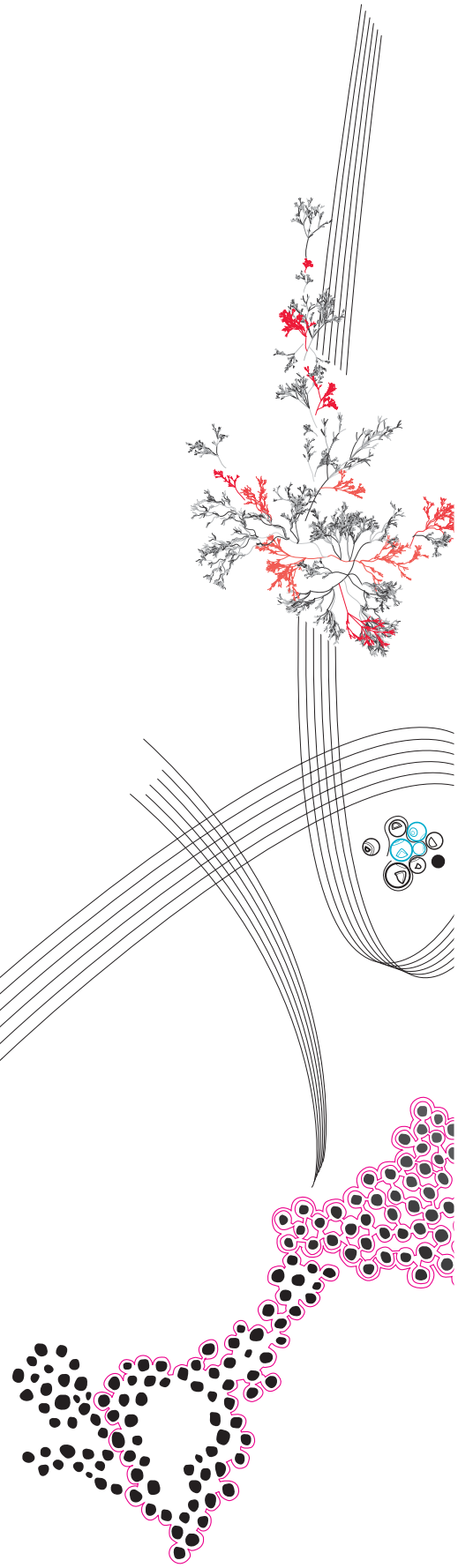


Premaster Computer Science
Literature Review

Real-time Quishing detection using ML techniques

Michal Gorlas
Thijs Besseling

Supervisor: Dr. Dipti K. Sarmah

November, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

List of Tables

List of Figures

Abstract

Quick-Response (QR) Codes - two-dimensional matrix barcodes - have grown popular for their convenience, drawing the attention of cybercriminals who exploit them for phishing - a social engineering practice aiming to convince victims to, for example, enter their bank details on websites controlled by the attackers. This practice, when done using a QR code is called Quishing (QR code phishing). Multiple authors described techniques, such as embedding hidden content, used by cybercriminals to hide dangerous content within QR codes. A couple of approaches in the literature to detect malicious content such as direct malware distribution or 0-day payloads (exploits for not yet patched vulnerabilities), were proposed. This, together with the growing usability of Machine Learning (ML), especially classification algorithms, which appear to be widely used in phishing detection, prompts us to investigate the potential use of ML techniques in addressing Quishing. This paper explores the problem of Quishing, the usefulness of ML in phishing detection, the structural patterns of a QR code, and how they can be used to detect Quishing. We propose a solution against Quishing, real-time ML detection based on anomalies in QR code structures. The proposed solution will enable users to detect Quishing embedded in QR codes.
Keywords: Cyber security, QR codes, Quishing, Machine Learning

1 Introduction

QR codes are a popular and convenient tool for sharing content. Over the years they became common in various areas such as education, business, and marketing. In education QR codes are found useful for academic libraries, providing easier access to digital content [1]. Their usefulness in various advertisement areas has been extensively evaluated in recent years. Endres et al. [2] shows how QR codes may be used to increase the number of answers to a survey, impacting the potential success of an advertisement campaign. Another study, by Meydanoğlu et al. [3] was focused on factors leading to QR codes being scanned by customers, and on how QR codes affect customers' behavior.

As a natural consequence of popularity, they also caught the attention of cybercriminals. Already in 2013, Vidas et al. [4] raised concerns about the potential misuse of this technology to spread phishing, that is, a social engineering practice that aims to convince victims to do certain actions that compromise their cyber safety, such as sharing the bank details on attackers' website, or typing their passwords onto a fake website. The authors also mentioned that already in 2011 over 14 million American users scanned QR codes within only one month, which shows the large number of potential Quishing victims. Additionally, Cargrill et al. [5] highlights the main areas in which QR codes are exploited. Their study also shows that QR codes are perceived as trustworthy by the majority of the questioned audience. They included factors such as the cybersecurity training level of students, the context in which the QR code was used, the false positives rate, and students' familiarity with QR codes. Specific areas of QR code exploitation were evaluated by several authors in the past [6–8]. A more in-depth introduction to the problem of Quishing is given in Section 2.1. Several approaches to creating the QR code scanners, targeting different types of threats were evaluated in the past [6, 9–13]. They are examined in Section 2.3. Most of the mentioned scanners use Machine Learning (ML) based approach to detect phishing and/or different types of malicious content. The usefulness of ML in phishing detection is being addressed in Section 2.2.

The previously mentioned scanners, however, do **not** address the problem of Quishing directly, instead, the focus of these scanners is on detecting various forms of malicious content, such as direct malware distribution, or 0-day payloads - exploits that take advantage of vulnerabilities that have not yet been patched [14]. We see the potential for improvement in Quishing detection, in developing software that utilizes an ML model trained to detect **only** Quishing.

The absence of a recent ML-based tool targeting Quishing detection leads us to formulate research questions, as stated in Section 3. In Section 4, methods used to obtain our findings are outlined, followed by Section 6, where the findings are presented. In Section 7 the findings are discussed, first in Subsection 7.1 we answer our first research question by reviewing existing work focused on structural patterns of a QR code, and how these patterns may be used for Quishing detection. Finally, in Section 8 we present our conclusion and suggestions for future research.

2 Background and Related work

This section discussed the roots of the problem of Quishing, which may be found in Subsection 2.1. The evaluation of using Machine Learning in phishing detection may be found in Subsection 2.2, and the previous approaches proposed in the scientific literature for detecting malicious content, including but not limited to Quishing, in Subsection 2.3.

2.1 The Problem of Quishing

Phishing spread through QR codes, is known under the term "Quishing". Quishing gained the attention of security researchers shortly after the COVID-19 pandemic. Sharevski et al. [15] claim that the reason behind the enhanced attention of security researchers in Quishing has its roots in the growth of the QR codes popularity during the COVID-19 pandemic. Since they became the main method used for URL sharing, the natural consequence was the rise of malicious URLs shared this way, which was noticed by ENISA (European Union Agency for Cybersecurity), in their "Thread Landscape 2023" report based mainly on data from 2022. They outlined Quishing as one of the most common techniques of social engineering used in real-world attacks [16, Ch. 6.5, p. 76].

