is mainly reputation-based, its coverage relies on threat intelligence, and its warnings are generic instead of listing out detailed risk factors.

The scanner with a real-time anti-QRishing technique in [7] is, based on its stated scope, not proven to include lexical ML, reputation checking, redirect analysis and explainable risk fusion. QR-link detectors then rely on host-based features inside a real-time AI system [69], or on lexical features fed to a neural network optimized by squirrel search [70]. QRïS adds a Flutter application running on a Flask service [10], yet it assesses the structure of the code and leaves the destination unexamined.

Overall, these systems prove that secure scanning is possible on smartphones, albeit each only handles one or two types of evidence.

5. RISK FUSION AND WARNING DESIGN

5.1 Why Fuse Signals

Each signal is blind in its own way: reputation lookups miss newly created URLs, lexical patterns can be fooled by well-formed attacks, and redirect analysis only helps if a redirect is present. By aggregating multiple independent signals into a combined score, we allow each signal to compensate for the others' weaknesses; the combination of several detection modules has been used in similar settings to identify covert channels [71]. Finally, weighted scoring offers a simple and intuitive method to combine multiple signals. Each individual module returns a normalized score, and the combination is performed as a weighted sum that is then split into intervals for the final risk category. This approach is intuitive and provides immediate insight into the relative contribution of individual components via adjustable weights.

5.2 Communicating Risk

The safe, suspicious and phishing warnings proposed by SEQR [1] suggest that defining a small set of risk classes is indeed feasible. A proper warning communicates the risk class, lists the primary reasons (e.g., "redirects to a different domain" or "matches a reported phishing URL"), and recommends the course of action. Field studies on browser warnings confirm that the design of the warning significantly impacts the propensity of the user to heed it [72], that active warnings are better than passive warnings [73], and that different users are more easily deceived depending on their demographics [74]. For highly risky warnings, confirmation should therefore be requested before opening such links, and suspicious links should not be opened automatically.

5.3 Explaining a Score

Model-agnostic explainers such as LIME [75] and SHAP [76] credit a model's output to individual input features, and so could supply the "key reasons" displayed beside a scanner's score, for instance that a URL is long, includes an IP address along suspicious keywords.

6. COMPARATIVE ANALYSIS OF EXISTING WORK

6.1 Comparison of Representative Approaches

Table -2 sets representative studies side by side on the dimensions that matter most for this survey. The comparison is qualitative and is based on how each work describes itself.

Table -2: Representative approaches (qualitative)

Approach	Level of analysis	Technique	Coverage of reputation, redirects and explanation
QsecR [5]	Decoded URL	ML on 39 static feature classes	Neither reputation nor redirect analysis; explains risk only to a limited degree
SEQR [1]	Decoded URL	Reputation-based, three warning classes	Depends on threat intelligence; lacks lexical ML; warnings are generic
Anti-QRishing scanner [7]	QR scanning	Real-time anti-Qrishing technique	A joint use of ML, reputation, redirects and risk fusion is not established

Approach	Level of analysis	Technique	Coverage of reputation, redirects and explanation
QR image deep learning [8], [64], [65], [66]	QR image	CNN / deep-learning image classification	Omits decoded-URL, reputation and redirect analysis; opaque model with little explanation
Quishing ML [9]	QR-code pixels	Machine learning on pixel patterns	Lacks a decoded destination and multi-signal risk analysis
QRïS [10]	QR structure	24 structural features, XGBoost / RF	Interpretable and usable before decoding, yet gives no URL, reputation or redirect evidence
B-PhishQR [11]; dual-modulated QR [12]; ScreenID [13]	QR authenticity	Blockchain, signatures, screen fingerprints	Strong on authenticity and revocation, but depend on dedicated issuance or hardware and lack lightweight URL-risk analysis
PhishOFE, comparative study, parked-domain framework [14], [15], [16]	URL / domain	ML with engineered features	Not QR applications; do not handle QR payloads or provide user-facing risk fusion
TrustLens (proposed)	Decoded URL	Lexical ML + reputation + redirect analysis, fused	Combines all three signals; gives an explainable Safe / Suspicious / Malicious verdict

Three points stand out. First, QR-specific systems commonly rely on one main source of evidence, either the static URL characteristics [5], reputation [1], QR image content [8], [9] or QR code structure [10]. Second, redirect analysis and explainable risk fusion are rarely central objectives, and no reviewed system is described to combine every signal discussed here. Third, URL-level frameworks [14]–[16] provide powerful feature engineering, but are not part of a QR-scanning application, requiring further integration.

6.2 Summary of Related Literature

Table -3 summarizes twenty representative studies from 2022 to 2026, most recent first, giving the problem each address, its methodology, its proposed solution and the gap what remain. Reading across these reports, four patterns emerge. First QR-specific works handle QRs' decoded URLs or QRs themselves rarely together, and never both along with reputation and redirect evidence. Second image-based QR classifiers report exceptionally high accuracy (95.89% to 98.13%) on small, often synthetic datasets [64]-[66], while the interpretable structural method reports 83.18% on 400,000 synthetic samples [10] so accuracy figures are not directly comparable across works. Third evaluation sizes vary greatly, from 4,000 URLs for QsecR [5] to 101,063 URLs for the feature-engineering comparison [15]. Finally, only SEQR [1] and the real-world phishing study [27] consider real user behaviour, leaving the user-centred evaluation domain under-explored. This motivates the tiered, explainable design of Sec. 7.

