Chapter 1

Introduction

1.1 Background

A Quick Response code (QR code) is a two-dimensional matrix composed of black and white pixels, offering a compact and easily scannable method for storing information. QR codes have surged in popularity due to their higher data capacity and improved readability compared to traditional barcodes. As smartphone usage continues to soar, businesses are increasingly adopting QR codes as a convenient means to direct users to their websites and products. QR codes are engineered to be readable from any orientation and can still be scanned even if partially damaged or obscured, making them versatile for consumer applications [1]. Typically, QR codes are “scanned” by capturing an image of the code with a smartphone camera and then interpreted by a QR code reader app installed on the device. The reader decodes the message and performs an action based on its content, such as launching a marketplace app for downloading mobile applications if the encoded data contains a link. Often, QR codes represent hyperlinks, prompting the device's web browser to visit the specified website. Despite their convenience, the ease of creating and distributing QR codes has also attracted malicious actors looking to engage in phishing attacks, known as QRishing. This poses a security threat to the widespread adoption of QR codes, as attackers may overlay legitimate QR codes with malicious ones or create entirely fake QR code advertisements to deceive users.

1.2 Introduction to QR Code

A two-dimensional barcode with a significantly larger data capacity than its one-dimensional equivalents is the quick response (QR) code [2]. Numerous data kinds, including binary, alphanumeric, numeric, and Kanji characters, can be stored thanks to this functionality. The QR code is available in multiple versions, each with a greater data capacity. The several code versions are displayed in Fig. 1.

Fig. 1. Examples of various version of the QR codes

As seen in Fig. 2, the various sections of the QR code have distinct functions. The scanner can identify a QR code and its direction thanks to the finder patterns. Only versions 2 and later have the alignment pattern. As the version number rises, so does the number of alignment patterns. For minor image distortion, it offers compensating support. The masking pattern and error correction level information are stored in the format information [3]. The decoder uses the timing pattern to figure out how wide the modules are. The data section, error correction codes, and the remaining 8-bit codewords that separate the data and error correction codes make up the remaining region.

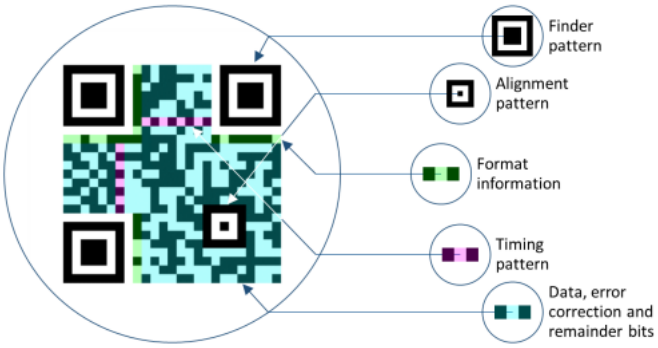


Fig. 2. Structure of the QR code (version 2)

Denso Wave created the QR code in order to control production at its manufacturing facilities. Since then, its use has expanded rapidly across several industries. It serves as a follow-on digital activity, like visiting a website or pairing an account, for marketing, as a link to extra information, for security-related tasks, like device pairing or authentication, and for connecting digital and real locations. It serves as the "physical shortcuts" for quick access to digital information and the internet.

1.2.1 Uses of QR Code

Originally developed to track automobile parts in manufacturing facilities, QR codes are increasingly being used in mobile devices and urban areas.

- Advertising: Encoding text, geolocation, and URLs or contact details to make them readily accessible to the user is the most popular use case in advertising. Most urban areas have billboard ads with QR codes that provide information to prospective clients without requiring them to physically enter the URL in order to access a webpage. Tesco, a supermarket company, increased online sales and expanded its market share in South Korea by utilizing QR codes [4]. By incorporating QR codes into hairstyles, a shampoo business devised yet another creative and economical marketing strategy. People with these haircuts served as moving billboards for shampoo since, when scanning, their "hair tattoos" led to the company's website.
- Mobile Payments: Additionally, QR codes facilitate mobile payments and offer the ability to buy goods or services by just scanning a code. This type of payment is known as "one-click." The user is taken to the company's website or an intermediary payment agency after scanning the corresponding QR code. In certain nations, PayPal, one of the largest payment companies, has already implemented this payment method.
- Access Control: When combined with other security-enhancing techniques, QR codes are utilized for physical access control. Researchers combined the One Time Password (OTP) technique with QR codes to offer a secure authentication mechanism. A mobile application that creates QR codes, a client PC with a camera to scan the code, and the main server that stores the user data are all involved. The client PC scans a QR code that the user creates with an encrypted password encoded in order to authenticate.
- Augmented Reality and Navigation: Additionally, digital government services use QR codes to efficiently disseminate important information to the general public. In addition to facilitating user navigation through park trails and museums, QR codes are employed to boost public participation. They are also incorporated into games and utilized as additional educational content [5]. Additionally, QR codes can be utilized to facilitate information sharing amongst participants in the same social event or to enhance the learning process. Additionally, intriguing and inventive applications of QR codes are showcased, such as when they serve as a surface for an augmented

reality program to be installed, producing and displaying to the user stunning 3D virtual items.

1.3 QR Codes as Attack Vectors

This section explains various QR code-based attack scenarios. Social engineering is the attack scenario that is most commonly covered in the media. Social engineering, as employed in information technology (IT) security, is the technique of tricking others into giving the social engineer private information, mostly for data theft. Phishing is one of the most often used social engineering techniques. Malicious QR codes are used by attackers to send visitors to phony websites that pose as trustworthy in order to steal private data, including credit card numbers, usernames, and passwords. To take advantage of QR codes, there are two primary attack vectors:

- The entire QR code is changed by the attacker: This attack is straightforward but successful. A new QR code with a malicious link encoded is created by an attacker and pasted over an existing one, such as a billboard advertisement [6].
- The attacker alters specific QR code modules: This modification's primary concept is that the encoded content is altered only by altering the color of particular QR Code modules that the user will be redirected to upon scanning the code.

1.4 Threat Model

There is a considerable number of malicious QR codes in the network, and attacks based on QR codes are frequent. QRLJacking and vulnerabilities in scanning apps are examples of risks that are not covered by the threat model for QR codes. The attack surface and security issues can be used to analyze the threat model.

1.4.1 Attack Surface

S_i stands for the ith attack surface in Fig. 3. In order to model the threats and security requirements, this section will examine S_1, S_2, and S_3 in turn [7].

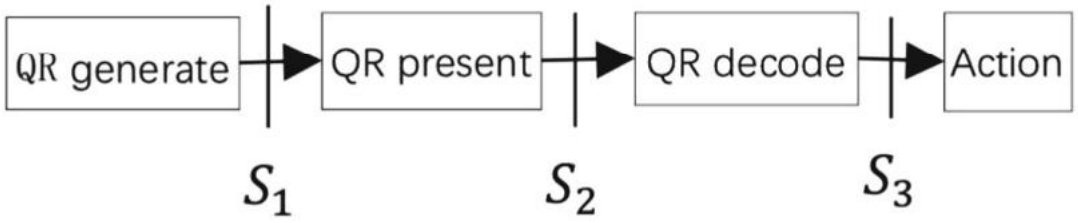


Fig. 3. QR code attack surface

S_1: Authentication measures are not provided by the QR code standard. The purported producer might not have created the QR code. In order to create illicit QR codes, attackers could pose as trustworthy manufacturers. Therefore, it would be simple to fabricate the QR codes.

S_2: By executing certain actions, such as turning some white blocks into black ones to alter the message in the QR code, attackers might alter the shown QR codes. In the meanwhile, the message is vulnerable to eavesdropping because of the nature of QR codes.

S_3: Attackers may employ malicious messages (such as malicious URLs) to carry out attack schemes like the QRishing attack due to a lack of message verification procedures [8]. By examining the attack surface of the QR code, we were able to create the security specifications that are displayed as follows:

- The QR code generator's identity authentication.
- Confidentiality of messages sent by QR codes.
- QR code integrity.
- Malicious message validation.

1.4.2 Challenges of Designing QR Code Security Mechanism

Designing a QR code security system is difficult because of the properties of QR codes and associated applications. The following are the primary characteristics:

- The nature of QR code itself: The characteristics of QR codes themselves, such as their small storage capacity, ease of message leakage, inability to be read by humans, etc.

- Message diversity: QR codes can convey a wide range of messages, and it can be challenging to distinguish harmful ones from huge, unpredictable messages, such as dangerous URLs.
- The independence of function modules: QR codes are insecure because their function modules—generation, display, decoding, and action—are implemented independently [9].
- Variety of application requirements: The security requirements may vary greatly based on the function of QR codes in applications. For instance, applications involving public message sharing do not require the anonymity of QR codes. The security systems are complicated by the many security needs.

1.5 Usable Security Design Requirements

It was discovered that the majority of the most widely used Android QR code scanners are unable to identify phishing attempts. However, to create software that aids the user in determining whether a URL is trustworthy, a more thorough examination of security, privacy, and usability considerations is required. Creating design standards is a significant task that will aid in the creation of a safe and practical multi-layer framework for processing QR codes. These design principles ought to be created in a way that strengthens the QR code and the reader software while also assisting the user in identifying possible dangers. In order to facilitate research in the fields of security and human-computer interaction concerning attack scenarios, a set of prerequisites is required [10]. The following three types' needs can be separated for this purpose: (1) Secure QR Code Requirements, (2) Service Layer Requirements and (3) Usability Requirements

1.5.1 QR Code Requirements

The security criteria for the QR coding technique are listed in this section. Improvements to the coding system are regarded as independent of the application of the QR code reader.

- Visual QR Codes: Visual QR codes greatly aid the user in identifying altered or substituted QR codes in metropolitan areas in the event of an assault. The more intricate the topic, the more difficult it is for an attacker to covertly alter QR codes. We recommend examining the effect of intricate color schemes incorporated into the overall ad's color scheme on the user's capacity to recognize malevolent alterations in

order to make it more costly for an attacker to replace the original QR code (for example, in billboard advertising).

- Digital Signatures: Digital signatures have shown themselves to be a successful way to increase security in several fields. In order to confirm the code's creator and, thus, determine whether the code has been altered, we advise emphasizing the incorporation of digital signatures into the standardization of QR codes [11]. Attackers must alter the checksum and the verification procedure in order to use a digital signature, which makes QR code-based attacks much more difficult. However, the area needed to encode actual data is reduced due to the higher amount of data to be encoded. Additionally, like SSL, QR code scanners must be modified to validate digital signatures and show whether the verification was successful. To suggest a change to the specification, the integration of digital signatures must be developed.

1.5.2 Service Layer Requirements

In order to strengthen secure QR codes, the problems outlined in this part focus on protecting the QR code reader application. Enhancing the security mechanisms included into QR codes and figuring out whether a user's action is required to avoid a harmful code are the main goals of service layer enhancements.

- Masking: There is a pattern to how the black and white modules are distributed in a QR code that complies with specifications. The mask that specifies whether or not to alter the color of the module under consideration determines this pattern. A certain level of faulty pixels has no detrimental effect on decoding the QR code because of the robustness offered by the error-correcting Reed-Solomon codes [12]. The likelihood that the QR code has been altered increases with the degree of departure from an even distribution of black and white modules. To employ masking features to protect reader software, a thorough examination of the trade-off between error rate and security would be helpful.
- Malicious URL Detection: Generally speaking, there are various methods for effectively separating potentially harmful URLs from benign ones. However, an attacker may utilize shortened URLs to hide harmful URLs. Metrics can be used to assess a URL's credibility. Additionally, the originator of encoded URLs can be confirmed using URL blacklists or whitelists.

1.5.3 Usability Requirements

This section describes research challenges with respect to the user's decision making process on the trustworthiness of a QR code.