We found a similar pattern that proves Sharevski et al. [15]'s claim, as shown in Section 6, our search for the literature containing the keyword "quishing" resulted in papers no older than 2022. The search using the keyword "phishing" in combination with "qr code", resulted in a limited number of relevant papers published after 2022. The earliest effort to describe Quishing as a security issue was the paper from Vidas et al. [4], published in 2013. While the term "Quishing" does not appear in their work, they proposed the term "QRishing", which was evaluated later by a few authors between 2013 and 2019 [13, 17, 18]. Both "Quishing" and "QRishing" refer to exactly the same phenomenon - QR code phishing. The term "Quishing", however, was more frequently used in the literature published after 2019 Vidas et al. [4], in their work conducted an experiment on the academic population, investigating how likely the students are to fall for "malicious" QR codes. The authors, however, do not try to obfuscate URL in a similar fashion as the real attacker may do. Another limitation pointed out by the authors is that the tested population may be unrepresentative, given the fact that it was limited to the campus population.

Similar work has been done by Kusyanti and Arifin [17], who investigated factors that affect the likelihood of an individual scanning a QR code. Results of their work showed that there is a direct connection between an individual's attitude, perceived security and risk, and the likelihood of scanning a QR code. Another paper on "QRishing" presented "Anti-Qrishing" detection solution. Latif [13] proposes a mix of Address bar-based and Domain-based approaches to detect "QRishing" in real-time. Their tool showed an accuracy of 98% in the test environment. It can be however less effective against more sophisticated hiding techniques used by attackers these days (which we will cover in more detail in Subsection 2.2). Term "QRishing" was also mentioned by Guarda, Augusto, and Lopes [18] in their "The Art of Phishing" paper. While the paper was more concentrated on Phishing in general, one subsection was dedicated to the raising problem of misuse of QR codes to spread phishing.

Returning to the term Quishing, as already mentioned, it became popular in the scientific literature in 2022. Apart from the already mentioned paper from Sharevski et al. [15], few other authors have put their focus on Quishing. Bekavac, Mayer, and Strecker [19] in their paper, propose "SafeQR" codes, a set of design changes making tampering or changing the QR codes by malicious actors easily recognizable. The authors highlight the current primary defense strategy against Quishing as user education, and propose an alternative approach by securing QR codes by design. While on the conceptual level, the solution proposed by Bekavac, Mayer, and Strecker [19] should make tampering with a QR code significantly more difficult, they mentioned no testing on "real-world" examples of phishing has been conducted.

On the other hand, Amoah and Hayfron-Acquah [9] in their paper first discuss privacy issues that come together with QR codes, and then propose a solution based on Naive Bayes classification and logistic regression algorithms to detect phishing URL obtained from a QR code. Authors, however, do not justify the choice of Naive Bayes classification and logistic regression over other available algorithms such as Random Forest or Support Vector Classifier. They also mention that

the proposed solution may not be accurate against new phishing.
Also in a very recent (2024) paper by Vaithilingam and Shankar [20], authors mention Quishing as one of the threats coming together with QR codes. They highlight that while it is often possible to spot phishing messages by looking at typographical errors for example domain name in the URL address, it is not the case for QR codes, as the URL address, and therefore domain name, is not readable before encoding.

2.2 Machine Learning in Phishing Detection

Machine Learning (ML) is a rapidly evolving field. Applications of ML find themselves in many areas of tech-industry and science, including tasks such as fraud detection or image classification. These types of tasks are resolved by using **supervised learning**, that is when the algorithm is being trained on the dataset that contains the desired solution (so-called *labels*). The learning process of such an algorithm involves calculating and adjusting the error to achieve the desired output by comparing the calculated and predicted outputs [21] [22, p. 7-8].
Supervised Learning has been shown to be useful for phishing detection [12, 23–26]. Gangavarapu, Jaidhar, and Chanduka [26] in their paper point out that ML techniques are successfully used by some of the biggest email providers such as Google (Gmail), Yahoo (Yahoo Mail), and Microsoft (Outlook), for phishing detection on the server side. However, Machine Learning is not the only known technique to detect phishing, Dutta [27] presents three most common types of phishing detection, namely:

- Heuristic/ML approach - usually based on both supervised and unsupervised learning
- Proactive approach - a combination of ML and support system to make predictions against "zero-day" attacks (described more explicitly by Nakamura and Dobashit [28])
- Black/Whitelist - classic approaches based on automatically blocking known phishing websites that found themselves on a particular blacklist

The author argues that the Black/Whitelist approach is inefficient given the growing amount of phishing websites. The author, however, do not address the benefits of integrating techniques, especially for 0-day attacks where relying only on ML could be insufficient.
A similar conclusion is stated by Mathankar et al. [23], who state in their paper that most of the blacklist-based solutions are not effective against modern phishing attacks. They attribute this to the ease of registering new names, and challenges associated with updating blacklists. Also similarly to Dutta [27], they propose an ML approach, namely an ML model trained to detect phishing websites.
Another example of similar argumentation was presented by Uplenchwar et al. [24], who proposed a phishing attack detection system for text messages. The authors decided to design the system based mainly on ML due to its advantages over more "traditional approaches" such as the blacklist approach. They argue that the blacklist approach takes more time for detection and gives considerably higher false rates, compared to the ML-based approach. However, in their proposed solution, authors supported their detection system by doing additional checks using a blacklist approach.

2.3 "General Purpose" QR Codes Scanners

As mentioned in Section 1, several authors in the past have proposed solutions to detect malicious content in QR codes [6, 7, 9–13, 20, 29]. This subsection reviews the solutions proposed by those authors.
Rafsanjani et al. [6], propose "QsecR" - a QR code scanner for Android to detect malicious URL addresses. The authors define a detection framework that consists of three phases: check for redirection, feature classification, and then detection itself. Their detection method is not based on an ML classification algorithm but rather on the static classification of predefined features, such as blacklist feature, lexical feature, host-based feature, and content-based feature. To evaluate the performance of the proposed scanner, the authors used the following performance metrics: **Accuracy, Precision, False Positive Rate, Recall,** and **F1 score**.

These metrics also belong to the frequently used metrics for the ML algorithms evaluation [22]. The results shown by Rafsanjani et al. [6] are promising, with all measures around 93%. These results were achieved using a dataset of 4000 URLs, including 2000 benign and 2000 malicious samples. The authors mention the need for improvement in the response time of the scanner, and performance in the context of device resources used while performing a scan.

