International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 10, NCREHAASTE - 2026

SOCmind: An Explainable Multi-Vector Framework for Phishing, Anti-Evasion, and QR-Code Threat Detection

Imran Khan
Department of Computer Application,
MIT, Moradabad, Uttar Pradesh, India
imranofficialtask@gmail.com

Mohammad Aayaj
Department of Computer Application,
MIT, Moradabad, Uttar Pradesh,
India ayazcho60@gmail.com

Ruby Faizan
Assistant Professor
Department of Computer Application,
MIT, Moradabad, Uttar Pradesh,
India Faizan.ruby@gmail.com

ABSTRACT:

SOCmind is the multi-vector phishing, anti-evasion, and QR-code detection framework. The phishing attacks' domain sets, infrastructure, and delivery mechanisms are evolving and adapting to avoid traditional blacklist-based protection solutions. QR codes can contain hidden and obfuscated malicious destination URLs for users. The framework combines URL and domain features, WHOIS/DNS analysis, and TLS and redirection checks, typo-squatted and homoglyph detection, anti-evasion analysis, QR-code decoding, and 11-featured Random Forest with percentage-based explanation of decisions. The paper details the research question, system architecture, and methodology, as well as a suggested evaluation approach for phishing, anti-evasion, and QR-code-based attacks.

Index Terms - Phishing detection, anti-evasion, QR-code phishing, quishing, typosquatting, homoglyph, Random Forest, explainable AI, cybersecurity.

INTRODUCTION:

Phishing is defined by MITRE ATT&CK as a social engineering technique that utilizes electronic means to trick an individual into giving up access to resources, credentials, or other valuable information, such as by appearing to originate from a trusted entity or landing on a trusted domain. Specifically, within the ATT&CK framework, phishing is categorized as technique T1566 within Initial Access, which is further divided into spearphishing links, attachments, services, and voice [15]. The volume of phishing activity is massive and growing, with 971,181 phishing incidents reported in the first quarter of 2026 alone, which is a 13.8 percent increase compared to the last quarter of 2025 [3]. Thus, the ability to detect previously unseen and altered phishing resources that may not have a known reputation is an important capability. The problem becomes more complicated when one deliberately manipulates exposed features. The adversarial research literature reports phishing detectors to be vulnerable to evasion attempts via perturbations learned in an evasion space: these preserve the malicious intent while perturbing the sample towards the benign decision region [4, 29]. Domain impersonation also complicates detection, as typosquatting, homoglyphs, and combisquatting can appear similar to the legitimate brand [21, 24]. Thus, a practical phishing detector must consider both the lexical similarity and the visual presentation of the destination.

QR codes complicate the delivery of phishing attacks since the resource destination is hidden inside a scannable glyph. In effect, Krombholz et al. note that malicious QR codes can carry phishing URLs that are challenging to detect for the untrained eye and that QR readers historically lacked strong security notifications [9]. Field observations of QR code phishing campaigns ("quishing") confirm that convenience-driven attacks are often successful in convincing users into engaging with a malicious QR payload [22]. Thus, analyzing QR code payloads and destinations becomes necessary for detecting such attacks.

Explainability is another factor that contributes to the operational utility of a phishing detector. A black-box detector that only outputs a binary classification fails to report which evidence was responsible for the detection. LIME shows the value of local surrogate explanations for classification models [19], and similarly, SHAP notes the importance of feature attribution for individual predictions [14]. Moreover, NIST's explainable-AI guidelines highlight the need for human-interpretable models that faithfully represent the system under study [18]. As such, SOCmind does not treat explanation as a post hoc analysis but rather integrates risk attribution during the inference stage in the Random Forest module.

In this paper, we analyze SOCmind – an implemented multi-vector threat assessment framework focused on phishing, anti-evasion, and QR code analysis. The detection layer uses lexical features, WHOIS and DNS telemetry, transport security inspection, and typosquatting/homoglyph analysis. The anti-evasion layer employs dual-profile cloaking detection, DOM form exfiltration analysis, perceptual-hash matching for logos, and ID-related infrastructure analysis. Lastly, the QR code processing module extracts and decodes supported QR code payloads while analyzing the chosen URI scheme.