- Content Display: Because QR codes cannot be read by humans, content display is necessary to let the user know what is actually encoded.
- Content Preprocessing: Simply displaying the encoded content in the case of shortened URLs or redirects is insufficient to tell the user of whether the content is benign or harmful. As a result, acceptable content preparation technologies are required. To show the user the final URL, for example, shortened URLs could be run in the background.
- Anti-Phishing Tools: Phishing is one of the main issues with modified QR codes. In terms of usability, it is critical that the user can easily understand the verification procedure. The primary difficulty, though, is in appropriately alerting the user to an occurrence, much like with SSL.
- Content Verification: Verification techniques, including blacklists, should be prioritized in addition to content preprocessing prior to presentation [13]. In many situations, warnings are ineffective at alerting users to potential dangers and the consequences of their choices. These results underline the necessity for more investigation into practical methods for distinguishing between confirmed and unverified content.

1.6 The QR Code Phishing Attacks

It is hardly surprising that phishers exploited QR codes' vulnerability to initiate their assaults, given how quickly everyone and every sector adopted them. Phishing is an attack that uses technical trickery or social engineering to try to "fish" the victim for private information, such as bank account details or login credentials. Like any new technology, there will be risks and vulnerabilities that could compromise users' security and privacy. Fig. 4 provides a summary of the QR code assaults.

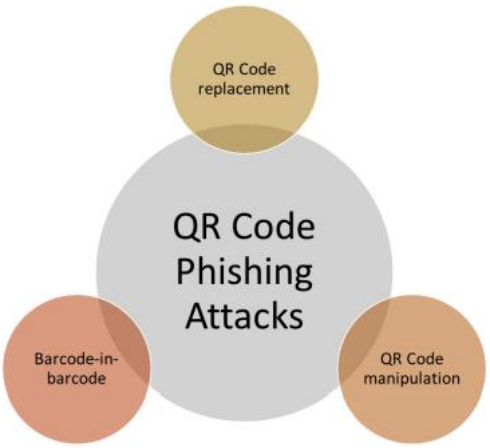


Fig. 4. QR code phishing attacks

Three types of QR Code Phishing Attacks can be distinguished:

i. QR Code Replacement: It is possible to turn a valid QR code that is displayed into a harmful one by either adding black modules to the white ones or by altering the entire code. The mask, character encoding mode, character count indicator, mixing modes, data component, and/or error correction section of the current QR code can all be used to manipulate it. Such manipulation can be carried out by using a pen, for instance, to alter the white modules on the codes to black, by comprehending the structure of the QR code as shown in Fig. 2. Phishing attacks may result from this. According to a survey, most fraudulent QR codes that are identified lead users to phishing and exploit websites rather than downloading malware directly.

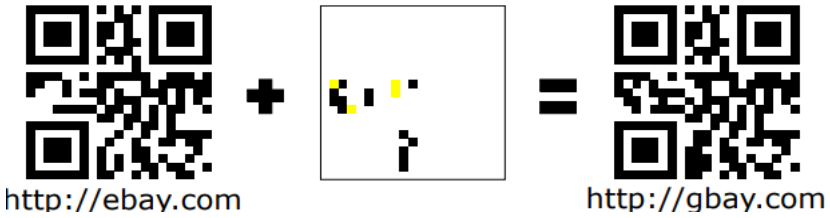


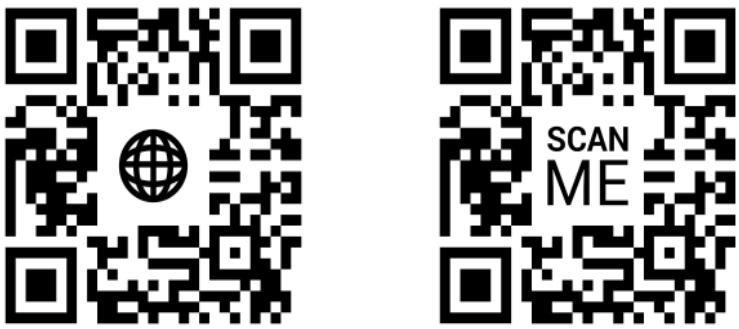
Fig. 5. Example of a manipulated QR code

Fig. 5 illustrates how to modify both the white and black modules to make the link that initially leads to "ebay.com" point to "gbay.com." This type of assault is comparable to typo squatting, in which the attackers register website addresses that differ from the correct ones

[14]. This phony version of the desired website could be accessed by the user who typed the address incorrectly.

ii. QR Code Manipulation: The risk posed by malicious QR codes is heightened by applications that automatically launch the embedded link without allowing users to verify it beforehand. Additionally, the use of URL shorteners complicates visual verification for users. Furthermore, mobile operating systems often hide the full URL of the loaded website for ease of use on small screens, making it challenging for users to assess the nature of the link. This limitation, coupled with the inability of most QR code scanners to detect modifications, exacerbates the risks associated with QR code usage [4]. Although many mobile applications display the decoded URI, they may not thoroughly check URIs and web content for malicious intent.

iii. Barcode-in-barcode attack: The barcode-in-barcode exploit is another documented method of manipulating QR codes. Because QR codes allow for data correction with a percentage of data inaccuracy, parts of unknown coding can be ignored by the decoder. As seen in Figs. 6a and 6b, this space can be utilized to display text or a symbol. By utilizing this capability, it is possible to incorporate a different barcode inside the QR code while maintaining the ability to decode the original code. Aztec, Data Matrix, or even another QR code are additional barcode types that can be incorporated.



(a) A QR code with an icon. (b) A QR code with text.

Fig. 6. Example of QR codes.

Fig. 7 shows an example of a barcode-in-barcode.

Fig. 7. A QR code with embedded Aztec barcode (in shaded red).

When the scanner is moved to scan the QR code, the embedded barcode will show up in the scanner frame before the outer QR code [15]. As a result, rather than the outer QR code, the embedded barcode will be decoded. Additional research reveals that certain programs are able to scan both the inner (embedded) and exterior (original) barcodes. By using this technique, attackers can take advantage of a flaw in certain scanners that allows them to read embedded barcodes.

1.6.1 Phishing Components

The phishing medium, the assault vector, and the technological methods employed during the attack are the three parts of phishing techniques. Figure 8 illustrates the connection between the technological methods, vectors, and phishing mediums. The primary way that phishing assaults are communicated to victims is through the medium. The three most widely utilized bases are voice, internet, and short messaging services (SMS). The internet's accessibility has made it much easier for phishers to contact their victims.

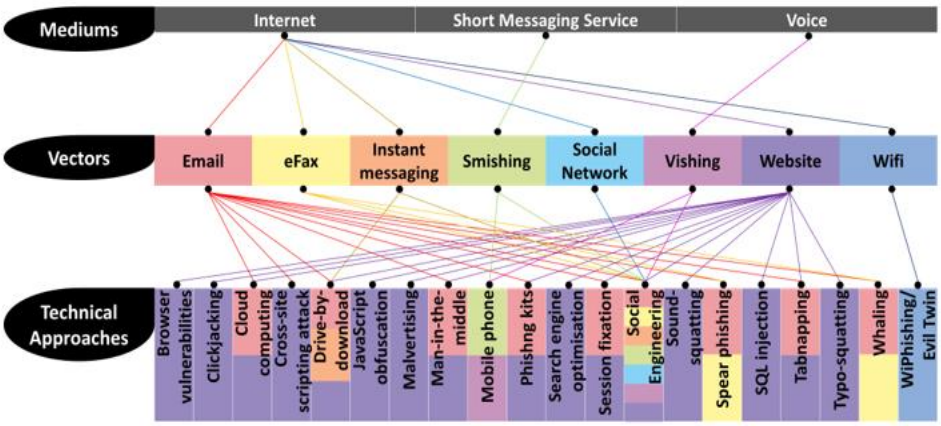


Fig 8. The relationship among the phishing strategies' technological approach, medium, and vector. For ease of identification, the background colour of the technical approaches is set to the colours of the related vectors.

The vehicle used to launch phishing attacks is the channel vector that the phishers use depending on the media [16]. Phishers can access these vectors through the internet, including social networks, websites, instant messaging, eFax, and email. Phishing by voice is known as vishing, and phishing via short message service (SMS) is known as smishing. In order to increase the efficacy of phishing, technical measures are employed in addition to social engineering phishing. Drive by-download, man-in-the-middle (MITM), cross-site scripting (XSS) attack, tab napping, spear phishing, whaling, search engine optimization (SEO), session fixation, malvertising, social engineering, JavaScript obfuscation, browser vulnerabilities, mobile devices, cloud computing, and WiPhishing or Evil Twins are some of the various methods that phishers currently employ. Additionally, there are technical methods like SQL injection, typo-squatting, and sound-squatting that avoid social engineering or the necessity to trick the victim. Additionally, phishing kits are classified as technological techniques; nevertheless, they are a tool to help implement phishing attacks rather than a stand-alone phishing attack.

1.7 Countermeasures

Given the potential and seriousness of QR code exploitation for phishing, it is essential to implement safeguards to safeguard QR code users. Fig. 9 summarizes the four security concerns that researchers found with QR codes. The focus will be on countermeasures in the context of QR codes being used in phishing and pharming attacks, which center on the issue of the codes' integrity because the attacks include code manipulation.

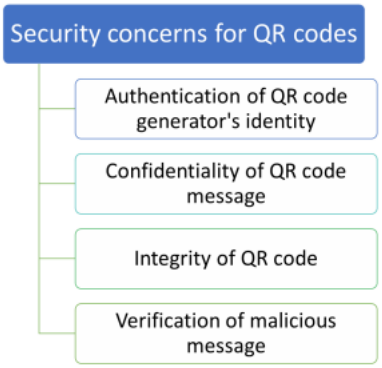


Fig. 9. Security concern for QR codes

A variety of countermeasures, from software-based to user-centered, were suggested. Increasing user awareness through education about the potential of QR codes to be used in phishing or pharming attacks is one of the user-centered countermeasures. Additionally, users can be informed and reminded to never divulge important information via a QR code link and to always verify the URL that the code points to. For improved phishing protection, software-based countermeasures must be used in conjunction with user-centered countermeasures [17]. Scanners for QR codes that can display the decoded URL so the user may check it before clicking on it. The ZXing Barcode Scanner is one example of such a scanner. Adding automated inspection services like PhishTank API and Google Safe browsing can improve QR code scanners even more. A harmful message detecting party with whitelist and blacklist integration is an additional feature that checks and blocks malicious URLs before the user opens them. Some defences against QR code phishing scams are listed below.

- Stringent Data Correction Percentage: Enforcing a stringent data correction percentage is crucial in QR code phishing detection to minimize the chances of tampered or manipulated codes being successfully scanned. QR codes use error correction techniques, such as the Reed-Solomon algorithm, to recover data even when part of the code is damaged. However, attackers often exploit relaxed correction thresholds to embed malicious content. By setting a higher standard for error correction, the system becomes more robust against such manipulation, ensuring that only legitimate and unaltered codes are processed.
- Prioritize the Type of Barcode to Check: Modern barcode scanners are designed to read multiple barcode types, such as QR codes, Data Matrix codes, and linear barcodes. For phishing detection purposes, prioritizing QR codes is essential, as they are the most commonly used format for embedding URLs [18]. By narrowing the focus to QR codes, the scanner reduces the likelihood of processing irrelevant or less secure barcode formats, streamlining the detection process and enhancing overall security against phishing attacks.
- Display the Border of the Code Being Detected: When multiple QR codes or barcodes are present in the scanner's view, it can be difficult to determine which one is being read. Displaying the border of the detected code provides visual feedback to users, allowing them to verify that the correct code is being scanned. This is particularly

useful in environments where malicious codes might be placed close to legitimate ones, such as on posters or product packaging. By ensuring clarity, users can avoid unintentional scans of harmful codes.