Minocha, Goyal, and Gandhi [7] propose a solution that utilizes a deep learning model to detect malicious and benign QR codes. Their system employs Convolutional Neural Networks (CNNs) for classifying QR code images after training the model on a dataset of 10,000 QR codes. The authors evaluate various CNN architectures, including ResNet50, MobileNetV3, and InceptionV3, to identify the most effective model. According to their results, InceptionV3 achieved the highest accuracy, with a score of 98.13%. This success is due to its inception modules, which capture features at different levels of detail, and extra classifiers added in the middle layers to make training more stable and help the model work well with different types of QR codes.

Similarly, Alaca and Çelik [29] suggests a hybrid approach that combines optimization-based feature selection with lightweight deep learning models to detect fake QR codes. In particular, the authors extract features from QR code pictures created from the CSE-CIC-IDS2018 dataset, which contains a variety of attack types including DDoS and FTP-BruteForce, using the MobileNetV2 and ShuffleNet CNN models. The best characteristics from these models are chosen using Harris Hawk Optimization (HHO), and then the Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) algorithms are used for classification. The suggested hybrid method outperforms individual CNN models with a classification accuracy of 95.89%. Note that potential dataset bias may exist, as the paper lacks details on data validation, such as cross-validation or additional testing sets.

Another solution, BarAI is an ML-based barcode scanner that can identify malicious URLs encoded in both 1D and 2D codes. It was proposed by Al-Zahrani et al. [10]. It is created to prevent barcode injection attacks, in which a malicious 1D barcode is placed on a genuine QR code. BarAI uses classifiers like Naive Bayes, Support Vector Machine, Logistic Regression, K-Nearest Neighbors, and Decision Trees to scan the lexical features of URLs, including length, structure, and the presence of odd letters or suspicious characters. The Decision Tree classifier achieved the highest accuracy of 90.243% because to its hierarchical decision-making, which successfully captures nonlinear correlations between features, as well as its ability to withstand noise and ambiguous regions in the data.

Song et al. [11] propose the QRFence framework, which is based on machine learning and used for the identification of malicious URLs that are embedded in the QR code, with a focus on mitigating threats associated with QR codes-in particular, malicious link dissemination. Unlike other existing approaches, QRFence performs threat intelligence in real-time while the QR code is being deciphered. This detection framework leverages several classification algorithms, namely IBk, J48, and Logistic Regression, trained with features extracted from URL, HTML, and JavaScript. In this scenario, the training datasets had a great number of malicious and benign samples each, at which point, after the modeling was done, it achieved an accuracy rate of 93.20%. Contrary to other solutions, this system incorporates detection permissions and does not rely on browser plugins. Thus, this guarantees heightened security independent of the classic web security mechanisms. QRFence does not rely on the black/white list approach. According to the authors, this makes it more adaptive to emerging real-world attacks.

Amoah and Hayfron-Acquah [9] developed a machine learning-based phishing detection model to address security concerns related to QR code usage. The authors implemented the model using Naive Bayes and logistic regression classifiers, training these classifiers on URLs extracted from QR codes to classify them as phishing or legitimate. They tested the model on a dataset of 500,000 URLs and achieved an accuracy of 96.47%. Unlike our approach, which emphasizes feature extraction from the structural patterns of the QR code itself without decoding, this solution focuses on analyzing the decoded content of the QR code to detect phishing URLs, thereby avoiding any potential exposure of harmful content to the user.

Vaithilingam and Shankar [20] created an ML-based system that incorporates Natural Language Processing (NLP) to improve QR code security by preventing attacks such as QRLjacking (QR code login hijacking) and Quishing (QR code phishing). The system uses a mix of machine learning methods, such as logistic regression to categorize URLs as dangerous or valid based on attributes

including length, special characters, domain names, and host names, and anomaly detection to find odd patterns in URLs. Furthermore, Naive Bayes is utilized to improve the classification by estimating the likelihood that a URL is dangerous based on its textual characteristics. Together, these algorithms are able to identify dangerous patterns, flag questionable QR codes, and provide users with real-time feedback. In order to provide a thorough defense against a variety of QR code-based assaults, the framework additionally uses a ranking aggregation and scoring process that assesses each QR code's security according to its content and user-defined criteria. Another example of a solution for QR code phishing in real time was proposed by Latif [13] they combined two main approaches: Address Bar-Based and Domain-Based, with additional validation from blacklists. The Address Bar-Based approach verifies certain elements in the URL, such as the usage of shortened links, 'http' instead of 'https', suspicious symbols like '@', file extensions like '.exe' or '.apk', and overly long URLs or IP addresses instead of the domain name. In the Domain-Based approach, the characteristics of a domain are considered for phishing site checks, such as domain age using WHOIS data, traffic ranking via Alexa, and blacklist verification with Phistank API. The solution enforces condition validation of indicators of phishing and online checks on domain legitimacy to ensure both static and real-time checks. QR codes are accordingly classified as safe, suspicious, or phishing by the model, which achieved an accuracy of 98.4% and was tested with 500 QR codes. One of the older solutions QRphish developed by Alnajjar et al. [12] in 2016. QRphish represents a machine learning-based system for detecting phishing URLs embedded in quick QR codes. It developed its classifier by fusing three different types of features, that is, lexical-based features related to the URL which can be suspicious characters, IP address presence, and number of dots; host-based features, such as domain age and page rank; and finally, QR code-specific features refer to content length/type and error correction level. It uses the Naive Bayes classifier, trained with both legitimate and phishing URLs, in order to classify the content of the QR code as either safe or malicious. Based on these features, QRphish checks, in real-time and upon a scan, a given QR code's URL and decides if it contains phishing. The implemented API allows for integration into a mobile application to show runtime phishing detection efficiently and fast. QRphish resulted in a promising accuracy score of 93.34%.

3 Hypothesis and Research Questions

As mentioned in Section 1, existing real-time detection scanners that use ML, do not target Quishing explicitly. This leads to our hypothesis that a QR code scanner that uses ML, specifically targeting Quishing will outperform more general solutions. Thus our main research question states as follows: **"Can ML techniques help detecting Quishing in real-time?"**. Answering this question will require evaluating the following research questions:

Research Question (RQ1) - "What are the features of a QR code that can be used to determine whether it contains Quishing content or not?"

- Sub-question (RQ1-SQ1): "What are the structural patterns of a QR code?"
- Sub-question (RQ1-SQ2): "Which of these patterns can be used to train the model for Quishing detection?"

Research Question (RQ2) - "How can the features be used to train the ML model for Quishing detection?"

- Sub-question (RQ2-SQ1): "How can the features be extracted from the QR code image in order to train an accurate ML model for Quishing detection?"
- Sub-question (RQ2-SQ2): "Which ML classification algorithm performs best in terms of performance measures for Quishing detection when trained on a dataset of Quishing and benign QR Codes?"

Research Question (RQ3) - "How the ML model trained for Quishing detection can be implemented within an application to detect Quishing?"

In this report, we aim to answer RQ1, while RQ2 and RQ3 will be addressed in the next phase of our work, as outlined in our Research Proposal.