Table -3: Summary of related literature

Sr. No	Paper title	Author(s)	Year	Methodology	Existing problem statement	Existing problem solution	Future scope / gap
1	B-PhishQR – A Blockchain-Based Framework for Secure QR Code Verification Against Phishing Attacks [11]	R. Yalda, A. Khan, N. Nepal	2026	Verification rooted in a blockchain: cryptographic assurances that a code is genuine, and revocation upon compromise of codes by issuers.	Phishers exploit QR codes whose genuineness a user has no means of checking.	A mechanism that allows users to cryptographically prove that a QR code is authentic and can be revoked in real time.	Whilst focusing on authenticity and integrity the system has an increased infrastructure and complexity associated with managing cryptographic keys.
2	Development, Evaluation, and Implementation of SEQR – a Usable Secure QR Code Scanner [1]	M. Mossano, M. F. Veit, T. Länge, B. M. Berens, F. Sharevski, M. Volkamer	2026	A scanner driven primarily by reputation, which highlights Phishing, Suspicious or Safe warnings, fielded in a user study against the built-in iOS and Samsung scanners, by 556 participants.	QRishing attacks, combined with scanners that offer users little help in judging a code.	The SEQR scanner with three warning classes; participants judged correctly 93.35% of the time, versus 75.24% with iOS and 65.11% with Samsung.	Mostly reputation-based; has no lexical ML classification and no detailed explainable risk factors; relies on threat intelligence; warnings are generic.
3	Quick Response Code Security Attacks and Countermeasures: A Systematic Literature Review [2]	D. Njuguna, J. G. Ndia	2025	A systematic review spanning 50 studies and cataloguing attack types, detection techniques and countermeasures; no system is constructed.	Knowledge about QR-code attacks and defences is spread across many sources.	A consolidated overview of QR attacks, detection methods and countermeasures drawn from 50 studies.	A review, not an implemented end-user detection system; offers no deployable QR-scanning application.
4	Efficient Malicious QR Code Detection System Using an Advanced Deep Learning Approach [8]	A. A. Alsulami, Q. Abu Al-Haija, B. Alturki, A. Yafoz, A. Alqahtani, R. Alsini, S. S. Binyamin	2025	Classifies QR-code images as malicious with deep learning; doesn't decode the payload.	Malicious QR codes need to be detected efficiently, preferably in real time.	A deep-learning detector that reports high accuracy and comes with a real-time prototype.	Only image-based detection is performed. Detection does not take into account any decoded URL, reputation or redirection. Higher computational costs. Poor explanation of risk at destination.