- Notify the User if Multiple Barcodes Are Detected: In scenarios where multiple barcodes are detected, it is essential to notify the user and provide them with the ability to select the correct code. This is important because attackers may deliberately place phishing QR codes near legitimate ones to deceive users. By prompting users to make an informed choice, the system reduces the chances of accidental scans of malicious codes and enhances security against phishing attempts [19].
- Show the Decoded URL Before Launching It: Phishing attacks often rely on users blindly following URLs embedded in QR codes. By displaying the decoded URL before launching it, users are given the opportunity to review the destination. This step allows users to identify potentially suspicious or malicious links by checking for irregularities, such as unfamiliar domains or mismatched URL structures. It adds a layer of transparency and empowers users to make safer decisions.
- Use a URL Check Algorithm: To further enhance phishing detection, implementing a URL check algorithm is essential. This algorithm can validate URLs by analyzing patterns, cross-referencing known phishing domains, and checking for other indicators of malicious intent, such as obfuscated links or misspelled domain names. By incorporating this automated validation process, the scanner can provide real-time alerts about potentially harmful links, ensuring users are better protected from phishing attacks [20].

Chapter 2

Literature review

S. Ismail, et.al (2021) suggested a novel framework to authenticate QR Code and detect phishing [21]. This framework was effective for detecting the phishing and malicious URLs while validating the QR Code. It was employed for detecting abnormal and malicious links, embedded in the QR codes and utilized as a vector attack. A testing was done prior to suggest this framework. To ensure that the code was scanned perfectly, the suggested framework aimed to detect the malicious and phishing page. Moreover, it was proved useful for prevent users from harmful QR Codes.

A. S. Rafsanjani, et.al (2023) presented a secure and privacy-friendly QR code scanner (QsecR) relied on a model of detecting malicious URL [22]. This model was designed on the basis of classifying predefined static features. The URLhaus and PhishTank were considered to gather a dataset of 4000 real-time random URLs. Diverse QR code scanners were used to compute its security and privacy. The experiments exhibited the supremacy of presented

model over others and its accuracy was counted 93.50% and precision was 93.80% to detect phishing URL in QR.

B. Herlina, et.al (2023) projected a model to detect phishing URLs in QR codes based on Machine Learning (ML) methods [23]. Such URLs were similar to real websites, which the cybercriminals had created for attaining user information. This work focused on gathering a dataset from several sources, having features of address bar, domain and HTML. Six ML methods: DT, RF, SVM, MLP, AENN and XGBoost were adopted. The simulation indicated that the last method was performed well and offered an accuracy of 92% to detect phishing URLs in QR codes.

M. Sahay, et.al (2024) aimed at deploying machine learning (ML) techniques for detecting SQL injections in applications [24]. The malicious actions were recognized in a cloud-based SaaS scenario. An analysis was conducted to verify whether the QR codes were practical and secure in the anti-phishing recommendations for the suggested method (secure QR code). These recommendations were deployed after exploring the extensive cases and attack vectors. The social engineering called phishing was found a frequent scam in QR codes as an attack vector.

G. A. Amoah, et.al (2022) aimed to discover the privacy issues related to QR codes [25]. The introduced method employed a count vectorizer to extract words of every URL, and considered the URLs comprised in QR code for extracting features. The genuine URLs were differentiated from the phishing ones after tokenizing the traits and words. The Naive Bayesian machine learning (NB-ML) methods were employed in a recursive loop with logistic regression (LR). This resulted in generated an effective mechanism to mitigate the quishing (QR-phishing) behavior.

M. Thakare, et.al (2023) suggested to deploy machine learning (ML) and convolutional neural networks (CNNs) into the QR code scanning procedure for detecting and blocking malicious codes [26]. The secure QR codes were assisting in making the online transactions more secure and preventing web phishing attacks. The developed method was adopted to detect malware URLs with Support Vector Machine (SVM) method. This method was designed to make the communication more scalable, flexible, and secure.

G. R. Charan, et al. (2023) suggested machine learning-based detection of phishing links and UPI transaction QR codes explores the complexities of this crucial subject, offering insights into fraud and facilitating the exploitation of UPI ecosystems [27]. In order to avoid fraud and enhance security measures, this study aims to address this issue by utilizing machine learning and artificial intelligence techniques. Phishing is the deliberate attempt to impersonate a trustworthy source in order to trick people into divulging personal information, such as usernames, passwords, or bank account details. QR codes, a type of data storage that can be quickly accessed by a smartphone's camera, can also be used in phishing attempts. User knowledge and defense against phishing scams.

B. B. Gupta, et al. (2024) presented a phishing email detection framework created especially for enterprise information systems that combines CNN for classification and Bidirectional Encoder Representations from Transformers (BERT) for feature extraction [28]. Key features are extracted from email content using BERT's linguistic skills, and a convolutional neural network (CNN) model tuned for phishing detection processes the features. With a 97.5% accuracy rate, our suggested model shows excellent competence in spotting phishing emails. By successfully handling the growing complexity of phishing assaults, this method sets a new standard for email security and represents a significant leap in the use of deep learning to cybersecurity.

Abdulla Al-Subaiey, et al. (2024) proposed a high-performance machine learning model for email classification, addressing shortcomings in previous research, including dependence on proprietary datasets and lack of practical application [29]. Using the largest and most complete public dataset, the model attains a f1 score of 0.99 and is intended for implementation in pertinent applications. Explainable AI (XAI) is also incorporated to improve user confidence. This study provides a useful and extremely precise solution, aiding in the battle against phishing by giving consumers access to a real-time web-based tool for detecting phishing emails.

Sakib Shahriar Shafin, et al. (2024) suggested a novel approach that makes use of eXplainable AI (XAI) to improve FS in machine learning models for phishing website detection [30]. In particular, we use aggregated local interpretable model-agnostic explanations (LIME) to identify particular localized patterns and SHapley Additive exPlanations (SHAP) for global perspective. By identifying the most useful traits, the suggested SHAP and LIME-aggregated feature selection (SLA-FS) framework makes

phishing detection more accurate, quick, and flexible. We test the efficacy of this method by comparing the performance of three ML models before and after FS on a current web phishing dataset. Our results show that KNN lags whereas random forest (RF) and XGBoost (XGB) greatly benefit from the SLA-FS framework, with RF having an accuracy of 97.41% and XGB having an accuracy of 97.21%.

Brij Bhooshan Gupta, et al. (2024) presented a framework that combines the hyperparameter optimization capabilities of the Whale Optimization Algorithm (WOA) with the sequential data processing advantages of a Recurrent Neural Network (RNN) [31]. Our model leverages a large Kaggle dataset with more than 11,000 URLs, each with 30 properties. A thorough validation procedure shows that the WOA's hyperparameter modification improves the RNN's performance. With an overall accuracy of 92%, the results, as measured by precision, recall, and F1-score metrics, outperform baseline models. This work highlights the WOA's efficacy in honing machine learning models for the crucial task of phishing detection, in addition to showcasing the RNN's ability to learn intricate patterns.

A. B. Majgave, et al. (2024) suggested a veritable pre-trained deep transformer network model for phishing behavior detection called the transformer-based Deep Belief Network (TB-DBN) [32]. The suggested hybrid model was used to create a cross-validation method with grid search hyper-parameter tuning based on the Intelligence (IBBA). Using a probabilistic estimate guided boosting classifier model, predictions were produced to categorize the phishing URLs and assess their performance in terms of F1-score, accuracy, precision, recall, and specificity. The reputation of the source, the findings of the content analysis, and any unusual behavior will all be taken into consideration when determining the risk rating of the URL. The results demonstrate that adjusting variables improves the performance of deep learning systems based on Python. The results of the suggested approach are excellent, with a precision of 99.2% and an accuracy of 99.4%.

D. Sturman, et al. (2024) examined how phishing knowledge, cue use, and decision-making styles affect the detection of phishing emails. An online email sorting task and assessments of phishing knowledge, email decision styles, cue use, and email security awareness were performed by participants (N = 145) [33]. The ability to distinguish between phishing and legitimate emails was only predicted by cue use. Greater phishing detection and a tendency to label all emails as phishing were linked to phishing knowledge. Lower detection of phishing emails was predicted by a preference for intuitive decision-making, which was motivated by

a stronger propensity to identify emails as authentic. These results lend credence to the idea that expert performance is made possible by the unique cognitive process of cue use.

Jawhara Aljabri, et al. (2024) presented a deep learning-based Dwarf Mongoose Optimization for Phishing Attack Detection (DMODL-PAD) technique for an Internet of Things platform. The DMODL-PAD method's goal is to detect phishing attacks on the IoT platform automatically by using feature selection (FS) with a hyperparameter tuning technique [34]. Unlike the FS approach, the DMODL-PAD method uses the DMO model. The phishing attack detection procedure can then be carried out by a hybrid stacked autoencoder model. The hyperparameter tuning procedure in the DMODL-PAD approach makes use of a (JSO). The DMODL-PAD technique's experimental validation has been investigated using a benchmark database. The comprehensive results demonstrated the DMODL-PAD technique's improved detection performance over other current approaches. These outcomes demonstrated the effectiveness of the DMODL-PAD approach in identifying IoT phishing assaults.

Dennik Baltuttis, et al. (2024) investigated how visual risk indicator can assist staff in identifying phishing attempts, acknowledging the shortcomings of solely technology- or human-centered approaches [35]. In order to do this, we used an eye-tracking lab experiment where participants assessed the reliability of emails with different levels of trustworthiness. Our analysis, which focuses on how humans process information when spotting phishing attempts, shows that having a visual risk indicator available can have a big impact on trust and response behavior without negating implicit phishing cues (like obvious senders or anonymous recipients). In order to obtain the desired directing effects, our findings indicate that businesses should calibrate visual risk indicators appropriately. However, the calibration is still a trade-off that is dependent on the context of the company. We talk about the implications for phishing attempt mitigation using integrative cybersecurity techniques.

N. Kamble, et al. (2024) Used the Fractional Dingo Hunter Prey Optimization-SqueezeNet (FDHPO-SqueezeNet) approach, the phishing websites are successfully identified. The various aspects are extracted from the website data independently, including ocular features, web features, and Natural Language Processing (NLP) features [36]. SqueezeNet is then utilized to detect phishing websites after the best features have been chosen, fused, and enhanced. SqueezeNet is then used to detect phishing attempts, and its detection effectiveness is improved by training it with the FDHPO approach. The findings of the experiment showed

that the created FDHPO-SqueezeNet technique performed better than other popular phishing detection methods, with a maximum of 94.05% accuracy, 94.26% sensitivity, and 93.75% specificity, respectively.

F. Rashid, et al. (2024) demonstrated that these techniques are not generalizable, meaning they only function well when test sets are separated from the same training dataset [37]. Given that most phishing attempts are brief and make use of recently created domain names, this is a serious problem. Additionally, URL data gathered at different firms may change because different network vantage points and middleboxes record URLs in slightly different formats. In order to improve the model's transferability across datasets, we provide a methodology based on Unsupervised Domain Adaptation. We test our method on three datasets and demonstrate that the average increase in cross-dataset F1 score performance is 0.06, and in some circumstances, it can reach 0.2.

R. J. van Geest, et al. (2024) provided the groundwork for workable and reliable phishing detection architectures by introducing a revolutionary framework that is specifically made to be applicable in the actual world [38]. To assess its efficacy, resilience, and detection speed, we create a proof of concept. We also present a novel approach for modeling bypass attacks on single-analysis base models. Our tests show that the suggested hybrid framework performs better than individual models, exhibiting increased efficacy, resilience to attempts at circumvention, and real-time detection capabilities. Our proof of concept outperforms the existing state-of-the-art method while consuming less computing time, achieving an accuracy of 97.44%. The findings provide light on the complex aspects of hybrid models that go beyond their efficacy and highlight how crucial holistic applicability is in hybrid approaches to meet the urgent need for

E. Zhu, et al. (2024) suggests a new phishing detection model called phishing detection based on hybrid features (PDHF), which combines the best aspects of automatic and artificial deep learning [39]. Using an enhanced bidirectional search algorithm and the recently created feature importance assessment index, redundant features are eliminated to produce the best fake phishing features. A one-dimensional character convolutional neural network (CNN) and a chaotic quantized attention mechanism are used to learn deep characteristics from URLs in order to increase the effective time of phishing detection. According to the experimental data, PDHF obtains an accuracy of 0.9965, precision of 0.9942, recall of 0.9940, and □1-score of 0.9941, outperforming numerous state-of-the-art techniques. suggests a new phishing

detection model called phishing detection based on hybrid features (PDHF), which combines the best aspects of automatic and artificial deep learning [39]. Using an enhanced bidirectional search algorithm and the recently created feature importance assessment index, redundant features are eliminated to produce the best fake phishing features. A one-dimensional character convolutional neural network (CNN) and a chaotic quantized attention mechanism are used to learn deep characteristics from URLs in order to increase the effective time of phishing detection. According to the experimental data, PDHF obtains an accuracy of 0.9965, precision of 0.9942, recall of 0.9940, and □1-score of 0.9941, outperforming numerous state-of-the-art techniques.