4 Methodology

This section outlines the methodology used to obtain the literature reviewed in section 6 and 7. We conducted the literature review using rapid literature review (RLR) because of its flexibility, which allowed us to work efficiently within our specified timeframe. Studies show RLRs often yield results consistent with systematic reviews, demonstrating their reliability as a synthesis tool [30]. Although the specific term "RLR" was only introduced in 2021 [31], the concept has been discussed and utilized by various authors over the past two decades [30, 32]. These Papers [30–32], highlighted the applicability of RLR for the cases where similar transparency and bias minimization as with systematic literature review process is needed but within a limited time-frame [33]. This makes it a suitable choice for our work given the circumstances.
We conduct the RLR by following the steps mentioned by Moons, Goossens, and Thompson [30], Smela et al. [31], and Dobbins [34] in their papers:

1. **Develop a Search Protocol**: we define a set of keywords in combination with filters, as may be seen at Table 1 in Subsection 5.1

2. **Conduct Literature Search:** The targeted search was done using FindUT and Google Scholar search engines [35, 36]. These engines provide access to multiple scientific databases, such as IEEE Xplore, Springer, Scopus, and ScienceDirect [37–40]. The search engines were chosen based on the University of Twente's recommendations, as they cover many of the most important databases and resources in Computer Science [41].

3. **Select Studies**: Lastly, we selected our findings based on their relevance to our research questions. Additionally, we ensured that each paper had undergone peer review by verifying the peer-review status of the journal in which it was published.

5 Proposed solution

In order to answer RQ1 - "What are the features of a QR code that can be used to determine whether it contains Quishing content or not?", and its sub-questions, we propose the following:

1. **Addressing SQ1- RQ1**: We hypothesize that structural patterns of a QR code may reveal anomalies (such as unjustified inconsistency of codewords length) that can be used for Quishing detection, therefore our proposed solution to answer RQ1, starts with the extensive review of scientific literature on QR codes using the targeted search with criteria mentioned in 5.1. The selected papers about the structural pattern of a QR code are discussed in Section 7.1.

2. **Addressing SQ2- RQ1**: Building upon the knowledge of structural patterns of a QR code, we further review the literature that demonstrates how these patterns have been exploited for benign (such as secret sharing) as well as malicious purposes (such as Quishing). Methods presented in the literature used to exploit the QR code patterns are discussed in Section 7.1.2.

5.1 Search Criteria

The keywords for the targeted search, along with the database sources, language, and time frames, are listed in Table 1 below. These keywords were derived from our research questions, focusing on key concepts such as "Quishing" and what it was called before it became its own topic "QR code phishing" based on the main research question.

In the case of RQ1, we targeted the structure of QR codes and came up with keywords such as "QR Code Patterns," and "Fake QR Code" In the case of RQ2, we came up with keywords such as "Machine Learning Phishing" illustrate the problem, we also included terms like "Advertisement QR Codes" regarding the use of QR codes to give inside in how QR codes are used. We combined

these key terms during the search to identify literature relevant to our research questions. As stated in our proposal, due to time constraints, we limited our research to English-language literature. Additionally, because Quishing became more popular after the COVID-19 pandemic, the time frame for the papers was limited to those published from 2019 onwards. However, for pre-pandemic solutions to QR code phishing, we included literature from as early as 2013, when one of the first papers to specifically address QR code phishing as a distinct malicious attack came up with the term 'QRishing'. [4].

This method, as mentioned in the proposal, is used by multiple universities to identify the keywords for targeted searches [42–46]. When using Google Scholar, the sort by relevancy filter is used and the first 10 pages were then reviewed [47].

TABLE 1: Targeted search criteria (updated at 29.10.2024)

Keyword	**Database**	**Language**	**Time frame**	**Total results**[b]	**Selected papers**
"quishing"	Scholar	English	2019-now	798	6
"phishing kw[a]:qr code"	FindUT/Scholar	English	none	2240	8
"qrishing"	FindUT	English	2013-2019	20	4
"ML phishing"	FindUT	English	2019-now	518	5
"QR codes academic"	Scholar	English	none	24	1
"advertisement kw[a]:qr code"	FindUT/Scholar	English	none	19100	2
"qr codes kw[a]: patterns"	Scholar	English	none	103000	1
"Fake QR codes"	Scholar	English	none	20000	2

[a] "kw" stands for "keyword" here, used in FindUT to put search engine's focus on the occurrence of a given phrase in papers' content.
[b] This number includes duplicates shown by search engines, as well as papers that were classified as not relevant or bad quality, as further described in 5.2.

5.2 Quality assurance

To check the reliability of the found literature, we conducted multiple checks. We used checks from the CRAAP list to check the literature quality, as outlined by Meriam Library [48]. Specifically, we did the checks for peer-review to check if it was reviewed by experts, a relevance check if it has a similar topic as our research topic, and a date check to check the timeliness of the research. Additionally, we checked the citation count, considering how many other papers have cited the study to check the trustworthiness and influence of the paper. The checks can be found below:

1. Each paper's peer-review status was checked by visiting the publication's website to make sure it followed a peer-review process.

2. To verify the relevancy with our research topics, we review each paper's abstract, introduction, and keywords. We focused on keywords from the search criteria table 1, to identify papers involving Quishing solutions or techniques relevant to detection.

3. Applying time frame filters from 2019 onward to capture current solutions and techniques for Quishing detection, while including foundational studies from 2013 to trace the development of QR code phishing (see: 5.1).

4. Checking the citation count to evaluate the influence of the paper in its field [49].

5. To avoid overall bias, we gathered sources from multiple databases: IEEE Xplore for technical research, Springer for engineering, Scopus for interdisciplinary coverage, and ScienceDirect for scientific studies [37–40]. This triangulation helped us see more perspectives of solutions. [50]."

6 Results

This section presents the results gathered using the targeted literature search described in Section 4, with the search criteria outlined in Section 5.1, and verified using the checklist in Section 5.2. We gathered our literature search review in a spreadsheet format with the following columns: name, date, citation count, key terms, relevance, limitations, and the URL to the paper. The complete table with the results of our search, may be found in the Gitlab repository dedicated to this report[51].

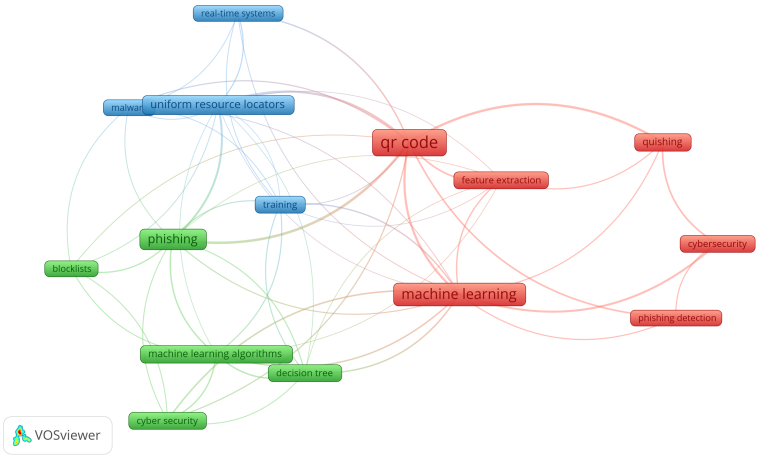


FIGURE 1: Visualization based on collected literature sources, showing the used keywords in each source and the relationship.