Sr. No	Paper title	Author(s)	Year	Methodology	Existing problem statement	Existing problem solution	Future scope / gap
5	QR Code Scanner with Anti-Qrishing Real-Time Technique [7]	R. Shevchuk, P. Ihas	2025	Real-time scanner employing an anti-QRishing technique; it's published scope doesn't mention lexical ML, reputation, redirects or risk fusion working in concert. ML framework centred on optimized feature engineering; features sought are discriminative and efficient.	People scan codes that send them to phishing destinations (QRishing).	A QR scanner built specifically to counter QRishing in real time.	Combination of lexical ML, reputation, redirect analysis and explainable risk fusion not established; narrow publication scope.
6	PhishOFE: A Novel Machine Learning Framework for Real-Time Phishing URL Detection With Optimized Feature Engineering [14]	Y. A. Kustiawan, K. I. Ghauth	2025	ML framework centred on optimized feature engineering, aiming for features that are both discriminative and efficient.	Real time phishing URLs detection requires discriminative features for fast computation.	Machine Learning Approach for Real Time Phishing URLs Detection Using Optimized Features	A stand-alone URL framework, not a QR security application; further integration needed.
7	Evaluating the Impact of Feature Engineering in Phishing URL Detection: A Comparative Study of URL, HTML, and Derived Features [15]	Y. A. Kustiawan, K. I. Ghauth	2025	URL, HTML, and feature-based features are evaluated on ten different machine learning models using a total of 101,063 URLs to determine how much each feature adds.	Unclear contributions of URL, HTML and derived features in detection.	A comparative study to understand the impact of feature engineering on the detection of phishing URLs. The study uses 101,063 URLs and 10 models.	Not QR-specific; complex framework; some methods need access to the web page.
8	Machine Learning for Early Detection of Phishing URLs in Parked Domains [16]	J. D. Duarte et al.	2025	ML framework for detecting new and parked domain phishing URLs.	Early detection of phishing URLs on new and parked domains..	An early-detection ML framework for phishing on new and parked domains.	Limited to particular domains; neither a general QR scanner nor an integrated destination-risk engine; offers no risk fusion for users.
9	Detecting Quishing Attacks with Machine Learning Techniques Through QR Code Analysis [9]	F. Trad, A. Chehab	2025	Pixel pattern data is extracted from codes generated by benign URLs (5000) and phishing URLs (5000). These are used to train ML models. The XGBoost model gave the best results (AUC: 0.9106).	Quishing should be detected from the QR code itself, without opening the payload.	ML applied to QR pixel patterns. XGBoost gave best results (AUC: 0.9106) using data from benign URLs (5000) and phishing URLs (5000).	Works at QR level rather than on the decoded destination with multi-signal risk analysis; pixel features resist interpretation; dataset is small, synthetic and uniform.
10	QRïS: A Preemptive Novel Method for Quishing Detection Through Structural Features of QR [10]	M. W. Akram, K. Sood, M. U. Hassan	2025	Derives 24 protocol-level and statistical structural features from the QR layout and classifies them with XGBoost and Random Forest; ships as a Flutter app with a Flask backend.	Black-box and content-based QR classifiers either lack interpretability or expose users to the payload.	Interpretable pre-decode detection using 24 structural features on 400,000 synthetic samples, reaching up to 83.18% accuracy, delivered as a Flutter app on a Flask backend.	Less accurate than black-box CNNs; restricted to black-and-white square-module codes; offers no URL, reputation or redirect evidence; needs a connection.
11	BERT-PhishFinder: A Robust Model for Accurate Phishing URL Detection with Optimized DistilBERT [56]	A. Aljofey, S. A. Bello, J. Lu, C. Xu	2025	Fine-tunes an optimized DistilBERT language model that reads the URL text directly instead of engineered features.	Accurate phishing URL detection needs models that stay robust beyond hand-crafted features.	A phishing URL classifier built on an optimized DistilBERT model.	Not QR-specific; transformer inference is heavy for on-device use; no reputation or redirect evidence.
12	Quashing Quishing Attacks Using Self-Authenticating Dual-Modulated QR Codes [12]	I. R. Barron, G. Sharma	2025	Embeds cryptographic signatures into dual-modulated QR codes; a dedicated scanner checks the signature to authenticate the payload.	Users cannot verify the payload of a printed or displayed QR code.	Secure dual-modulated QR codes with self-verifying signatures that prove the authenticity of the payload.	Needs a specialized scanner and code generation at the issuer; leaves ordinary codes already in circulation unprotected.
13	PhiUSIIL: A Diverse Security Profile Empowered Phishing URL Detection Framework Based on Similarity Index and Incremental Learning [55]	A. Prasad, S. Chandra	2024	This method uses similarity index along with incremental learning for detecting phishing URLs. A large data set has been provided.	Phishing URL detectors need varied, up-to-date features and datasets.	This framework uses similarity index along with incremental learning for detecting phishing URLs. A large data set of URLs has been provided.	Not QR-specific; not packaged as a scanner with explanations for users.

14	Recognition of Valid QR Codes with Machine Learning [65]	A. Minocha, A. Goyal, R. Gandhi	2024	Trains CNN models (ResNet50, MobileNetV3, Inception) on QR images using histogram-density features; 10,000 codes.	Valid QR codes must be told apart from phishing ones.	CNN-based classification of QR images with histogram-density features, reaching up to 98.13% accuracy on 10,000 codes.	Small dataset; image-level only; features cannot be interpreted; no destination reputation or redirect analysis.
Sr. No	**Paper title**	**Author(s)**	**Year**	**Methodology**	**Existing problem statement**	**Existing problem solution**	**Future scope / gap**
15	Hooked: A Real-World Study on QR Code Phishing [27]	M. Geisler, D. Pöhn	2024	Real world experimentation has been carried out to observe the results of QR code phishing.	Little empirical evidence exists about how real users react to QR-code phishing.	Experimental results from QR code phishing.	Studies threat instead of offering a deployable detector; no multi-signal URL analysis.
16	QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework [5]	A. S. Rafsanjani, N. B. Kamaruddin, H. M. Rusli, M. Dabbagh	2023	Extracts 39 feature classes from the decoded URL and classifies it in an Android scanner; tested on 4,000 URLs.	Malicious URLs in QR codes are a growing security concern, and blacklist-based scanners miss new sites.	An Android scanner reporting 93.50% accuracy and 93.80% precision on 4,000 URLs.	Fixed, predefined features; fairly small dataset; lacks broader ML, reputation, redirect analysis and explainable risk fusion.
17	Cyber Attack Detection with QR Code Images Using Lightweight Deep Learning Models [64]	Y. Alaca, Y. Çelik	2023	Extracts features from QR images with lightweight CNNs (MobileNetV2, ShuffleNet), selects them with Harris Hawks optimization and classifies with SVM or kNN.	Attacks encoded as QR images must be detected at low computational cost.	A lightweight deep-learning pipeline; SVM reached 95.89% accuracy.	Not designed for quishing; small dataset; black-box features; no URL or reputation analysis.
18	Malicious URLs and QR Code Classification Using Machine Learning and Deep Learning Techniques [66]	D. O. D. R. Lourenco, M. S. Sriraj, K. K. Thambi, V. Ranjan	2023	Generates synthetic QR codes from URL lists and trains a custom CNN, ResNet50 and VGG19 on them.	Malicious URLs delivered through QR codes call for ML / DL detection.	Three image models compared on synthetic QR codes, reporting 97.13%, 86.65% and 74.18% respectively.	Synthetic, imbalanced data; limited interpretability; no destination reputation or redirect analysis.
19	Enhancing Malicious QR Code Link Detection with Swarm-Intelligent Squirrel-Search Optimized ANN [70]	S. Marappan, H. Marappan	2023	Lexical features extracted from the decoded URL are classified using an artificial neural network optimised using the squirrel search algorithm.	Malicious links inside QR codes should be detected from the decoded URL.	An ANN optimized with squirrel-search on lexical URL features.	Lexical evidence only; no reputation, redirect analysis or user-facing explanation.
20	ScreenID: Enhancing QRcode Security by Utilizing Screen Dimming Feature [13]	G. Xue, Y. Li, H. Pan, L. Yang, Y.-C. Chen, X. Ji, J. Yu	2022	Checks QR legitimacy through screen-specific fingerprints derived from the screen-dimming feature.	QR codes shown on screens can be forged or swapped.	Validation of displayed codes by fingerprints produced through screen dimming.	Depends on hardware and specialized scanners; limited to screen-displayed codes; no URL-risk analysis.