A. Prasad, et al. (2023) introduced PhiUSIIL, a framework for detecting phishing URLs that uses incremental learning and similarity indexes [40]. Bit squatting, combosquatting, homograph, punycode, homophone, zero-width characters, and other visual similarity-based assaults can all be successfully identified with the use of the similarity index. The framework can continuously add new data to its knowledge base thanks to the incremental learning approach. Furthermore, different security requirements of people or organizations can be accommodated by creating different security profiles. In order to create a phishing URL dataset, called the PhiUSIIL phishing URL dataset, which includes 134850 authentic and 100945 phishing URLs, PhiUSIIL extracts URL features, downloads the webpage from URL to extract HTML features, and generates additional features from existing information. The trial findings demonstrate the framework's efficacy and guarantee that it continues to be current and effective against new and advanced phishing techniques.

L. Wang, et al. (2023) suggest PDTGA, a technique that enhances the efficacy of phishing scam detection in Ethereum by utilizing graph representation learning based on temporal graph attention [41]. In particular, the time encoding function interacts with node attributes, edge features, and the graph topology to model the time signal, and we directly learn the functional representation of time. Three datasets of varying sizes were created using a real-world Ethereum phishing scam dataset that included over 250,000 transaction records between over 100,000 account addresses. We first compiled the recurring pattern of Ethereum phishing scam activity using data analysis. Next, we created three types of transaction edge features and fourteen types of account node features. Experimental evaluations based on the above three datasets demonstrate that PDTGA with 94.78% AUC score and 88.76% recall score outperforms the state-of-the-art methods.

C. C. L. Tan, et al. (2023) suggested a novel hybrid identity-based phishing detection method that makes use of the textual and visual identity of websites [42]. Our approach improves the accuracy of logo detection by using new image elements that resemble human vision, building on previous anti-phishing work that used the website logo as visual identity. The suggested hybrid approach combines a textual identity—specifically, brand-specific keywords extracted from the webpage content through textual analysis techniques—with a visual identity. With an overall accuracy of 98.6%, we empirically showed on several benchmark datasets that this joint visual-textual identity detection approach considerably enhances phishing detection performance. Comparable true positive rates and a 3.4% decrease in false positive rates were seen when benchmarking findings against an established approach, supporting our goal of lowering the misclassification of valid

M. Abdullah Alohali, et al. (2023) presents a new metaheuristics deep learning-oriented phishing detection method for a safe and sustainable environment (MDLPD-SSE) [43]. The main goal of the MDLPD-SSE model that is being given is to detect phishing websites. To do this, the input URL is pre-processed using the MDLPD-SSE technique to convert it into a format that is compatible. Furthermore, feature subsets were derived using an enhanced simulated annealing-based feature selection (ISA-FS) method. Additionally, this study uses the long short-term memory (LSTM) model to detect phishing. Lastly, the hyperparameters pertinent to the LSTM model were adjusted by utilizing the bald eagle search (BES) optimization process. Our results showed that the suggested model was superior, with an increased accuracy of 95.78%.

D. Sturman, et al. (2023) created to investigate the functions of time constraint, phishing characteristics, and cue utilization in identifying phishing emails [44]. Participants in two trials had to filter emails that included both phishing and legitimate communications. Either a high or low time pressure condition was assigned to the participants. Response bias and detection sensitivity were used to evaluate performance. Participants performed a phishing knowledge test and were categorized as having either higher or lower cue use. Higher cue use helped participants distinguish between phishing and legitimate emails when they were blinded to the study's purpose (N = 191). In contrast to those who used cues less frequently, they also noted a greater bias in favor of labeling emails as phishing.

K. Omari, et al. (2023) presented a novel strategy that uses Gradient Boosting Classifiers (GBCs) to solve this problem [45]. Phishing websites are deliberately designed to mimic the

look of trustworthy websites in order to trick users into disclosing personal information. As such, these malicious websites have subtle characteristics that set them apart from their real-world counterparts. Conventional rule-based approaches frequently struggle to adequately capture these nuanced differences. Gradient Boosting, on the other hand, provides an ensemble learning framework that can leverage the combined power of weak classifiers to produce a strong model that is adept at recognizing these elusive characteristics. Our experimental results unequivocally demonstrate the higher performance of our suggested method on a number of measures, including as F1-score, accuracy, precision, and recall.

M. M. Alani, et al. (2022) introduced PhishNot, a machine learning-based phishing URL detection tool [46]. As a result, our work predominantly employs a "learning from data" driven strategy that has been verified using a relevant dataset and scenario. To ensure the system's practical usability, the number of input features was lowered to 14. With a very high accuracy of 97.5%, Random Forest demonstrated the greatest performance in the experiments. Additionally, the combination of our system's high phishing detection rate and high speed (an average of 11.5 seconds per URL) when placed on the cloud makes it more practical.

D.-J. Liu, et al. (2022) used a variety of deep learning techniques to propose three semantic-based phishing detection models at varying depths while taking into account the semantic information of different scales [47]. Multi-scale Data-layer Fusion (MDF), Multi-scale Feature-layer Fusion (MFF), and Multi-scale In-depth Fusion (MIF) are the three models that have been suggested. With an F1-Measure of 0.9830, an AUC value of 0.9993, and a false positive rate of 0.0047, the MIF model performs the best on a complex dataset, according to experimental results on a created complex dataset. All three models have good recognition capabilities. Following a 6-month active discovery experiment in which 3016 phishing websites were identified in a real network environment, and additional comparison with both visual and text methods, it is discovered that the suggested model is

S. M. Alshahrani, et al. (2022) presented a novel detection model that uses data mining with the Particle Swarm Optimization technique (PSO) to increase and empower the method of detecting phishing URLs [48]. To find the phishing prospects from the URL, feature selection is done using a variety of methods. This method uses data mining rules to extract the features that were mined from the URL. The URL format is used to choose the features. PSO approaches are used to classify these features that were found by the data mining rules.

Phishing URLs can be identified thanks to feature selection with PSO optimization. This method maximizes the true positive rate for phishing URL identification by using a high number of rule IDs. Based on the structure of the URL itself, the studies demonstrate that feature selection utilizing data mining and particle swarm optimization greatly aids in identifying phishing URLs.

Hadil Shaiba, et al. (2022) introduced a model for Hunger Search Optimization with Hybrid Deep Learning-enabled Phishing Detection and Classification (HSOHDL-PDC) [49]. The HSOHDL-PDC model that is being described focuses on efficiently identifying and categorizing phishing attempts using website URLs. Furthermore, because URLs typically contain words that are of little significance, the SOHDL-PDC model employs character-level embedding rather than word-level embedding. Furthermore, a hybrid Convolutional Neural Network-Long Short Term Memory (HCNN-LSTM) method is used to detect and categorize phishing attempts. With the aid of the HSO method, the hyperparameters used in the HCNN-LSTM model are tuned, leading to better results. Several datasets were used to validate the performance of the suggested HSOHDL-PDC model, and the results demonstrated the model's superiority over other current methods.

T. O. Ojewumi, et al. (2022) used three machine learning models that were trained on a dataset of fourteen (14) features to analyze and apply a rule-based method for phishing detection [50]. The machine learning techniques include Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbor (KNN). The Random Forest model turned out to perform the best out of the three algorithms that were employed. Rules from the Random Forest Model were taken out and included into the PhishNet browser extension for Google Chrome. Throughout this study, PhishNet is constructed with web technologies like HTML, CSS, and Javascript. Consequently, PhishNet makes it possible for extremely effective phishing detection on the internet.

S. Minocha, et al. (2022) suggested a novel phishing detection method that combines the k-nearest neighbor classifier and Binary Modified Equilibrium Optimizer (BMEO) with a suggested AV-shape transfer function (AV-BMEO). The classifier's hyperparameter tuning and feature selection are done using AV-BMEO, which has strong exploration and exploitation capabilities [51]. To improve the exploration capabilities of the suggested system, the AV-shape transfer function is created using opposition-based learning. On eighteen datasets, the statistical validation demonstrates that AV-BMEO performs better than

seventeen algorithms in terms of accuracy and the number of selected features. In comparison to various state-of-the-art methods on phishing datasets, further validation demonstrates an enhanced prediction of phishing websites with the ideal number of attributes.

F. Zheng, et al. (2022) suggested a deep convolutional network that integrates word-level and character-level representation information, called the Highway Deep Pyramid Convolution Neural Network (HDP-CNN) [52]. The URL string sequences are first fed into HDP-CNN, which then embeds them at the character and word levels, respectively. The character-level and word-level embedding representations of the URL are then connected via the Highway network, and local characteristics of various sizes are extracted from the region embedding layer. With an accuracy of 98.30%, a true positive rate (TPR) of 99.18%, and a true negative rate (TNR) of 94.34%, the experimental findings show that our approach works better than alternative approaches.

A. A. Orunsolu,, et al (2019) presented an improved predictive model based on machine learning to increase the effectiveness of anti-phishing campaigns [53]. For the purpose of creating an efficient feature vector, the predictive model's Feature Selection Module is utilized. Utilizing the incremental component-based system, these features are retrieved from the URL, webpage characteristics, and webpage behavior in order to provide the prediction model with the resulting feature vector. Naïve Bayes and Support Vector Machines, which were trained on a 15-dimensional feature set, are used in the suggested method. The trials were conducted using datasets that included 2500 benign and 2541 phishing incidents. For both SVM and NB predictive models, the experimental findings show an impressive performance with 0.04% False Positive and 99.96% accuracy using 10-fold cross-validation.

R. Wazirali, et al. (2021) suggested a method for effectively detecting phishing attempts on URLs. Our solution relies on the Conventional Neural Network (CNN) algorithm, clustering and feature method, and Software Defined Network (SDN) technology [54]. The Support Vector Machine (SVM) algorithm and Recursive Feature Elimination (RFE) form the basis of the feature selection technique. The URL phishing detection process is moved from the user's hardware to the controller layer via the SDN, which then continuously trains on fresh data before sending the results to the SDN-Switches. CNN and RFE-SVM are utilized to improve phishing detection accuracy. As a result, neither obtaining the target website's content nor utilizing any third-party services are necessary for the proposal model. It swiftly classifies the actual URL by using the sequential pattern characteristics after capturing the information and

sequential patterns of URL strings without the need for prior knowledge about phishing. According to the experimental findings, our suggestion demonstrated the accuracy and resilience of differentiating between phishing and trustworthy websites. Our recommendation detects phishing attempts with an accuracy of 99.5%.

Y. Al-Hamar, et al. (2021) offered a method for identifying unique Spear-phishing attempts by comparing them to the necessary similarities in the targeted domain [55]. The plan is to use multiple new grading methods to determine whether the domain is authentic or a fake. Thus, by offering a novel enterprise solution, this study tackles targeted attacks on certain organizations. The domain names that are the subject of this detection method are often registered domain names that the victims trust. The investigation's findings demonstrate that this detection system has demonstrated a considerable reduction in email phishing attacks.