As shown in the visualization in Fig. 1, QR codes are a topic with a connection to machine learning techniques, which in our found literature are frequently applied to detect threats like phishing. However, while there are machine learning solutions linked to QR code security, most of these algorithms are generally focused on phishing detection using URLs classification as shown in Section 2.3 and less specifically targeted at detecting 'Quishing' based on feature extraction. This finding addresses the identified gap in focused research on QR code phishing detection, which remains uncommon. Most research explores a variety of malicious uses of QR codes, while only a few studies specifically target QR code phishing detection [4, 23]. Most of them extract the URL from the QR code and perform phishing detection on the URL by using methods such as blacklisting or machine learning algorithms such as KNN, Random Forests, SVM, XGBoost, etc [6, 7, 23, 24]. Results: Among our results, we found about 25 papers containing information about QR codes. Some studies explore specific methods to improve QR code security by embedding cryptographic mechanisms to safeguard QR code content [8, 52]. We found an article demonstrating methods for creating fake QR codes [53], which led us to discover that the error-correction feature in QR codes can also be used to hide secrets [54, 55]. The first article [54], provides thorough information on the structural components of QR codes. Most of these research are based on either data extraction or injection, which is useful in understanding the structure of QR codes. The papers discussing machine learning algorithms for detecting malicious patterns or scanning URLs are useful for deciding which machine learning approaches to use in our research and the training of those models for Quishing detection. This helps us gain a better understanding of the available patterns, which correlates to our first sub-question mentioned in Section 3 of SQ1: "What are the structural patterns of a QR code?". The Table 2 below, shows our findings for each keyword. Not all papers are included, as some papers were found based on very specific searches about topics we learned form the papers shown in Table 2.

TABLE 2: Results

Keyword	Papers found per keyword
"quishing"	[9, 12, 15, 18–20]
"phishing kw[a]:qr code"	[5–8, 10, 11, 13, 29]
"qrishing"	[4, 13, 17, 18]
"ML phishing"	[12, 23–26]
"QR codes academic"	[1]
"advertisement kw[a]:qr code"	[2, 3]
"qr codes kw[a]: patterns"	[56]
"Fake QR codes"	[53, 57]

7 Discussion

This section discusses our findings obtained using methods described before in Section 4. As mentioned already in 1, Subsection 7.1 will be dedicated to answering our RQ1, namely to discuss existing literature about QR code patterns and their relevancy for Quishing detection.

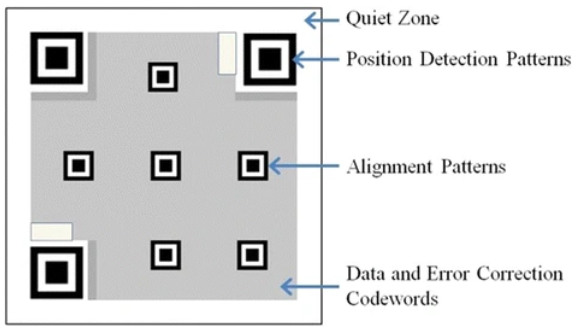
7.1 Structural Patterns of a QR Code

In order to create a Quishing detection solution, is essential to gain knowledge about the structure of a QR code. It will be used for feature selection in the later stages of our work. This Subsection consists of two parts. The first part addresses Sub-Question 1 of the RQ1, the structural patterns of a QR code are discussed, and the second part addresses Sub-Question 2 of the RQ1 - the usefulness of certain QR patterns for the sake of ML learning training is being discussed.

7.1.1 Structural Patterns - SQ1- RQ1

Lin and Chen [54] in their work focus on proposing a solution for sharing secrets through QR codes. Their work is, however, relevant to the scope of our work, as they excessively described the construction of a QR code. The particular zones of a QR code are shown in Fig 2.

FIGURE 2: Areas of a QR code



Note. From "High payload secret hiding technology for QR codes." J Image Video Proc. 2017, 14 (2017). Lin and Chen [54]. CC BY 4.0.

A QR code consists of:

- Quiet Zone - enclosing area on all four surfaces, does not contain any meaningful data
- Position Detection Patterns - describe the position and rotational orientation of the QR code, which appears in the top right, left, and bottom left corners

- Alignment Patterns - similar to the Position Detection Patterns, but smaller, used by scanners to recognize the position
- Data and Error Correction Codewords - modules that store binary values in 8-bit parts (codewords).

Additionally (not shown in Fig. 2), a QR code contains also: **format information** - which contains information about error correction levels, **version information**, **separators** - sections alongside the position detection patterns, used to separate these from the encoding region [56].
Lin and Chen [54] highlight the significance of the error correction codewords. They outline four correction levels of QR code:

- L - with recovery capacity of 7%
- M - with recovery capacity of 15%
- Q - with recovery capacity of 25%
- H - with recovery capacity of 30%

As mentioned by the authors, these percentage rates were defined as ISO standard[58]. The percentage here shows how much of a QR code can be unreadable or damaged while the data is still decodable. Mishra and Mathuria [56] show the different recovery capacities for the "L" level, namely 5% instead of 7%. The remaining error correction levels are consistent with the ones shown by Lin and Chen [54].

7.1.2 Important QR codes patterns for training the ML model - SQ2- RQ1

Already in 2013, Wu, Lin, and Wong [52] proposed a way of embedding data to hide the existence of a QR code. Authors propose a steganographic scheme to "camouflage" the appearance of a QR code utilizing techniques of edge detection and VQ compression.
In much more recent work, Koptyra and Ogiela [8] presents another steganographic technique of embedding information in QR codes, which uses segment manipulation. The authors present stages of QR code creation, namely:

1. Data Analysis - based on the input the encoding mode is determined
2. Data Encoding
3. Error Correction Coding - done using Reed-Solomon Codes [59], exploitation of this particular mechanism was also evaluate by Chow et al. [55], and will be discussed later in this Subsection
4. Structure Final Message
5. Module Placement in Matrix
6. Data Masking
7. Format and Version Information