A. RESEARCH CONTRIBUTIONS

- An integrated phishing-detection architecture that combines lexical, DNS/WHOIS, transport security, and typosquatting/homoglyph evidence,
- An anti-evasion analysis layer that addresses dual-profile cloaking, DOM form exfiltration, perceptual hashing of brand comparison, and specified infrastructure indicators,
- A research-suitable evaluation method that untangles phishing-detection, anti-evasion analysis, QR-code decoding, and avoids unsubstantiated performance claims.
- A QR-code forensics pipeline for image/PDF extraction, recursive Base64 and URL-hex decoding, and analysis of specified custom URI schemes.

B. PAPER ORGANIZATION

Section II surveys related work. Section III identifies the research problem, outlines the research gap, details the objectives and scope of this study, and specifies the research questions. Section IV describes the SOCmind architecture. Section V explains the implemented detection and anti-evasion methodology in detail. Section VI introduces the QR-code analysis pipeline as well as the stage of explainable Random Forest. Section VII details the experimental design, evaluation metrics, and benchmarking procedure. Section VIII interprets the results and reports on the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 10, NCREHAASTE - 2026

experiments. Section IX discusses limitations, while Section X concludes the study and suggests directions for future research.

II. LITERATURE REVIEW

A. PHISHING DETECTION AND URL BASED ANALYSIS

Initial research on phishing detection focused on content-based information retrieval and web page features. The CANTINA method utilized TF-IDF and characteristics of contents to classify phishing webpages [30]. CANTINA+ expanded the feature set and applied machine learning methods, near-duplicate hashing, and login-form filtering [28]. This line of work highlighted the importance of using combinations of characteristics typical of phishing websites rather than just URL blacklisting. The later surveys confirmed that URL-based, host-based, content-based, and hybrid approaches are the most widely studied [12, 23]. In addition, more recent articles have demonstrated that even the choice of rule-based, login-URL-based, Random Forest, and feature selection methods can significantly affect the outcome of phishing detection [8, 17, 20, 27].

Several URL-based articles studied the properties of characters, special symbols, length, subdomains, domains, randomness, and words. The URLNet method demonstrated the effectiveness of using representation learning over URL characters and words [11]. Other works compared the performance of Random Forest, SVM, and other classical machine learning models trained on lexical URL features [26], [2], [8, 17, 20, 27]. Notably, lexical features allow for quickly flagging phishing domains before engaging in any resource-intensive interactions. However, domains that masquerade as legitimate ones may be difficult to distinguish from real ones using lexical analysis.

B. DOMAIN IMPERSONATION, TYPOSQUATTING, AND HOMOGLYPHS

Domain-name impersonation is a common phishing technique that has been studied extensively. Large-scale measurements of typo-squatting and other squatting methods have been performed [24]. The case of internationalized domain names adds additional challenges since characters from different writing systems may be confusable. Previous research on homograph attacks has triggered the development of normalization, Punycode analysis, and script-aware approaches [21]. SOCmind applies these ideas by analyzing Punycode, checking for Cyrillic lookalikes, leet-speak, and combisquatting.

C. ADVERSARIAL EVASION AND CLOAKING

The initial assumption is wrong; adversaries may alter phishing samples in such a way that they shift towards the less-detectable region of the feature space. SpacePhish demonstrated this evasion hypothesis on phishing samples [4]. Multi-SpacePhish later extended the finding by analyzing additional modification spaces, thus emphasizing the importance of testing detectors on modified malicious inputs [29]. Separate research on phishing webpages also observed that attackers could benefit from the discrepancy between the scanners' behavior and actual user interaction [5], while SOCmind's anti-evasion design is more limited, utilizing cloaking probes with dual-profile, DOM form-exfiltration audit, pHash visual comparison and infrastructure-based checks as implemented within the project.

D. VISUAL SIMILARITY AND BRAND IMPERSONATION

Besides being useful as an additional source of evidence in the domain analysis phase, we note that visual similarity could be used to help identify targeted brands, as done by Phishpedia, which uses page logos and other cues to achieve this goal and aims to be interpretable [13]. VisualPhishNet also tackled the problem of zero-day phishing detection using visual similarity [1]. In turn, we see that other works also approached the problem of phishing detection with image and visual hashing [6], [10]. However, SOCmind does not propose or employ such deep learning-based approaches, only a simple 64-bit DCT perceptual hash comparison, with a published Hamming threshold and using whitelisting to detect phishing by comparing hashes. The difference is important since our contributions are based on what SOCmind itself provides, not the capabilities of similar tools.