7. TRUSTLENS: PROPOSED FRAMEWORK

TrustLens is a secure QR scanner for Android which is currently being developed. This section describes its design, which is inspired by the results of the preceding survey. It is a design proposal and no implementation has been evaluated yet.

7.1 Architecture

Fig -1 depicts the pipeline. A user scans a code with the camera or chooses an image from the gallery, and the decoded payload then passes through ten stages:

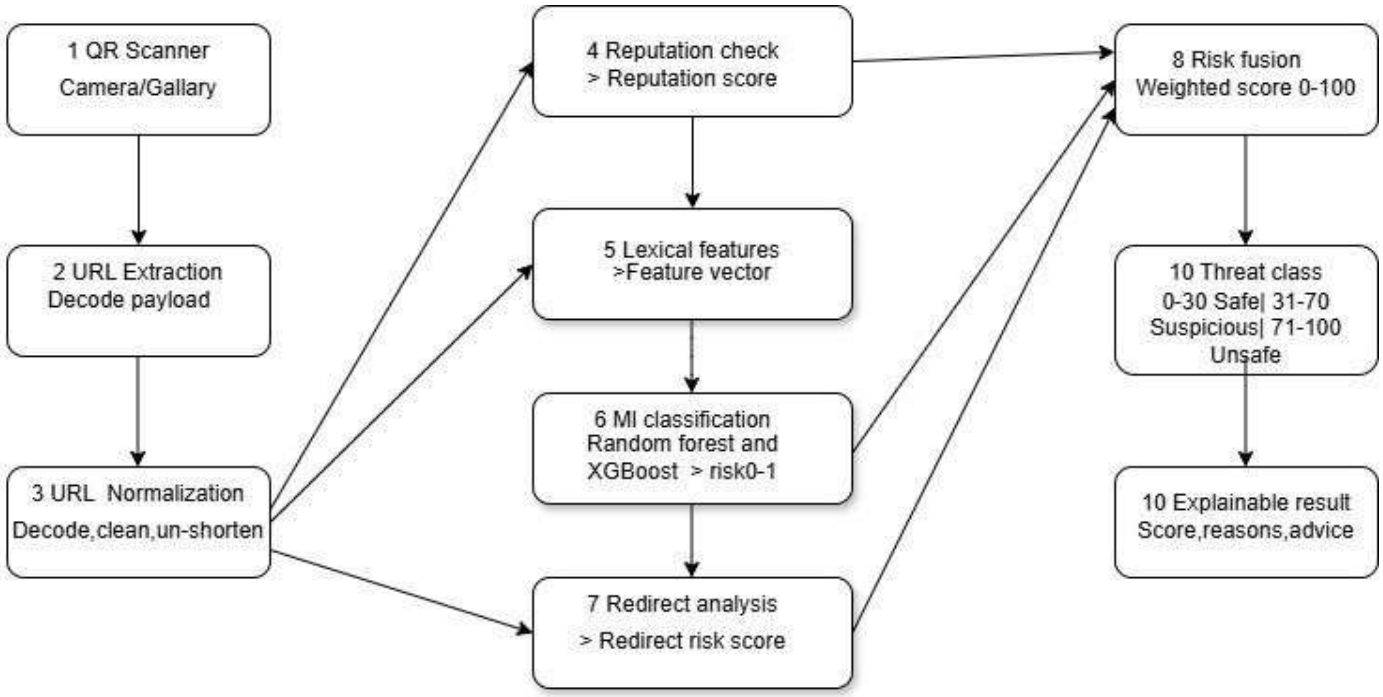


Fig -1: Proposed TrustLens pipeline, from QR scan to explainable verdict

•**Scanning and extraction (stages 1-2):** Google ML Kit or ZXing decodes the payload, from which a URL is extracted. Non-url content is presented to the user but does not receive a threat score.

•**Normalization (3**): The URL is decoded, cleaned and normalized, and any shortened link is expanded.
•Reputation check (4): the normalized URL is compared against a local blacklist and against threat-intelligence APIs (Google Safe Browsing [36], PhishTank [37], VirusTotal [38]), giving a reputation score.
•**Lexical features and ML classification (5-6):** the URL's length, domain length, any IP address, the number of dots and hyphens, HTTPS, keywords, number of subdomains, special characters, path length, and shortener use are assembled into a feature vector. A Random Forest classifier [57] exposed through a Flask REST API turns this vector into an ML risk score from 0 to 1, whilst Logistic Regression and XGBoost [58] are tested as alternatives.