Later, it is mentioned that secret embedding is being done during data analysis and data encoding stages. The algorithm presented by the authors uses the fact that it is allowed to have empty segments in the QR code. It is worth mentioning that while the algorithm proposed by authors [8] is primarily for information hiding in the sense of information protection, a similar technique may be as well used by malicious actors.
The secret embedding method presented by Lin and Chen [54], also consists of embedding the secret into the data codewords of a QR code. The procedure starts by determining the "tolerant capacity", that is, the value based on the QR version and error correction level. Later the data codeword is divided, producing pairs of data modules. Two of these pairs are randomly (based on a secret key) chosen to be appended with secret bits. The whole process is repeated as long as needed for embedding the complete secret. Extraction is based on a secret key that was previously used for the selection of pairs to be appended. The authors mention that the method they propose

can be easily applied to QR applications.
Similarly, Chow et al. [55] in their work also propose a method of embedding by exploiting an error correction mechanism, as well as they discuss similar approaches doing similar things. The authors also describe the structure of a QR code, which was already shown in Fig 2. In their proposed scheme for secret sharing, Chow et al. [55] makes use of manipulating codewords, by exploiting the error redundancy in the QR code structure. Their approach is to distribute a QR code with a secret message by encoding its information into some number of QR code shares, and recover the message by XORing codewords modules contained in the encoding region and adding the function patterns. The main goal of this approach, is again, to securely share some messages using a QR code. However, the error correction mechanism can also be used by malicious actors, as shown in the next paper we are going to discuss.
Takita, Okuma, and Morii [53] in their work put their focus on proposing the construction scheme of a fake QR code that won't be trivial to spot. The authors create a fake QR code that shows a benign URL with a high probability, and a malicious URL with a small probability. That is, if a potential victim read the QR code again, they will likely see a benign URL. The proposed scheme is based on the error correction mechanism, the QR code outputs two different data by creating a miscorrection. The authors describe how the miscorrection is done while using the already mentioned Reed-Salomon Code. They outline what is the condition for the occurrence of miscorrection, namely:

- The minimum distance between two codewords is given by $d = n - k + 1$ when using (n, k) [1] Reed-Salomon Code, and the number of correctable errors is given by $t = \lfloor (d-1)/2 \rfloor$.
- The difference between two codewords is denoted as $d(c_A, c_B)$, and if c_A approaches c_B within $l = d(c_A, c_B) - t$, the general formula for the probability of occurrence of a miscorrection is:

$$P_m = \binom{d(c_A, c_B)}{l} \cdot \left(\frac{1}{2^8 - 1}\right)^l \tag{1}$$

[2]

- When decoding the QR code, an error occurrence in the symbol can be described as reading the black module as white or the other way around, authors point out that codeword (usually) does not introduce errors when reading an undamaged QR code.

The authors then present two methods to increase the probability given by Eq. 1. The first of the proposed methods can be described as adding noise to the corresponding module of a QR code. The second of the methods to increase the probability of a miscorrection presented by Takita, Okuma, and Morii [53], makes use of a so-called "pad codeword" - that is, a codeword used to fill up the space, in case the data allocated in codeword does not fill up the full capacity of the 8-bits allocated for each codeword. Pad codeword may be seen as a placeholder for codewords that do not use all the space given. Obviously, such codewords do not contain any significant information and therefore may be easily manipulated without impacting the actual information. The method presented by the authors consists of altering the pad codewords in order to minimize the distance between codewords that contain actual data, and therefore increase the probability of miscorrection.
The techniques to detect a QR code created using such methods were proposed by Ohigashi et al. [57]. They propose to base the detection on the number of errors, corrected errors, and the symmetry in the location of the errors. The authors however did not attempt to utilize ML techniques for the sake of detecting fake QR codes. Their conclusions about features that can be used for the detection, together with methods proposed by Takita, Okuma, and Morii [53], leads us to consider noise in Data and Error Correction Codewords, as an important feature for our model to be trained on. Similarly, the frequency of error-correction activation when scanning the code (corrected errors), as well as unusual and inconsistent lengths of codewords, especially unusual sizes of pad codewords - such as pad codeword that do not necessarily fill up the space as intended,

[1] encodes k symbols into codeword with n symbols[53]
[2] $\binom{\cdot}{\cdot}$ is the binomial coefficient

are features that should be considered by our model while making predictions.
Therefore, addressing the SQ1- RQ1, the features of a QR code that can be used to train the model are:

- amount of noise in Data and Error Correction Codewords (see Fig 2),
- frequency of error-correction activation when scanning the code,
- inconsistent lengths of codewords, especially pad codewords.

7.2 Limitations

For the literature review, we did not do a background check on the authors' qualifications or affiliations (Authority), nor did we assess the authors' purposes due to time constraints. The model we are going to train in the next phase of our work, will not be targeting all types of malicious URLs. Our model, and research in general, focus purely on Quishing - phishing content embedded in QR codes. As the consequence of focusing on features that can be extracted from a QR code **without extracting the data**, that is, without extracting the URL, our model will not be classifying the QR code based on the features of the URL address. This approach was however already addressed in the model proposed by Amoah and Hayfron-Acquah [9], and Alnajjar et al. [12], as described before in Section 2.3. This opens a gap for future work that concentrates on how different types of malicious content are embedded/hidden using similar techniques as the ones described in already mentioned Section 2.3. For scanners targeting all different types of malicious URLs distributed through QR codes, we recommend the reader to review previous papers 2.3 [10–12, 23, 27].

8 Conclusion and future work

In this paper, we highlighted the problem of Quishing, which has become more frequently discussed in recent scientific literature. We went through literature talking about **why** Quishing became a serious security issue, and what are the proposed mitigation techniques. Next, we took a closer look at the literature, in which authors proposed QR code scanners for malicious content detection. While these solutions gave promising results, neither of the scanners is focused on Quishing, but on different types of malicious content (such as direct malware or 0-day payloads distribution), which may lead to a lower detection rate against Quishing attacks, as we have also stated in our hypothesis.
The review of existing literature about QR codes and summarizing the most important structural patterns of QR codes, that may be useful for Quishing detection, allowed us to put the focus on Data and Error Correction Codewords as the main feature to train the model on.
We have also reviewed how ML has been used to mitigate phishing, which leads us to the idea of using a similar approach to detect Quishing. We have also reviewed how several authors in the past made use of ML to detect malicious content in QR codes. This information is what we are going to base our answer for RQ2 and its sub-questions. In the next phase of our work, we are going to extract the features of a QR code, and use them to train models that utilize classification algorithms that have been proven in phishing detection for detecting Quishing, and evaluate their performance.
That will allow us to develop a proof-of-concept of a real-time ML-based application to detect Quishing, and therefore provide an answer to the RQ3.

9 Acknowledgments

During the preparation of this work, the authors used Grammarly Spell Checker, in order to correct typos and grammar mistakes. After using this tool, the authors reviewed the content as needed and takes full responsibility for the content of the work.

References

[1] M. Paul and S. Naikar. "Innovative Use of QR Codes in Academic Libraries: Benefits and Challenges". In: *Journal of Emerging Technologies and Innovative Research* 11 (June 2024), pp. 565–572.

[2] K. Endres, E. O. Heiden, K. Park, M. E. Losch, K. K. Harland, and A. L. Abbott. "Experimenting with QR Codes and Envelope Size in Push-to-Web Surveys". In: *Journal of Survey Statistics and Methodology* 12.4 (Apr. 2023), pp. 893–905. ISSN: 2325-0992. DOI: 10.1093/jssam/smad008.