X. Xiao, et al. et. (2021) suggested the self-attention CNN, a novel Convolutional Neural Network (CNN) incorporating self-attention, for identifying phishing Uniform Resource Locators (URLs) [56]. In particular, self-attention CNN balances the datasets of valid and phishing URLs by first using Generative Adversarial Networks (GAN) to generate phishing URLs. It then builds our new classifier, which consists of four blocks: the input block, the attention block, the feature block, and the output block, using CNN and multi-head self-attention. Lastly, a high-accuracy response for an unknown website URL can be provided by the trained classifier. Detailed tests show that self-attention CNN has an accuracy of 95.6%, which is 1.4%, 4.6%, and 2.1% better than CNN-LSTM, single CNN, and single LSTM, respectively.

P. A. Barraclough, et al. (2021) presented an innovative methodology that uses ML algorithms with extensive features to enable more accurate phishing attack detection by integrating heuristic, online content, and blacklist-based approaches [57]. A thorough assessment was conducted using evaluation methodologies (metrics) to gauge the effectiveness of the suggested approach, based on the Adaptive Neuro-Fuzzy Inference System (ANFIS), Naïve Bayes (NB), PART, J48, and JRip with features. The accuracy of all the classifiers was between 99% and 99.33%. PART achieved the top performance with a speed of 0.006 seconds (secs) and an accuracy of 99.33%. We show through experiments that the suggested approach is capable of identifying phishing websites in real-time with high accuracy and can adapt well to novel phishing attempts. When compared to analogous approaches in the field, the suggested approach performs the best.

E. Zhu, et al. (2020) suggested DTOF-ANN (Decision Tree and optimum Features based Artificial Neural Network), a neural-network phishing detection model based on optimum feature selection and decision trees, as a solution to this issue [58]. In order to eliminate duplicate points from the public datasets, the conventional K-medoids clustering technique is first enhanced with an incremental selection of initial centers. The undesirable and pointless features are then eliminated using an optimal feature selection procedure that is based on the newly created feature evaluation index, decision tree, and local search approach. Ultimately, the neural network classifier's ideal structure is built by appropriately modifying its parameters and training it using the chosen best features. DTOF-ANN performs better than many of the current techniques, according to experimental results.

A. Subasi, et al. (2020) introduced a methodology for detecting sophisticated phishing websites [59]. We used various machine learning algorithms to determine whether a website was phishing or real. The implementation of a precise intelligent phishing website detection system involved the use of multiple classification approaches. The performance of the machine learning techniques is assessed using classification accuracy, F-measure, and area under receiver operating characteristic (ROC) curves (AUC). According to experimental data, Adaboost with SVM performed better than any other classification approach, attaining the greatest accuracy of 97.61%.

G. Sonowal, et al. (2017) offered PhiDMA (Phishing Detection using Multi-filter Approach), a multilayer model for phishing detection [60]. The PhiDMA model comprises five layers: the URL characteristics layer, the lexical signature layer, the string matching layer, the auto upgrade whitelist layer, and the accessibility score comparison layer. People with visual impairments can easily access a prototype implementation of the suggested PhiDMA paradigm thanks to its accessible interface. The experiment's outcome demonstrates that the model can identify phishing websites with an accuracy of 92.72%.

R. S. Rao, et al. (2019) provide an application called Jail-Phish that increases the precision of search engine-based methods by detecting Phishing Sites Hosted on Compromised Servers (PSHCS) and freshly registered, authentic websites. In order to determine the similarity score between the suspected website and the matched domain in the search results, Jail-Phish compares them [61]. While there are certain similarities between pages of the same website, such as in logos, favicons, photos, scripts, styles, and anchorlinks, there is also a significant amount of dissimilarity between sites in PSHCS. Therefore, in order to identify the PSHCS,

we employ the similarity score between the suspect site and the matching domain as a criterion. From the experimental results, it is observed that Jail-Phish achieved an accuracy of 98.61%, true positive rate of 97.77% and false positive rate less than 0.64%.

K. L. Chiew, et al. (2019) suggested the Hybrid Ensemble Feature Selection (HEFS), a novel feature selection framework for a machine learning-based phishing detection system. Primary feature subsets are generated in the first phase of HEFS using a unique Cumulative Distribution Function gradient (CDF-g) technique. These subsets are subsequently fed into a data perturbation ensemble to generate secondary feature subsets [62]. Using a function perturbation ensemble, the second phase extracts a set of baseline features from the secondary feature subsets. Using only 20.8% of the original features, the baseline features accurately identify 94.6% of phishing and legal websites, according to the total trial results, which indicate that HEFS works best when combined with the Random Forest classifier.

O. K. Sahingoz, et al. (2019) suggested a real-time anti-phishing system that makes use of features based on natural language processing (NLP) and seven distinct categorization algorithms [63]. Language independence, utilization of large amounts of genuine and phishing data, real-time execution, identification of new websites, independence from third-party services, and feature-rich classifiers are some of the characteristics that set the system apart from prior research in the literature. A new dataset is created and the experimental findings are tested on it in order to gauge the system's performance. The Random Forest algorithm with only NLP-based features performs best, detecting phishing URLs with an accuracy rate of 97.98%, based on experimental and comparative findings from the deployed classification algorithms.

J. Chen, et al. (2018) examined the application of this description-experience gap to human-automation interaction using a cyber domain phishing detection task [64]. In two tests, system reliability, description, and experience (i.e., feedback) were systematically varied in easy and difficult phishing detection tasks to measure participants' success in identifying phishing emails and their faith in the phishing detection system. The findings indicated that human performance within the system was significantly impacted by system reliability; however, the advantages of having a more dependable system can vary depending on the difficulty of the task. In terms of both objective and subjective trust measures, giving feedback also improved trust calibration; however, giving a description of system reliability only improved subjective trust.

S. Smadi, et al. (2018) proposed a novel framework which combines a neural network with reinforcement learning to detect phishing attacks in the online mode for the first time [65]. By using the concept of reinforcement learning to improve the system dynamically over time, the suggested model can adapt itself to create a new phishing email detection system that reflects changes in recently investigated behaviors. By automatically adding more emails to the offline dataset in the online mode, the suggested methodology addresses the issue of a limited dataset. To investigate any new phishing behaviors in the new dataset, a novel algorithm is suggested. We show that the suggested method can effectively manage zero-day phishing assaults through extensive testing on well-known data sets, attaining high accuracy, TPR, and TNR of 98.63%, 99.07%, and 98.19%, respectively.

H. Y. A. Abutair, et al. (2017) provided a robust global Phishing Threat Intelligence (PTI) environment by introducing a Multi-Agent System (MAS) as an adaptive intelligent technique that operates on top of distributed Case-Based Reasoning (CBR) Phishing Detection Systems (CBR-PDSs) as a Phishing Detection System Architecture (PDSA) that operates on a large scale globally [66]. PTI's worldwide partnerships reduce customers' vulnerability to sophisticated or difficult-to-detect spear phishing attacks, quarantine phishing threats through global threats sharing, and implement a proactive phishing detection technique. Additionally, integrating two intelligent systems into a single interactive architecture makes prediction easier, boosts accuracy, confronts advanced phishing threats' dynamic and variable behaviors with ease, and reduces false negative rates. In a PTI framework, the suggested architecture demonstrates the unified interaction between distributed CBR-PDSs and intelligent agents.

H. Y. A. Abutair, et al. (2017) presented the CBR-PDS, or Case-Based Reasoning (CBR) Phishing Detection System. As a fundamental component, it mostly relies on CBR approach [67]. Unlike previous classifiers that require extensive pre-training, the suggested method is very dynamic and flexible, as it can readily adjust to recognize novel phishing assaults using a comparatively small data set. We use a balanced set of 572 phishing and authentic URLs to test our system in various settings. The CBR-PDS system's accuracy surpasses 95.62%, according to experiments, but it greatly improves classification accuracy using a small feature set and sparse data sets.

M. Volkamer, et al. (2017) suggested the idea of TORPEDO, which offers reliable tooltips that are just-in-time and just-in-place, to enhance phish detection [68]. These aid users in

spotting phishing links in emails. The exact URL is displayed in TORPEDO's tooltips, with the site highlighted. In order to allow the user to examine the URL before clicking on a link, link activation is temporarily postponed. Additionally, TORPEDO offers an information diagram that explains how to detect phishes. We compared TORPEDO's efficacy to the worst-case "status bar" offered by other Web email interfaces. The accuracy of TORPEDO users in identifying authentic emails and spotting phishes was substantially higher (85.17% against 43.31% correct answers for phish).

C. L. Tan, et al. (2016) suggested a method for detecting phishing attacks that is based on the distinction between a webpage's target and real identities [69]. There are three stages to the suggested phishing detection method, known as PhishWHO. A unique weighted URL tokens method based on the N-gram model is proposed in the first phase, which involves extracting identity keywords from the website's textual contents. In the second stage, a search engine is used to discover the target domain name, which is then chosen based on characteristics that are significant to identity. A three-tiered identity matching mechanism is suggested in the last stage to assess the query webpage's validity. According to the total trial results, the suggested solution performs better than the traditional phishing detection techniques that were taken into consideration.

Yuancheng Li, et al. (2016) aimed to phishing website identification that is based on minimum enclosing ball support vector machine (BVM) is to achieve high speed and high accuracy in phishing website detection [70]. First, we analyze the website's topology structure using the DOM tree, and then we utilize the Web crawler to extract 12 topological characteristics from the website in order to improve the integrity of the feature vectors. Afterward, the BVM classifier detects the feature vectors. When compared to the standard SVM, this approach has a comparatively high detection precision, which counteracts the poor convergence speed on large-scale data.

M. Moghimi, et al. (2016) classified webpages using the support vector machine (SVM) method [71]. According to our tests, the suggested model has a 99.14% true positive and a 0.86% false negative alarm rate for identifying phishing pages in online banking. The results of the sensitivity analysis show how much more influence our suggested features have over conventional features. By using a related technique, we were able to extract the hidden knowledge from the suggested SVM model. To improve the functionality and usability of our suggested approach, we integrated the extracted rules into the PhishDetector browser plugin.

An evaluation of the installed browser extension shows that it has a high degree of accuracy and dependability in identifying phishing attempts in online banking. PhishDetector is also capable of identifying zero-day phishing attempts.

2.1 Research Gaps

Following are the various research gaps: -

1. The schemes which are already proposed for the phishing detection are unable to establish relation between attribute set and target set due to which optimal level of accuracy is not achieved.

2. The models for the phishing detection are based on the machine learning techniques. The machine learning models are supervised model. In the previous research no, one proposed unsupervised model for the phishing detection.

3. The models which are already proposed are unable to work on the QR codes for the phishing detection. The model needs to propose which can detect phishing from QR codes.

Chapter 3

Present Work

3.1. Problem Formulation

Phishing leverages email deception and fraudulent websites to extract sensitive information from unsuspecting individuals. Despite various efforts to address this issue, there remains a lack of comprehensive solutions to combat phishing effectively. Therefore, leveraging machine learning techniques is crucial in mitigating cybercrimes, particularly those involving phishing attacks. The proposed study utilizes a phishing URL-based dataset sourced from a renowned repository, comprising attributes of both phishing and legitimate URLs gathered from over 11,000 websites. Following preprocessing, numerous machine learning algorithms

are employed and tailored to thwart phishing URLs and safeguard users. These algorithms include decision tree (DT), linear regression (LR), random forest (RF), naive Bayes (NB), gradient boosting classifier (GBM), K-neighbors classifier (KNN), support vector classifier (SVC), and a novel hybrid LSD model. The hybrid LSD model, which combines logistic regression, support vector machine, and decision tree (LR+SVC+DT) with both soft and hard voting mechanisms, is particularly promising in defending against phishing attacks with remarkable accuracy and efficiency. It is analyzed that hybrid method give good performance of phishing URL's about with the change in the trend it donot perform of phishing QR codes. The model needs to propose which can perform well on phishing QR codes.