- **Redirect analysis (7)**: the redirects are followed and the resulting chain is investigated for suspicious characteristics which yield a redirect risk score, with a maximum depth of analysis.
- **Risk fusion (8):** a weighted scoring approach is used to fuse the three scores to obtain an overall risk score ranging between 0 and 100.
- **Threat class and explainable result (9–10**): the overall score determines the threat class (Safe: 0–30; Suspicious: 31–70; Malicious: 71–100) together with an explanation of the most relevant contributing factors, a safety recommendation and the indication that the URL was not automatically opened.

7.2 Design Principles

- **Analyse before opening:** a link stays closed until the verdict is displayed, and Suspicious or Malicious links additionally need the user's explicit confirmation.
- **Layered evidence:** the decision is never based on a single source of information. Reputation, lexical and redirect evidence are combined in the final result.
- **Graceful degradation:** results are cached, and when the external services time out or become unreachable, the scanner falls back to lexical analysis.
- **Explainability:** the scanner provides an analysis and recommendation, not just a final score.
- **Bounded analysis:** the only information the scanner uses is the URL string, reputation evidence and a short redirect trace. No page content is retrieved or executed.
- **Low cost**: the scanner is implemented in open-source software and relies on free tier services. It can run on commodity mobile devices such as Android 8 or newer.).

7.3 Positioning

In contrast with systems described in Table -2 that introduce a new classifier, TrustLens is different by combining lexical ML, reputation checking, redirect analysis and an explainable fusion to deliver into a single QR scanner, a combination not found in any of the QR specific systems reviewed.

8. EVALUATION METHODOLOGIES AND DATASETS

Accuracy on a benchmark does not by itself show that a scanner keeps users safe, so the evaluation planned for TrustLens operates at three levels:

- **Algorithmic:** Accuracy, precision, recall, F1-score and false-positive rate of the lexical classifier on a held-out test set; a head-to-head comparison of Random Forest, Logistic Regression and XGBoost, whose standard library implementations are available [77]; and an evaluation of what each feature group meaningly contributes.
- **System-level:** How often QR decoding succeeds from the camera along the gallery; end-to-end latency between scan and verdict; whether labelled test URLs land in the correct Safe, Suspicious or Malicious band; and how the system behaves when APIs time out or enforce rate limits.
- **User-level:** a task-based study where participants decide whether QR codes are safe or malicious with and without TrustLens, comparing the proportion of correct decisions in the manner of [1] and also measuring how well warnings are understood and how often they are dismissed.

Labelled benign and phishing URLs are required for training and testing. Public benchmark corpora [55], [78], [79] and threat feeds can be used as sources, but were compiled for general purpose URL categorization, not QR codes, and QR-specific evaluations often used small subsets, such as the 4,000 URLs in [5]. Several QR-specific works generate synthetic codes from URLs in a list [9], [10], [66]; these codes carry URL-level labels and encoding parameters and may not reflect real-world variation. Since lexical models are likely to overfit to properties of a particular training set, testing on other datasets and a held-out partition from the same set in another timeframe is advisable. To evaluate end-to-end performance, the URL benchmark sets should be supplemented with a small set of real-world QR codes representing benign, phishing, shortening, and redirection categories. Finally, user studies require approval from an institutional ethics board.

9. CHALLENGES AND OPEN ISSUES

- **False positives and false negatives:** • blocking a safe URL is irritating to the user whereas letting a malicious one slip entirely defeats the purpose, thresholds must be calibrated.
- **Dependence on external services:** the latency, rate limits and downtime of APIs restrict real-time operation, which calls for caching and a local fallback.
- **Privacy:** forwarding scanned URLs to third-party reputation services reveals what the user is scanning, and so does performing analyses on these URLs locally seem appealing [5].
- **Evasion and drift:** attackers change their URLs, so features [5] and a model trained only once will decay, and so periodic retraining becomes necessary.
- **Redirect depth versus latency**: more hops uncovered reveals more but takes time and touches more hosts.
- **User behaviour**: warnings may be ignored, and so their wording, the confirmation steps and the presentation of reasons dictate the result [1], [72].
- **Dataset realism**: benchmark datasets may fail to represent the URLs that genuinely arrive through QR codes.
- **Scope limits**: payloads other than URLs, payment codes, executable codes [18] and analysis of page content fall outside what is covered at present.

10. FUTURE DIRECTIONS

- **Additional reputation sources:** consulting more free-tier threat-intelligence services as a cross-check, in order to reduce missed threats.
- **Stronger lexical models:** training on larger public datasets, comparing Random Forest, Logistic Regression and XGBoost, and tuning the fusion weights and thresholds.
- **Structural pre-check:** using QR-layout features [10] as an inexpensive signal ahead of payload decoding.
- **Feature-level explanations:** generating the reasons shown to users through SHAP-style attribution [76].