[3] E. S. B. Meydanoğlu, A. M. Çilingirtürk, S. Böhm, and M. Klein. "QR code advertising: a cross-country comparison of Turkish and German consumers". In: *International Journal of Internet Marketing and Advertising* 12.1 (2018), pp. 40–68. DOI: 10.1504/IJIMA.2018.089201.

[4] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. F. Cranor, and N. Christin. "QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks". In: *Financial Cryptography and Data Security*. Ed. by A. A. Adams, M. Brenner, and M. Smith. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 52–69. ISBN: 978-3-642-41320-9. DOI: 10.1007/978-3-642-41320-9_4.

[5] K. Cargrill, T. Abegaz, L. C. Parra, and R. DaSouza. "Scan Me: QR Codes as Emerging Malware Delivery Mechanism". In: *Proceedings of the Future Technologies Conference (FTC) 2023, Volume 2*. Springer Nature Switzerland, 2023, pp. 611–617. ISBN: 9783031474514. DOI: 10.1007/978-3-031-47451-4_44.

[6] A. S. Rafsanjani, N. B. Kamaruddin, H. M. Rusli, and M. Dabbagh. "QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework". In: *IEEE Access* 11 (2023), pp. 92523–92539. DOI: 10.1109/ACCESS.2023.3291811.

[7] A. Minocha, A. Goyal, and R. Gandhi. "Recognition of Valid QR Codes with Machine Learning". In: *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*. 2024, pp. 724–730. DOI: 10.1109/CSNT60213.2024.10546171.

[8] K. Koptyra and M. R. Ogiela. "Information Hiding in QR Codes using Segment Manipulation". In: *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 2024, pp. 397–400. DOI: 10.1109/PerComWorkshops59983.2024.10502885.

[9] G. A. Amoah and J. Hayfron-Acquah. "QR Code security: mitigating the issue of quishing (QR Code Phishing)". In: *International Journal of Computer Applications* 184.33 (2022), pp. 34–39. DOI: 10.5120/ijca2022922425.

[10] M. S. Al-Zahrani, H. A. M. Wahsheh, F. W. Alsaade, and N. Saxena. "Secure Real-Time Artificial Intelligence System against Malicious QR Code Links". In: *Sec. and Commun. Netw.* 2021 (2021). ISSN: 1939-0114. DOI: 10.1155/2021/5540670.

[11] J. Song, K. Gao, X. Shen, X. Qi, R. Liu, and K.-K. R. Choo. "QRFence: A flexible and scalable QR link security detection framework for Android devices". In: *Future Generation Computer Systems* 88 (2018), pp. 663–674. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.05.082.

[12] A. Alnajjar, M. Anbar, S. Manickam, O. Elejla, and H. El-Taj. "QRphish: An Automated QR Code Phishing Detection Approach". In: *Journal of Engineering and Applied Sciences* 11 (July 2016), pp. 553–560. DOI: 10.3923/jeasci.2016.553.560.

[13] K. Latif. "Anti-Qrishing Real-Time Technique on the QR Code Using the Address Bar-Based and Domain-Based Approach on Smartphone". In: *International Journal of Cyber-Security and Digital Forensics* 8 (Jan. 2019), pp. 134–143. DOI: 10.17781/P002571.

[14] X. Wang, K. Sun, A. Batcheller, and S. Jajodia. "Detecting "0-Day" Vulnerability: An Empirical Study of Secret Security Patch in OSS". In: *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2019, pp. 485–492. DOI: 10.1109/DSN.2019.00056.

[15] F. Sharevski, A. Devine, E. Pieroni, and P. Jachim. "Phishing with Malicious QR Codes". In: *Proceedings of the 2022 European Symposium on Usable Security*. EuroUSEC '22. Karlsruhe, Germany: Association for Computing Machinery, 2022, pp. 160–171. ISBN: 9781450397001. DOI: 10.1145/3549015.3554172.

[16] European Union Agency for Cybersecurity. *ENISA Threat Landscape 2023*. Oct. 19, 2023. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023.

[17] A. Kusyanti and A. Arifin. "QRishing: A User Perspective". In: *International Journal of Advanced Computer Science and Applications* 8.10 (2017). DOI: 10.14569/IJACSA.2017.081039.

[18] T. Guarda, M. F. Augusto, and I. Lopes. "The Art of Phishing". In: *Information Technology and Systems*. Ed. by Á. Rocha, C. Ferrás, and M. Paredes. Cham: Springer International Publishing, 2019, pp. 683–690. ISBN: 978-3-030-11890-7. DOI: 10.1007/978-3-030-11890-7_64.

[19] L. J. L. Bekavac, S. Mayer, and J. Strecker. "QR-Code Integrity by Design". In: *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI EA '24. New York, NY, USA: Association for Computing Machinery, 2024. ISBN: 9798400703317. DOI: 10.1145/3613905.3651006.

[20] S. Vaithilingam and S. A. M. Shankar. "Enhancing Security in QR Code Technology Using AI: Exploration and Mitigation Strategies". In: *International Journal of Intelligence Science* 14.02 (2024), pp. 49–57. DOI: 10.4236/ijis.2024.142003.

[21] R. Pugliese, S. Regondi, and R. Marini. "Machine learning-based approach: global trends, research directions, and regulatory standpoints". In: *Data Science and Management* 4 (2021), pp. 19–29. ISSN: 2666-7649. DOI: https://doi.org/10.1016/j.dsm.2021.12.002.

[22] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022. ISBN: 9781492032618.

[23] S. Mathankar, S. R. Sharma, T. Wankhede, M. Sahu, and S. Thakur. "Phishing Website Detection using Machine Learning Techniques". In: *2023 11th International Conference on Emerging Trends in Engineering I& Technology - Signal and Information Processing (ICETET - SIP)*. 2023, pp. 1–6. DOI: 10.1109/ICETET-SIP58143.2023.10151640.

[24] S. Uplenchwar, V. Sawant, P. Surve, S. Deshpande, and S. Kelkar. "Phishing Attack Detection on Text Messages Using Machine Learning Techniques". In: *2022 IEEE Pune Section International Conference (PuneCon)*. 2022, pp. 1–5. DOI: 10.1109/PuneCon55413.2022.10014876.

[25] M. M. Uddin, K. Arfatul Islam, M. Mamun, V. K. Tiwari, and J. Park. "A Comparative Analysis of Machine Learning-Based Website Phishing Detection Using URL Information". In: *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. 2022, pp. 220–224. DOI: 10.1109/PRAI55851.2022.9904055.

[26] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka. "Applicability of machine learning in spam and phishing email filtering: review and approaches". In: *Artificial Intelligence Review* 53.7 (Oct. 2020), pp. 5019–5081. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09814-9.