3.2. Objectives

Study and Analysis of Phishing QR Detection Techniques

Phishing QR detection techniques can be broadly categorized into three main approaches: heuristic-based methods, feature-based methods, and machine learning-based methods. Heuristic-based methods rely on predefined rules and patterns to identify malicious QR codes. These methods are straightforward and fast but often lack the flexibility to detect new and evolving phishing techniques. Feature-based methods, on the other hand, analyze specific characteristics of QR codes, such as the embedded URL's structure, domain reputation, and the presence of obfuscation techniques. While these methods provide better detection rates than heuristic-based methods, they still fall short in identifying sophisticated phishing attacks that continually adapt to bypass detection mechanisms. Machine learning-based methods represent the most advanced and promising approach to phishing QR detection. These methods leverage various algorithms to learn from vast datasets, identifying subtle patterns and anomalies that might indicate phishing attempts. Supervised learning techniques, such as decision trees, support vector machines, and neural networks, have shown significant success in this domain. Additionally, unsupervised learning methods, like clustering and anomaly detection, can identify new phishing techniques by recognizing deviations from normal QR code behavior. Despite their effectiveness, machine learning-based methods require substantial computational resources and well-curated training datasets to achieve high detection accuracy.

Implementation of Machine Learning Methods for Phishing QR Detection

To enhance phishing QR detection, various machine learning methods can be implemented and evaluated. Supervised learning algorithms such as Random Forest, Gradient Boosting Machines, and Convolutional Neural Networks (CNNs) can be trained on labeled datasets containing both benign and phishing QR codes. These algorithms can learn complex patterns and relationships between the QR code features and their corresponding labels, enabling accurate classification of new, unseen QR codes. In addition to supervised learning, semi-supervised and unsupervised learning methods can also be employed. Semi-supervised learning can leverage a small amount of labeled data combined with a larger pool of unlabeled data, effectively reducing the dependency on extensive labeled datasets. Techniques such as self-training and co-training can iteratively improve the model's performance. Unsupervised learning methods, like K-means clustering and autoencoders, can detect anomalies in QR codes that may indicate phishing attempts. These methods are particularly useful in identifying zero-day phishing attacks that have not been seen before.

Designing a Novel Approach for Phishing QR Detection

Building on the strengths and limitations of existing methods, a novel approach for phishing QR detection can be designed. This approach could integrate multiple machine learning techniques to leverage their complementary strengths. For instance, a hybrid model combining supervised learning for initial classification and unsupervised learning for anomaly detection can provide robust and adaptive phishing QR detection. The proposed approach could involve several stages. Initially, a CNN could extract features from the QR code images, capturing both visual and contextual information. These features could then be fed into a Random Forest classifier for preliminary classification. Concurrently, an autoencoder could analyze the QR code data to detect any anomalies, flagging potential phishing codes that deviate from the learned patterns. By combining the outputs of these models, the hybrid approach can achieve higher detection accuracy and adaptability.

Implementation and Comparison with Existing Methods

The proposed approach will be implemented and compared with existing methods using standard evaluation metrics such as accuracy, precision, and recall. Accuracy measures the overall correctness of the model, while precision and recall provide insights into the model's ability to identify true positives and minimize false positives. A comprehensive dataset containing a diverse range of QR codes, including benign and phishing examples, will be

used for training and testing. Initial experiments will focus on optimizing the hyperparameters of each component model to ensure the best performance. Cross-validation techniques will be employed to validate the model's robustness and prevent overfitting. The hybrid model's performance will be benchmarked against existing heuristic-based, feature-based, and standalone machine learning models. Expected outcomes include improved detection rates, lower false positive rates, and enhanced capability to identify new phishing techniques.

Following are the various objectives: -

1. To study and analyse various phishing QR detection techniques

2. To implement various machine learning methods for the phishing QR detection techniques

3. To design novel approach for the phishing QR detection techniques

4. Implement proposed approach and compare with existing in terms of accuracy, precision and recall

3.3. Research Methodology

The QR code phishing can be detected in diverse stages such as to pre-process the data, extract the attributes and classification. The research methodology is defined as: -

1. **Data set input and Pre-processing**: - The data will be taken as input and it will have processed to remove and missing values from the dataset.

2. **Segmentation:-** The technique of snake segmentation will be applied for segmenting the images. The Snake segmentation technique is inspired from the raster scan due to which it will cover maximum edges of the image SAC algorithm [6-8] is employed for modelling a parameterized primary contour curve in the image space, and an energy function (EF) is put forward to characterize the shape of the area in accordance with the internal and external power. The features of curve help to determine the first one and the attributes of image assist in describing the external energy including curvature, curve length, etc. EF is diminished to converge the primary contour curve $C(s) = (x(s), y(s), s \in [0,1])$ continuously to the boundary of the destination region in the restraints of both energies:

$$E(C) = \int_0^1 \alpha E_{int}\big(C(s)\big) + E_{img}\Big(C(s) + \gamma E_{con}\big(C(s)\big)\Big)\, ds - -(1)$$

Three portions are included in EF such as E_{int} uses to illustrate the internal energy for ensuring that the curve if smooth and regular; E_{img} is utilized to denote the image energy, assigned in accordance with the desired position attributes like edges; the constrained energy is represented with E_{con}. SAC algorithm is useful as the geometric restraints are taken in account. Avoiding the quality of image, the major focus is on extracting the closed boundaries. However, some limitations are occurred still. The challenging task is of tackling the region due to its dependence on the first contour. The position, shape and number of control points are capable of acquiring the preferred impact only in case of selecting an appropriate primary contour.

3. **Classification**: - This stage focuses on splitting the entire data into training and testing. The voting classification technique is implemented to classify the input images. Multiple classifiers are integrated in this technique to classify the QR codes. To prediction the phishing QR model of transfer learning is applied which is the combination of VGG16 and CNN model. The VGG16 is used as the base model over which CNN model is used for the training.

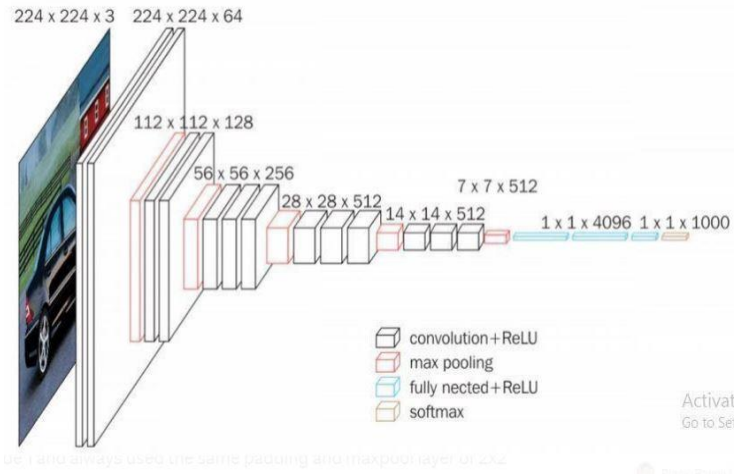


Figure 10: VGG16 Model Architecture

Following are the various specifications of VGG16 Model: -

1. This model illustrates sixteen layers with 16 and these layers contain weights. Around 13 conv layers, 5 MP layers and 3 dense layers are included. However, only 16 weight layers are there.

2. The tensor size as 224, 244 with 3 RGB channel is utilized for input in this model

3. This model does not contain a large number of hyper-parameters as it deploys conv layers of 3x3 filter with stride 1 and the same padding and max-pool layer of 2x2 filter having stride 2.

4. The arrangement of conv and max pool layers is done in consistent way in the entire framework.

5. 64 filters comprised in Conv-1 Layer, 128 in Conv-2, 256 in Conv-3, 512 in Conv 4 and Conv 5

6. Three FC layers has a stack of conv layers: 4096 channels are included in primary two, the last leads to perform 1000-way ILSVRC classification. Therefore, one thousand channels are comprised. The last one is known as the soft-max layer.

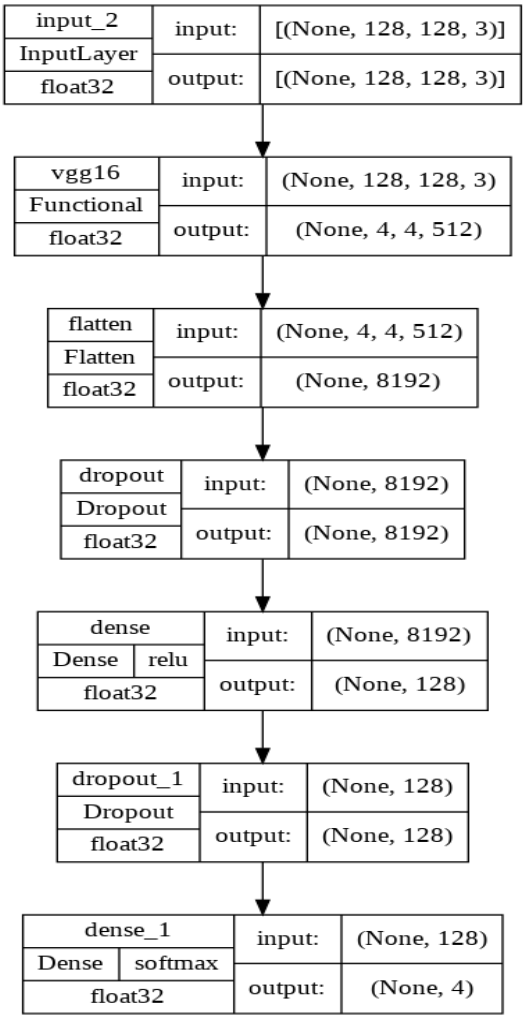


Figure 11: Proposed Transfer Learning Model

Chapter 4

Results and Discussion

4.1. Platform Used for Implementation

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, http://www.python.org/, and can be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

4.2. Results

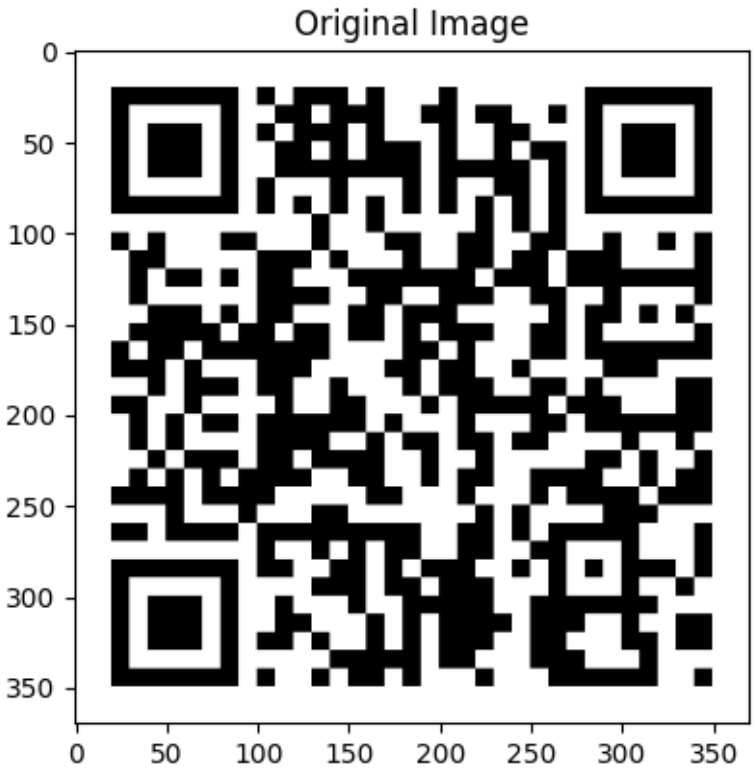


Figure 12: Input Sample Image

As shown in figure 12, the dataset of QR codes is taken as input for the phishing QR codes detections. The sample image which the test image is shown and displayed.