- **Feedback loop:** allowing users to report an incorrect verdict and using the recorded cases to adjust thresholds.
- **Offline mode:** storing recent reputation results and bundling a small compressed model [67] so that basic checks still work with no connection.
- **Link sharing:** receiving URLs shared from other applications via Android's share mechanism, so that the identical pipeline covers links that do not come from QR codes.
- **Payload types:** checking merchant and UPI payment QR codes.
- **Secure page preview:** a sandboxed view of suspicious destinations, planned as a longer-term addition.
- **Benchmark and user study:** a common benchmark of URLs delivered through QR codes, plus a user study once the application is complete.

11. CONCLUSIONS

QR-code phishing succeeds because users cannot tell where a QR code points and because many scanners automatically follow links without checking. Published work presents blacklist, lexical ML, content based, QR-image and cryptographic defences that are combined in QR-specific scanners to adopt a couple of them, but each manages one or two kinds of evidence and a few justify their verdicts. This survey categorized the area according to the level of analysis, detection technique and the way the risk is communicated, and compared twenty studies from 2022 to 2026. We presented TrustLens, a layered scanner that combines reputation, lexical ML and redirect-analysis signals into a three-class verdict with explanations before any link is opened. Realizing the design requires careful threshold calibration, robust handling of external services, representative datasets and, most of all, evaluation with real users.

ACKNOWLEDGEMENT

The authors are grateful to the Department of Computer Engineering, SVPM's College of Engineering, Malegaon Bk., Pune, for its support.