[27] A. K. Dutta. "Detecting phishing websites using machine learning technique". In: *PLoS ONE 16(10): e0258361* (2021). DOI: 10.1371/journal.pone.0258361.

[28] A. Nakamura and F. Dobashit. "Proactive Phishing Sites Detection". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. 2019, pp. 443–448. ISBN: 978-1-4503-6934-3.

[29] Y. Alaca and Y. Çelik. "Cyber attack detection with QR code images using lightweight deep learning models". In: *Computers & Security* 126 (2023), p. 103065. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2022.103065.

[30] P. Moons, E. Goossens, and D. R. Thompson. "Rapid reviews: the pros and cons of an accelerated review process". In: *European Journal of Cardiovascular Nursing* 20.5 (May 2021), pp. 515–519. ISSN: 1474-5151. DOI: 10.1093/eurjcn/zvab041.

[31] B. Smela, M. Toumi, K. Świerk, C. Francois, M. Biernikiewicz, E. Clay, and L. Boyer. "Rapid Literature Review: Definition and Methodology". In: *Journal of Market Access I& Health Policy* 11.1 (2023). ISSN: 2001-6689. DOI: 10.1080/20016689.2023.2241234.

[32] R. Ganann, D. Ciliska, and H. Thomas. "Expediting systematic reviews: methods and implications of rapid reviews". In: *Implementation Science* 5.1 (July 2010), p. 56. ISSN: 1748-5908. DOI: 10.1186/1748-5908-5-56.

[33] Y. Xiao and M. Watson. "Guidance on Conducting a Systematic Literature Review". In: *Journal of Planning Education and Research* 39.1 (2019), pp. 93–112. DOI: 10.1177/0739456X17723971.

[34] M. Dobbins. "Rapid review guidebook". In: *Natl Collab Cent Method Tools* 13 (2017), p. 25.

[35] *FindUT Research Tool*. https://ut.on.worldcat.org/discovery?lang=en. Accessed: 2024-10-07.

[36] *Google Scholar*. https://scholar.google.com. Accessed: 2024-10-07.

[37] *IEEE Xplore Digital Library*. https://ieeexplore.ieee.org. Accessed: 2024-10-07.

[38] *SpringerLink*. https://link.springer.com. Accessed: 2024-10-07.

[39] *Scopus*. https://www.scopus.com. Accessed: 2024-10-07.

[40] *ScienceDirect*. https://www.sciencedirect.com. Accessed: 2024-10-07.

[41] S. Mok. *Literature guide for Computer Science*. Accessed: 2024-10-07. June 12, 2024. URL: https://www.utwente.nl/en/service-portal/university-library/find-access-literature/guides-per-discipline/computer-science.

[42] Walden University Library. *Keyword Searching: Finding Articles on Your Topic: Select Keywords*. Accessed: 2024-10-23. URL: https://academicguides.waldenu.edu/library/keyword/search-strategy.

[43] Washington State University Libraries. *Conducting a Literature Review: Searching the Literature*. Accessed: 2024-10-23. URL: https://libguides.libraries.wsu.edu/litreview/searching.

[44] City University of Hong Kong Library. *Literature Review - Finding the Resources: Formulating your search statement*. Accessed: 2024-10-23. URL: https://libguides.library.cityu.edu.hk/c.php?g=423953&p=2897523.

[45] University of Wollongong Library. *Guides: Literature Review: How to search effectively*. Accessed: 2024-10-23. URL: https://uow.libguides.com/literaturereview/how.

[46] University of Twente Library. *FindUT Tips & Tricks | Service Portal | University of Twente*. Accessed: 2024-10-23. URL: https://www.utwente.nl/en/service-portal/university-library/find-access-literature/findut-tips-tricks.

[47] Google Scholar. *About Google Scholar*. Accessed: 2024-10-29. 2022. URL: https://scholar.google.com/intl/en/scholar/about.html.

[48] Meriam Library. *Evaluating Information – Applying the CRAAP Test*. Accessed: 2024-10-23. URL: https://library.csuchico.edu/sites/default/files/craap-test.pdf.

[49] F. Zhao, Y. Zhang, J. Lu, et al. "Measuring academic influence using heterogeneous author-citation networks". In: *Scientometrics* 118.3 (2019), pp. 1119–1140. DOI: 10.1007/s11192-019-03010-5.

[50] A. Bans-Akutey and B. Tiimub. "Triangulation in Research". In: *Academia Letters* (Sept. 2021), p. 3392. DOI: 10.20935/AL33922.

[51] T. Besseling and M. Gorlas. *Gitlab UTwente, ARS: Real-time Quishing detection using ML techniques*. Accessed: 2024-10-23. URL: https://gitlab.utwente.nl/s3533778/ars-quishing-detection.

[52] W.-C. Wu, Z.-W. Lin, and W.-T. Wong. "Application of QR-Code Steganography Using Data Embedding Technique". In: *Information Technology Convergence*. Ed. by J. J. (H. Park, L. Barolli, F. Xhafa, and H. Y. Jeong. Dordrecht: Springer Netherlands, 2013, pp. 597–605. ISBN: 978-94-007-6996-0.

[53] M. Takita, H. Okuma, and M. Morii. "A Construction of Fake QR Codes Based on Error-Correcting Codes". In: *2018 Sixth International Symposium on Computing and Networking (CANDAR)*. 2018, pp. 188–193. DOI: 10.1109/CANDAR.2018.00033.

[54] P.-Y. Lin and Y.-H. Chen. "High payload secret hiding technology for QR codes". In: *EURASIP Journal on Image and Video Processing* 2017.1 (Feb. 2017), p. 14. ISSN: 1687-5281. DOI: 10.1186/s13640-016-0155-0.

[55] Y.-W. Chow, W. Susilo, G. Yang, J. G. Phillips, I. Pranata, and A. M. Barmawi. "Exploiting the Error Correction Mechanism in QR Codes for Secret Sharing". In: *Information Security and Privacy*. Ed. by J. K. Liu and R. Steinfeld. Cham: Springer International Publishing, 2016, pp. 409–425. ISBN: 978-3-319-40253-6.

[56] A. Mishra and M. Mathuria. "A Review on QR Code". In: *International Journal of Computer Applications* 164 (Apr. 2017), pp. 17–19. DOI: 10.5120/ijca2017913739.

[57] T. Ohigashi, S. Kawaguchi, K. Kobayashi, H. Kimura, T. Suzuki, D. Okabe, T. Ishibashi, H. Yamamoto, M. Inui, R. Miyamoto, K. Furukawa, and T. Izu. "Detecting Fake QR Codes Using Information from Error-Correction". In: *Journal of Information Processing* 29 (2021), pp. 548–558. DOI: 10.2197/ipsjjip.29.548.

[58] International Organization for Standardization. *Information technology — Automatic identification and data capture techniques — QR code bar code symbology specification (ISO/IEC 18004:2024)*. 2024. URL: https://www.iso.org/standard/83389.html.

[59] S. B. Wicker and V. K. Bhargava. *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.