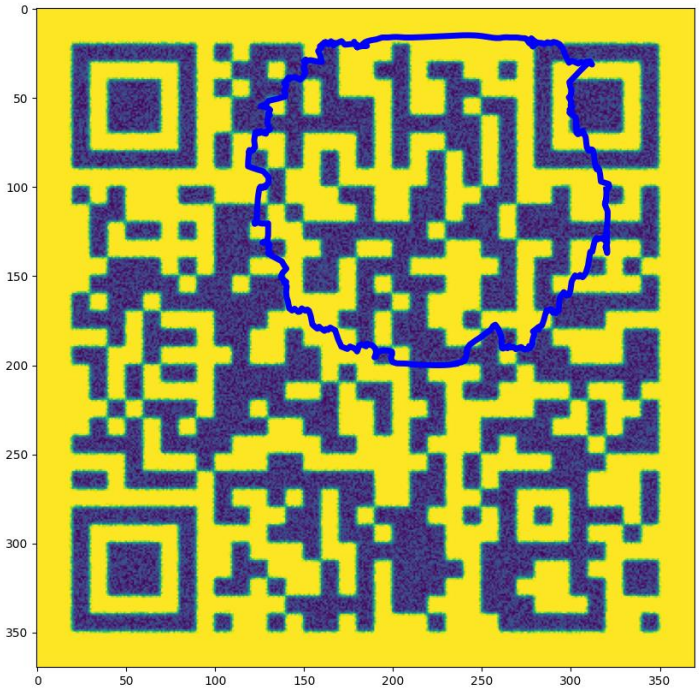


Figure 13: Snake based Segmentation

As shown in figure 13, the snake based segmentation is applied which will segment region interest from the image.

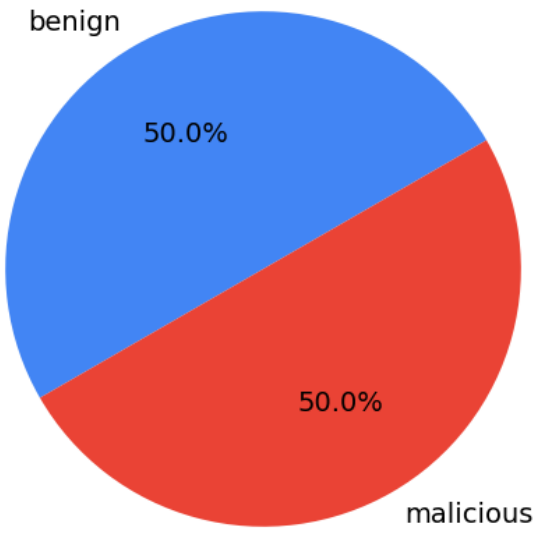


Figure 14: Class Distribution

As shown in figure 14, the whole dataset is divided into malicious and benign class. Both the classes have equal amount of data which is 50 and 50 percent.

Figure 15: Dataset Images

As shown in figure 15, the whole dataset has two classes which are benign and malicious. The sample images of both the classes is shown in the plotting.

Figure 16: Training and Model Loss

As shown in figure 16, the model is trained on the 4 epoch values. The model training accuracy and model loss is illustrated in the picture.

Table 1: Result Comparison

Model	Accuracy	Precision	Recall
Random Forest	66 Percent	56 Percent	66 Percent
SVM	77.59 Percent	78 Percent	78 Percent
KNN	69.88 Percent	70 Percent	70 Percent
Proposed Model	91 Percent	91.2 Percent	92 Percent

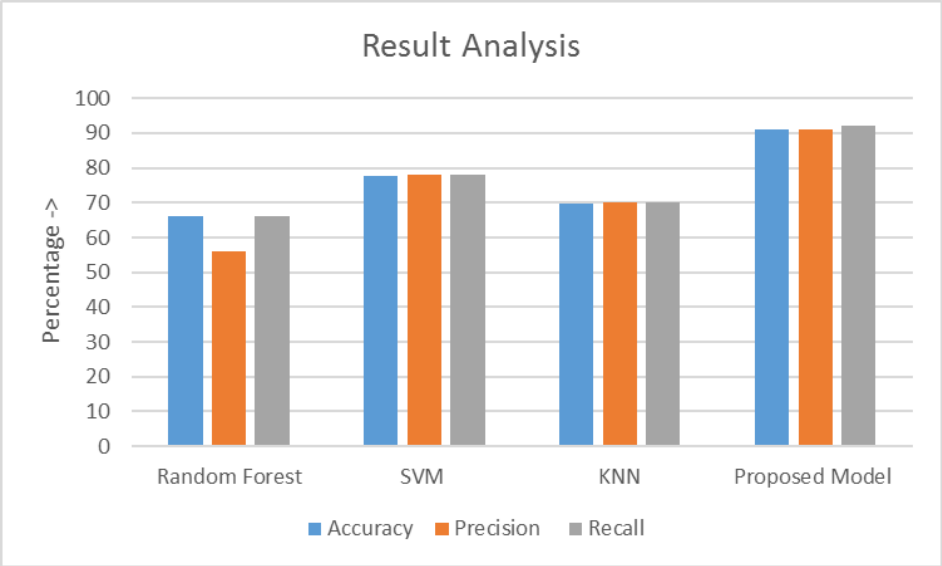


Figure 17: Result Analysis of Proposed Model

Figure 17 depicts that the results of the introduced approach are compared with the SVM, KNN and Random Forest. The proposed model achieves accuracy upto 95 percent, KNN Model, Random Forest Model and SVM Model has accuracy 69.88, 66 Percent and 77.59 percent respectively for the brain tumour detection which proves reliability of proposed model.

References

[1] H. A. M. Wahsheh and F. L. Luccio, ‘‘Security and privacy of QR code applications: A comprehensive study, general guidelines and solutions,’’ Information, vol. 11, no. 4, p. 217, Apr. 2020, doi : 10.1007/978-3-319-65127-9.

[2] V. Mavroeidis and M. Nicho, ‘‘Quick response code secure: A cryptographically secure anti-phishing tool for QR code attacks,’’ in Computer Network Security (Lecture Notes in Computer Science), vol. 10446. Cham, Switzerland: Springer, 2017 doi: 10.1007/978-3-319-65127-9.

[3] A. Y. Alnajjar, M. Anbar, S. Manickam, O. Elejla, and H. El-Taj, ‘‘QRphish: An automated QR code phishing detection approach,’’ J. Eng. Appl. Sci., vol. 11, no. 3, pp. 553–560, 2016

[4] X. Xiao, D. Zhang, G. Hu, Y. Jiang, and S. Xia, ‘‘CNN–MHSA: A convolutional neural network and multi-head self-attention combined approach for detecting phishing websites,’’ Neural Netw., vol. 125, pp. 303–312, May 2020

[5] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, ‘‘Hybrid rule-based solution for phishing URL detection using convolutional neural network,’’ Wireless Commun. Mobile Comput., vol. 2021, Sep. 2021, Art. no. 8241104.

[6] I. Ortiz Garcés, M. F. Cazares and R. O. Andrade, "Detection of Phishing Attacks with Machine Learning Techniques in Cognitive Security Architecture," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 366-370, doi: 10.1109/CSCI49370.2019.00071.

[7] F. Khan, M. Hasan and K. Das, "A weighted ensemble model for phishing website detection using random forest and deep neural network," 2023 5th International Conference on Sustainable Technologies for Industry 5.0 (STI), Dhaka, Bangladesh, 2023, pp. 1-6, doi: 10.1109/STI59863.2023.10465064

[8] F. Hossain, L. Islam and M. N. Uddin, "PhishRescue: A Stacked Ensemble Model to Identify Phishing Website Using Lexical Features," 2022 5th International Conference of

Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 2022, pp. 342-347, doi: 10.1109/IC2IE56416.2022.9970179.

[9] 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, vol. 11, pp. 92523-92539, 2023, doi: 10.1109/ACCESS.2023.3291811

[10] S. Merugula, K. S. Kumar, S. Muppidi and C. Vidyadhari, "Stop Phishing : Master Anti-Phishing Techniques," 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Vijaypur, India, 2022, pp. 1-5, doi: 10.1109/NKCon56289.2022.10126569.

[11] C. -Y. Wu, C. -C. Kuo and C. -S. Yang, "Phishing Detection with Browser Extension Based on Machine Learning," 2023 18th Asia Joint Conference on Information Security (AsiaJCIS), Koganei, Japan, 2023, pp. 81-87, doi: 10.1109/AsiaJCIS60284.2023.00023.

[12] S. Ariyadasa, S. Fernando and S. Fernando, "Combining Long-Term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites Using URL and HTML," in IEEE Access, vol. 10, pp. 82355-82375, 2022, doi: 10.1109/ACCESS.2022.3196018.

[13] A. Goyal et al., "Phishing Attack Detection Using MapReduce and Machine Learning," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-5, doi: 10.1109/ICCCNT61001.2024.10726226.

[14] Mahesh, Ananth and Dheepthi, "Using Machine Learning to Detect and Classify URLs: A Phishing Detection Approach," 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2023, pp. 1285-1291, doi: 10.1109/ICESC57686.2023.10193559.

[15] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen and R. Ramli, "A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2021, pp. 82-87, doi: 10.1109/ISCAIE51753.2021.9431794.

[16] N. Jindal, D. Rastogi, K. Joshi and D. Gupta, "Identification of Phishing Attacks using Machine Learning," 2023 Seventh International Conference on Image Information Processing (ICIIP), Solan, India, 2023, pp. 941-946, doi: 10.1109/ICIIP61524.2023.10537706.

[17] J. Siddhesh Vijay, K. Kulkarni and A. Arya, "Metaheuristic Optimization of Neural Networks for Phishing Detection," 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824203

[18] A. Saxena, A. Arora, S. Saxena and A. Kumar, "Detection of web attacks using machine learning based URL classification techniques," 2022 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2022, pp. 1-13, doi: 10.1109/CONIT55038.2022.9847838.

[19] A. Singh and P. K. Roy, "Malicious URL Detection using Multilayer CNN," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 2021, pp. 340-345, doi: 10.1109/3ICT53449.2021.9581880.

[20] C. -Y. Wu, C. -C. Kuo and C. -S. Yang, "A Phishing Detection System based on Machine Learning," 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA), Tainan, Taiwan, 2019, pp. 28-32, doi: 10.1109/ICEA.2019.8858325.

[21] S. Ismail, M. H. Alkawaz and A. E. Kumar, "Quick Response Code Validation and Phishing Detection Tool," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2021, pp. 261-266

[22] 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, vol. 11, pp. 92523-92539, 2023

[23] B. Herlina and H. Soeparno, “Machine Learning Model to Improve Classification Performance in The Process of Detecting Phishing URLs in QR Codes”, Journal of Theoretical and Applied Information Technology, vol. 10, no. 18, pp. 13-20, 2023

[24] M. Sahay, S. Vanjale and M. Mane, “Software as Service Attack Detection and Prevention for Deceitful QR code”, International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 4, pp. 454-462, 2024

[25] G. A. Amoah and H.-A. J.B., “QR Code Security: Mitigating the Issue of Quishing (QR Code Phishing)”, International Journal of Computer Applications, vol. 184, no. 33, 2022

[26] M. Thakare, S. Patil, H. Pawar, A. Sawant and K. Vatekar, “Qrshield: Qr Code-Based Attack Detection and Prevention for Software-As-A-Service (Saas) Applications”, International Research Journal of Modernization in Engineering Technology and Science, vol. 5, no. 4, pp. 12-20, 2023s

[27] G. R. Charan and K. D. Thilak, "Detection of Phishing Link and QR Code of UPI Transaction using Machine Learning," 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, 2023, pp. 658-663, doi: 10.1109/ICIMIA60377.2023.10426613.

[28] B. B. Gupta et al., “Advanced BERT and CNN-Based Computational Model for Phishing Detection in Enterprise Systems,” Computer Modeling in Engineering & Sciences, vol. 0, no. 0, pp. 1–10, Jan. 2024, doi: https://doi.org/10.32604/cmes.2024.056473.