REFERENCES

1. Mossano, M., Veit, M.F., Länge, T., Berens, B.M., Sharevski, F., Volkamer, M.: Development, Evaluation, and Implementation of SEQR – a Usable Secure QR Code Scanner. In: Proc. ACM CHI Conference on Human Factors in Computing Systems (CHI '26). ACM (2026)
2. Njuguna, D., Ndia, J.G.: Quick Response Code Security Attacks and Countermeasures: A Systematic Literature Review. J. Cyber Secur. (2025). https://doi.org/10.32604/jcs.2025.059398
3. Vidas, T., Owusu, E., Wang, S., Zeng, C., Cranor, L.F., Christin, N.: QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks. In: Financial Cryptography and Data Security Workshops, LNCS 7862, pp. 52–69. Springer (2013)
4. Kharraz, A., Kirda, E., Robertson, W., Balzarotti, D., Francillon, A.: Optical Delusions: A Study of Malicious QR Codes in the Wild. In: Proc. 44th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN). IEEE (2014)
5. Rafsanjani, A.S., Kamaruddin, N.B., Rusli, H.M., Dabbagh, M.: QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework. IEEE Access 11, 92523–92539 (2023)
6. Sheng, S., Wardman, B., Warner, G., Cranor, L.F., Hong, J., Zhang, C.: An Empirical Analysis of Phishing Blacklists. In: Proc. 6th Conf. on Email and Anti-Spam (CEAS) (2009)
7. Shevchuk, R., Ihas, P.: QR Code Scanner with Anti-Qrishing Real-Time Technique. In: Proc. 15th Int. Conf. on Advanced Computer Information Technologies (ACIT), pp. 559–564. IEEE (2025)
8. Alsulami, A.A., Abu Al-Haija, Q., Alturki, B., Yafoz, A., Alqahtani, A., Alsini, R., Binyamin, S.S.: Efficient Malicious QR Code Detection System Using an Advanced Deep Learning Approach. Comput. Model. Eng. Sci. (CMES), 1117–1140 (2025)
9. Trad, F., Chehab, A.: Detecting Quishing Attacks with Machine Learning Techniques Through QR Code Analysis. arXiv:2505.03451 (2025)
10. Akram, M.W., Sood, K., Hassan, M.U.: QRïS: A Preemptive Novel Method for Quishing Detection Through Structural Features of QR. arXiv:2510.17175 (2025)
11. Yalda, R., Khan, A., Nepal, N.: B-PhishQR – A Blockchain-Based Framework for Secure QR Code Verification Against Phishing Attacks. Telemat. Inform. Rep. 21 (2026)
12. Barron, I.R., Sharma, G.: Quashing Quishing Attacks Using Self-Authenticating Dual-Modulated QR Codes. IEEE Secur. Priv. (2025)
13. Xue, G., Li, Y., Pan, H., Yang, L., Chen, Y.-C., Ji, X., Yu, J.: ScreenID: Enhancing QRcode Security by Utilizing Screen Dimming Feature. IEEE/ACM Trans. Netw. 31(2), 862–876 (2022)
14. Kustiawan, Y.A., Ghauth, K.I.: PhishOFE: A Novel Machine Learning Framework for Real-Time Phishing URL Detection With Optimized Feature Engineering. IEEE Access 13, 169606–169627 (2025)
15. Kustiawan, Y.A., Ghauth, K.I.: Evaluating the Impact of Feature Engineering in Phishing URL Detection: A Comparative Study of URL, HTML, and Derived Features. IEEE Access 13, 126756–126768 (2025)
16. Duarte, J.D., et al.: Machine Learning for Early Detection of Phishing URLs in Parked Domains. IEEE Access 13, 145736–145753 (2025)
17. Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., Weippl, E.: QR Code Security. In: Proc. 8th Int. Conf. on Advances in Mobile Computing and Multimedia (MoMM), pp. 430–435. ACM (2010)
18. Scanzio, S., Rosani, M., Scamuzzi, M., Cena, G.: QR Codes: From a Survey of the State of the Art to Executable eQR Codes for the Internet of Things. IEEE Internet Things J. 11(13), 23699–23710 (2024)
19. Krombholz, K., Frühwirt, P., Kieseberg, P., Kapsalis, I., Huber, M., Weippl, E.: QR Code Security: A Survey of Attacks and Challenges for Usable Security. In: Human Aspects of Information Security, Privacy, and Trust, LNCS 8533, pp. 79–90. Springer (2014)
20. Yong, K.S., Chiew, K.L., Tan, C.L.: A Survey of the QR Code Phishing: The Current Attacks and Countermeasures. In: Proc. 7th Int. Conf. on Smart Computing & Communications (ICSCC), pp. 1–5. IEEE (2019)
21. Focardi, R., Luccio, F.L., Wahsheh, H.A.M.: Usable Security for QR Code. J. Inf. Secur. Appl. 48, 102369 (2019)
22. Wahsheh, H.A.M., Luccio, F.L.: Security and Privacy of QR Code Applications: A Comprehensive Study, General Guidelines and Solutions. Information 11(4), 217 (2020)
23. Dabrowski, A., Krombholz, K., Ullrich, J., Weippl, E.R.: QR Inception: Barcode-in-Barcode Attacks. In: Proc. 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM). ACM (2014)
24. Zhou, A., Su, G., Zhu, S., Ma, H.: Invisible QR Code Hijacking Using Smart LED. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3(3), 1–23 (2019)
25. Akram, M.W., Sood, K., Hassan, M.U., Subba, B.: Exemplifying Emerging Phishing: QR-Based Browser-in-the-Browser (BiTB) Attack. IEEE Netw. Lett. (2025)
26. Sharevski, F., Devine, A., Pieroni, E., Jachim, P.: Phishing with Malicious QR Codes. In: Proc. European Symposium on Usable Security (EuroUSEC). ACM (2022)
27. Geisler, M., Pöhn, D.: Hooked: A Real-World Study on QR Code Phishing. arXiv:2407.16230 (2024)
28. Still, J.D., Morris, T., Edwards, M.: Investigating University QR Code Interactions. In: Int. Conf. on Human-Computer Interaction, pp. 204–214. Springer (2024)
29. Dhamija, R., Tygar, J.D., Hearst, M.: Why Phishing Works. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI), pp. 581–590. ACM (2006)
30. Hong, J.: The State of Phishing Attacks. Commun. ACM 55(1), 74–81 (2012)
31. Sahoo, D., Liu, C., Hoi, S.C.H.: Malicious URL Detection Using Machine Learning: A Survey. arXiv:1701.07179 (2017)
32. Khonji, M., Iraqi, Y., Jones, A.: Phishing Detection: A Literature Survey. IEEE Commun. Surv. Tutor. 15(4), 2091–2121 (2013)
33. Chiew, K.L., Yong, K.S.C., Tan, C.L.: A Survey of Phishing Attacks: Their Types, Vectors and Technical Approaches. Expert Syst. Appl. 106, 1–20 (2018)
34. Zieni, R., Massari, L., Calabrese, M.C.: Phishing or Not Phishing? A Survey on the Detection of Phishing Websites. IEEE Access 11, 18499–18519 (2023)
35. Safi, A., Singh, S.: A Systematic Literature Review on Phishing Website Detection Techniques. J. King Saud Univ. Comput. Inf. Sci. 35(2), 590–611 (2023)
36. Google: Safe Browsing APIs. https://developers.google.com/safe-browsing
37. PhishTank. https://phishtank.org
38. VirusTotal. https://www.virustotal.com
39. Oest, A., Zhang, P., Wardman, B., Nunes, E., Burgis, J., Zand, A., Thomas, K., Doupé, A., Ahn, G.-J.: Sunrise to Sunset: Analyzing the End-to-End Life Cycle and Effectiveness of Phishing Attacks at Scale. In: Proc. 29th USENIX Security Symposium. USENIX (2020)
40. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a Dynamic Reputation System for DNS. In: Proc. 19th USENIX Security Symposium. USENIX (2010)

41. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In: Proc. Network and Distributed System Security Symposium (NDSS). Internet Society (2011)
42. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and Evaluation of a Real-Time URL Spam Filtering Service. In: Proc. IEEE Symposium on Security and Privacy, pp. 447–462. IEEE (2011)
43. Zhauniarovich, Y., Khalil, I., Yu, T., Dacier, M.: A Survey on Malicious Domains Detection through DNS Data Analysis. ACM Comput. Surv. 51(4), Article 67 (2018)
44. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In: Proc. 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '09), pp. 1245–1254. ACM (2009)
45. Blum, A., Wardman, B., Solorio, T., Warner, G.: Lexical Feature Based Phishing URL Detection Using Online Learning. In: Proc. 3rd ACM Workshop on Artificial Intelligence and Security (AISec), pp. 54–60. ACM (2010)
46. Verma, R., Dyer, K.: On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers. In: Proc. 5th ACM Conf. on Data and Application Security and Privacy (CODASPY), pp. 111–122. ACM (2015)
47. Marchal, S., François, J., State, R., Engel, T.: PhishStorm: Detecting Phishing with Streaming Analytics. IEEE Trans. Netw. Serv. Manag. 11(4), 458–471 (2014)
48. Sahingoz, O.K., Buber, E., Demir, O., Diri, B.: Machine Learning Based Phishing Detection from URLs. Expert Syst. Appl. 117, 345–357 (2019)
49. Rao, R.S., Pais, A.R.: Detection of Phishing Websites Using an Efficient Feature-Based Machine Learning Framework. Neural Comput. Appl. 31, 3851–3873 (2019)
50. Le, H., Pham, Q., Sahoo, D., Hoi, S.C.H.: URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. arXiv:1802.03162 (2018)
51. Aljofey, A., Jiang, Q., Qu, Q., Huang, M., Niyigena, J.-P.: An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. Electronics 9(9), 1514 (2020)
52. Bahnsen, A.C., Bohorquez, E.C., Villegas, S., Vargas, J., González, F.A.: Classifying Phishing URLs Using Recurrent Neural Networks. In: Proc. APWG Symposium on Electronic Crime Research (eCrime). IEEE (2017)
53. Yang, P., Zhao, G., Zeng, P.: Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning. IEEE Access 7, 15196–15209 (2019)
54. Asiri, S., Xiao, Y., Alzahrani, S., Li, T.: PhishingRTDS: A Real-Time Detection System for Phishing Attacks Using a Deep Learning Model. Comput. Secur. 141, 103843 (2024)
55. Prasad, A., Chandra, S.: PhiUSIIL: A Diverse Security Profile Empowered Phishing URL Detection Framework Based on Similarity Index and Incremental Learning. Comput. Secur. 136, 103545 (2024)
56. Aljofey, A., Bello, S.A., Lu, J., Xu, C.: BERT-PhishFinder: A Robust Model for Accurate Phishing URL Detection with Optimized DistilBERT. IEEE Trans. Dependable Secure Comput. (2025)
57. Breiman, L.: Random Forests. Mach. Learn. 45(1), 5–32 (2001)
58. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '16), pp. 785–794. ACM (2016)
59. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3146–3154 (2017)
60. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data? In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022)
61. Canali, D., Cova, M., Vigna, G., Kruegel, C.: Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages. In: Proc. 20th Int. Conf. on World Wide Web (WWW), pp. 197–206. ACM (2011)
62. Tan, C.L., Chiew, K.L., Wong, K., Sze, S.N.: PhishWHO: Phishing Webpage Detection via Identity Keywords Extraction and Target Domain Name Finder. Decis. Support Syst. 88, 18–27 (2016)
63. Jain, A.K., Gupta, B.B.: Towards Detection of Phishing Websites on Client-Side Using Machine Learning Based Approach. Telecommun. Syst. 68, 687–700 (2018)
64. Alaca, Y., Çelik, Y.: Cyber Attack Detection with QR Code Images Using Lightweight Deep Learning Models. Comput. Secur. 126, 103065 (2023)
65. Minocha, A., Goyal, A., Gandhi, R.: Recognition of Valid QR Codes with Machine Learning. In: Proc. IEEE 13th Int. Conf. on Communication Systems and Network Technologies (CSNT), pp. 724–730. IEEE (2024)
66. Lourenco, D.O.D.R., Sriraj, M.S., Thambi, K.K., Ranjan, V.: Malicious URLs and QR Code Classification Using Machine Learning and Deep Learning Techniques. In: Proc. 3rd Asian Conf. on Innovation in Technology (ASIANCON), pp. 1–10. IEEE (2023)
67. Menghani, G.: Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. ACM Comput. Surv. 55(12), 1–37 (2023)
68. Bekavac, L.J.L., Mayer, S., Strecker, J.: QR-Code Integrity by Design. In: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, pp. 1–9. ACM (2024)
69. Al-Zahrani, M.S., Wahsheh, H.A., Alsaade, F.W.: Secure Real-Time Artificial Intelligence System Against Malicious QR Code Links. Secur. Commun. Netw. 2021, 5540670 (2021)
70. Marappan, S., Marappan, H.: Enhancing Malicious QR Code Link Detection with Swarm-Intelligent Squirrel-Search Optimized ANN. In: Proc. Int. Conf. on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES). IEEE (2023)
71. Thirumuruganathan, S., Nabeel, M., Choo, E., Khalil, I., Yu, T.: SIRAJ: A Unified Framework for Aggregation of Malicious Entity Detectors. In: Proc. IEEE Symposium on Security and Privacy (SP), pp. 507–521. IEEE (2022)
72. Akhawe, D., Felt, A.P.: Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In: Proc. 22nd USENIX Security Symposium, pp. 257–272. USENIX (2013)
73. Egelman, S., Cranor, L.F., Hong, J.: You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI), pp. 1065–1074. ACM (2008)
74. Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L.F., Downs, J.: Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI), pp. 373–382. ACM (2010)
75. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 1135–1144. ACM (2016)
76. Lundberg, S.M., Lee, S.-I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 4765–4774 (2017)
77. Pedregosa, F., et al.: Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011)
78. Hannousse, A., Yahiouche, S.: Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study. Eng. Appl. Artif. Intell. 104, 104347 (2021)
79. Vrbančič, G., Fister Jr., I., Podgorelec, V.: Datasets for Phishing Websites Detection. Data Brief 33, 106438 (2020)