[29] Abdulla Al-Subaiey, M. Al-Thani, Naser Abdullah Alam, Kaniz Fatema Antora, Amith Khandakar, and SM, “Novel interpretable and robust web-based AI platform for phishing email detection,” Computers & Electrical Engineering, vol. 120, pp. 109625–109625, Sep. 2024, doi: https://doi.org/10.1016/j.compeleceng.2024.109625.

[30] Sakib Shahriar Shafin, “An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning,” Data Science and Management, Aug. 2024, doi: https://doi.org/10.1016/j.dsm.2024.08.004.

[31] Brij Bhooshan Gupta, Akshat Gaurav, Razaz Waheeb Attar, V. Arya, A. Alhomoud, and Kwok Tai Chui, “Optimized Phishing Detection with Recurrent Neural Network and Whale Optimizer Algorithm,” Computers, materials & continua/Computers, materials & continua (Print), vol. 0, no. 0, pp. 1–10, Jan. 2024, doi: https://doi.org/10.32604/cmc.2024.050815.

[32] A. B. Majgave and N. L. Gavankar, “Automatic phishing website detection and prevention model using transformer deep belief network,” Computers & Security, vol. 147, p. 104071, Dec. 2024, doi: https://doi.org/10.1016/j.cose.2024.104071.

[33] D. Sturman, E. A. Bell, J. C. Auton, G. R. Breakey, and M. W. Wiggins, “The roles of phishing knowledge, cue utilization, and decision styles in phishing email detection,” Applied Ergonomics/Applied ergonomics, vol. 119, pp. 104309–104309, Sep. 2024, doi: https://doi.org/10.1016/j.apergo.2024.104309.

[34] Jawhara Aljabri et al., “Hybrid stacked autoencoder with dwarf mongoose optimization for Phishing attack detection in internet of things environment,” Alexandria Engineering Journal, vol. 106, pp. 164–171, Jul. 2024, doi: https://doi.org/10.1016/j.aej.2024.06.070.

[35] Dennik Baltuttis and T. Teubner, “Effects of Visual Risk Indicators on Phishing Detection Behavior: An Eye-Tracking Experiment,” Computers & Security, pp. 103940–103940, Jun. 2024, doi: https://doi.org/10.1016/j.cose.2024.103940.

[36] N. Kamble and Dr. Nilamadhab Mishra, “Hybrid Optimization Enabled Squeeze Net For Phishing Attack Detection,” Computers & Security, pp. 103901–103901, May 2024, doi: https://doi.org/10.1016/j.cose.2024.103901.

[37] F. Rashid, B. Doyle, Soyeon Caren Han, and S. Seneviratne, “Phishing URL detection generalisation using Unsupervised Domain Adaptation,” Computer Networks, vol. 245, pp. 110398–110398, May 2024, doi: https://doi.org/10.1016/j.comnet.2024.110398.

[38] R. J. van Geest, G. Cascavilla, J. Hulstijn, and N. Zannone, “The applicability of a hybrid framework for automated phishing detection,” Computers & Security, vol. 139, p. 103736, Apr. 2024, doi: https://doi.org/10.1016/j.cose.2024.103736.

[39] E. Zhu, K. Cheng, Z. Zhang, and H. Wang, “PDHF: Effective Phishing Detection Model Combining Optimal Artificial and Automatic Deep Features,” Computers & Security, vol. 136, pp. 103561–103561, Jan. 2024, doi: https://doi.org/10.1016/j.cose.2023.103561.

[40] A. Prasad and S. Chandra, “PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning,” Computers & Security, p. 103545, Oct. 2023, doi: https://doi.org/10.1016/j.cose.2023.103545.

[41] L. Wang, M. Xu, and H. Cheng, “Phishing scams detection via temporal graph attention network in Ethereum,” Information Processing and Management, vol. 60, no. 4, pp. 103412–103412, Jul. 2023, doi: https://doi.org/10.1016/j.ipm.2023.103412.

[42] C. C. L. Tan, K. L. Chiew, K. S. C. Yong, Y. Sebastian, J. C. M. Than, and W. K. Tiong, “Hybrid phishing detection using joint visual and textual identity,” Expert Systems with Applications, vol. 220, p. 119723, Jun. 2023, doi: https://doi.org/10.1016/j.eswa.2023.119723.

[43] M. Abdullah Alohali et al., “Metaheuristics with deep learning driven phishing detection for sustainable and secure environment,” Sustainable Energy Technologies and Assessments, vol. 56, p. 103114, Mar. 2023, doi: https://doi.org/10.1016/j.seta.2023.103114.

[44] D. Sturman et al., “The role of cue utilization in the detection of phishing emails,” Applied Ergonomics, vol. 106, p. 103887, Jan. 2023, doi: https://doi.org/10.1016/j.apergo.2022.103887.

[45] K. Omari, “Phishing Detection using Gradient Boosting Classifier,” Procedia Computer Science, vol. 230, pp. 120–127, Jan. 2023, doi: https://doi.org/10.1016/j.procs.2023.12.067.

[46] M. M. Alani and H. Tawfik, “PhishNot: A Cloud-Based Machine-Learning Approach to Phishing URL Detection,” Computer Networks, vol. 218, p. 109407, Dec. 2022, doi: https://doi.org/10.1016/j.comnet.2022.109407.

[47] D.-J. Liu, G.-G. Geng, and X.-C. Zhang, “Multi-scale semantic deep fusion models for phishing website detection,” Expert Systems with Applications, vol. 209, p. 118305, Dec. 2022, doi: https://doi.org/10.1016/j.eswa.2022.118305.

[48] S. M. Alshahrani, N. Ahmed Khan, J. Almalki, and W. Al Shehri, “URL Phishing Detection Using Particle Swarm Optimization and燚ata燻ining,” Computers, Materials & Continua, vol. 73, no. 3, pp. 5625–5640, 2022, doi: https://doi.org/10.32604/cmc.2022.030982.

[49] Hadil Shaiba, J. S. Alzahrani, M. M. Eltahir, R. Marzouk, H. Mohsen, and Manar Ahmed Hamza, “Hunger Search Optimization with Hybrid Deep Learning Enabled Phishing

Detection and Classification Model," vol. 73, no. 3, pp. 6425–6441, Jan. 2022, doi: https://doi.org/10.32604/cmc.2022.031625.

[50] T. O. Ojewumi, G. O. Ogunleye, B. O. Oguntunde, O. Folorunsho, S. G. Fashoto, and N. Ogbu, "Performance evaluation of machine learning tools for detection of phishing attacks on web pages," Scientific African, vol. 16, p. e01165, Jul. 2022, doi: https://doi.org/10.1016/j.sciaf.2022.e01165.

[51] S. Minocha and B. Singh, "A novel phishing detection system using binary modified equilibrium optimizer for feature selection," Computers & Electrical Engineering, vol. 98, p. 107689, Mar. 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.107689.

[52] F. Zheng, Q. Yan, V. C. M. Leung, F. Richard Yu, and Z. Ming, "HDP-CNN: Highway deep pyramid convolution neural network combining word-level and character-level representations for phishing website detection," Computers & Security, vol. 114, p. 102584, Mar. 2022, doi: https://doi.org/10.1016/j.cose.2021.102584.

[53] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," Journal of King Saud University - Computer and Information Sciences, Dec. 2019, doi: https://doi.org/10.1016/j.jksuci.2019.12.005.

[54] R. Wazirali, R. Ahmad, and A. A.-K. Abu-Ein, "Sustaining accurate detection of phishing URLs using SDN and feature selection approaches," Computer Networks, vol. 201, p. 108591, Dec. 2021, doi: https://doi.org/10.1016/j.comnet.2021.108591.

[55] Y. Al-Hamar, H. Kolivand, M. Tajdini, T. Saba, and V. Ramachandran, "Enterprise Credential Spear-phishing attack detection," Computers & Electrical Engineering, vol. 94, p. 107363, Sep. 2021, doi: https://doi.org/10.1016/j.compeleceng.2021.107363.

[56] X. Xiao et al., "Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets," Computers & Security, vol. 108, p. 102372, Sep. 2021, doi: https://doi.org/10.1016/j.cose.2021.102372.

[57] P. A. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," Computers & Security, vol. 104, p. 102123, May 2021, doi: https://doi.org/10.1016/j.cose.2020.102123.

[58] E. Zhu, Y. Ju, Z. Chen, F. Liu, and X. Fang, “DTOF-ANN: An Artificial Neural Network phishing detection model based on Decision Tree and Optimal Features,” Applied Soft Computing, vol. 95, p. 106505, Oct. 2020, doi: https://doi.org/10.1016/j.asoc.2020.106505.

[59] A. Subasi and E. Kremic, “Comparison of Adaboost with MultiBoosting for Phishing Website Detection,” Procedia Computer Science, vol. 168, pp. 272–278, 2020, doi: https://doi.org/10.1016/j.procs.2020.02.251.

[60] G. Sonowal and K. S. Kuppusamy, “PhiDMA – A phishing detection model with multi-filter approach,” Journal of King Saud University - Computer and Information Sciences, Jul. 2017, doi: https://doi.org/10.1016/j.jksuci.2017.07.005.

[61] R. S. Rao and A. R. Pais, “Jail-Phish: An improved search engine based phishing detection system,” Computers & Security, vol. 83, pp. 246–267, Jun. 2019, doi: https://doi.org/10.1016/j.cose.2019.02.011.

[62] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, “A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,” Information Sciences, vol. 484, pp. 153–166, May 2019, doi: https://doi.org/10.1016/j.ins.2019.01.064.

[63] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, Mar. 2019, doi: https://doi.org/10.1016/j.eswa.2018.09.029.

[64] J. Chen, S. Mishler, B. Hu, N. Li, and R. W. Proctor, “The description-experience gap in the effect of warning reliability on user trust and performance in a phishing-detection context,” International Journal of Human-Computer Studies, vol. 119, pp. 35–47, Nov. 2018, doi: https://doi.org/10.1016/j.ijhcs.2018.05.010.

[65] S. Smadi, N. Aslam, and L. Zhang, “Detection of online phishing email using dynamic evolving neural network based on reinforcement learning,” Decision Support Systems, vol. 107, pp. 88–102, Mar. 2018, doi: https://doi.org/10.1016/j.dss.2018.01.001.

[66] H. Y. A. Abutair and A. Belghith, “A Multi-Agent Case-Based Reasoning Architecture for Phishing Detection,” Procedia Computer Science, vol. 110, pp. 492–497, 2017, doi: https://doi.org/10.1016/j.procs.2017.06.131.

[67] H. Y. A. Abutair and A. Belghith, “Using Case-Based Reasoning for Phishing Detection,” Procedia Computer Science, vol. 109, pp. 281–288, 2017, doi: https://doi.org/10.1016/j.procs.2017.05.352.

[68] M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz, “User experiences of TORPEDO: TOoltip-poweRed Phishing Email DetectiOn,” Computers & Security, vol. 71, pp. 100–113, Nov. 2017, doi: https://doi.org/10.1016/j.cose.2017.02.004.

[69] C. L. Tan, K. L. Chiew, K. Wong, and S. N. Sze, “PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder,” Decision Support Systems, vol. 88, pp. 18–27, Aug. 2016, doi: https://doi.org/10.1016/j.dss.2016.05.005.

[70] Yuancheng Li, Liqun Yang “A minimum enclosing ball-based support vector machine approach for detection of phishing websites,” Optik, vol. 127, no. 1, pp. 345–351, Jan. 2016, doi: https://doi.org/10.1016/j.ijleo.2015.10.078.

[71] M. Moghimi and A. Y. Varjani, “New rule-based phishing detection method,” Expert Systems with Applications, vol. 53, pp. 231–242, Jul. 2016, doi: https://doi.org/10.1016/j.eswa.2016.01.028.