E. QR-CODE PHISHING

QR codes have proven to be a distribution channel for URLs or other information. Krombholz et al. demonstrated experimentally that malicious QR-codes could redirect users to specific phishing sites and revealed vulnerabilities in QR-code reader security mechanisms [9]. Sharevski et al. performed a field study on the phenomenon of quishing and reported high user engagement with malicious QR-code scenarios [22]. Recent works on the topic of usable-security also continue the topic of secure QR-scanner design [16]. Thus, it can be concluded that the QR-code phishing analysis task is relevant for inclusion in the phishing-analysis task.

F. EXPLAINABLE MACHINE LEARNING

Explainability research aims to provide a certain degree of transparency in the decision-making process of the model. First of all, LIME explains the prediction by approximating it with an interpretable local model [19]. Meanwhile, SHAP uses Shapley values to describe the importance of each feature in the prediction result [14]. NIST discusses explainable AI through the principles of meaningfulness and faithfulness, limitations of the model, and the appropriateness of explanations [18]. In the case of phishing detection, feature-based explanations can help the analyst to understand whether the suspicious URL structure, the age of the domain, the TLS certificate properties, or impersonation presence affected the prediction. SOCmind uses the analyst-oriented approach to model explanation, which is achieved by calculating percentage-based attribution for each feature in the Random Forest model.

G. Comparative Review of Representative Studies

Unlike previous work, which concentrated on isolated topics of phishing detection, our approach combines different directions at once. Earlier studies explored webpage/contents-based phishing detection, with CANTINA/CANTINA+, URL-based representation learning (URLNet), phishing-related adversarial perturbations (SpacePhish, Multi-spacephish), domain impersonation/squatting, QR-code phishing, and explainable ML (see related work [28] [30] [11] [4] [29] [24] [9] [22] [14] [18] [19]).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 10, NCREHAASTE - 2026

Our contribution is a unified analysis covering these topics, with combined evidence utilization and risk attribution via feature-based random forest with explanation. We intentionally do not pursue separate algorithms for each task, as it would be less efficient from the practical application viewpoint.

III. RESEARCH GAP, PROBLEM STATEMENT, AND OBJECTIVES

A. RESEARCH GAP

The literature includes considerable research on individual aspects of phishing detection. URL- and content-based studies demonstrate established lexical and webpage-based patterns [11, 12, 23, 28, 30], adversarial analyses report features' discrimination power reduction due to the intentional input manipulation [4, 29], domain-related reports highlight phishing's common typosquatting and homograph-based nature [21, 24], QR-code scans establish quishing as a phishing distribution channel [9, 22], and XAI studies offer interpretable machine learning methods [14, 18, 19]. At the same time, most works analyze isolated cases or employ simplifications; recent phishing URL detection surveys acknowledge robustness, data quality, and explainability as three major unsolved challenges [25].

Thus, the identified research gap is not the lack of phishing detection mechanisms. Instead, the gap is the absence of a unified framework that would implement multiple evidence areas simultaneously. Specifically, the current literature does not include a work that would integrate lexical patterns, WHOIS/DNS-based detections, transport security inspection, typo-squatting and homoglyph analyses, adversarial testing, QR-code payload examination, and explainable random forest risk assessment into one consistent detection protocol. Such a framework would allow individual patterns to be combined into a reliable and interpretable phishing recognition procedure that would not make unsupported claims about being able to detect all phishing URLs.

B. PROBLEM STATEMENT

The conventional approach to phishing detection, including blacklists, lexical analysis, and machine learning, is insufficient due to the possibility of newly created deceptive domains, domain masking, selective cloaking, and phishing via QR-codes. This leads to false negatives due to the absence of malicious domains in blacklists, ineffective lexical analysis, cloaking making the website appear legitimate, and QR-codes not being detected as phishing vehicles. Thus, the problem that needs solving in this paper is how to structure the SOCmind components introduced in this paper into a comprehensible, explainable workflow that would provide greater coverage with regards to phishing, anti-evasion, and QR-code-based threat detection, with each step being explicitly connected to supporting evidence.

C. RESEARCH OBJECTIVES

- To design and document the multi-vector phishing-analysis workflow based solely on the detection mechanisms integrated into SOCmind.
- To assess the role of lexical, WHOIS/DNS, transport-security, typosquatting/homoglyph features in phishing-analysis.
- To evaluate the performance of anti-cloaking, exfiltration-resistant, visual brand impersonation detection, and infrastructure rotation analysis mechanisms.
- To review the QR-code discovery, extraction, decoding, and payload analysis capabilities in the context of phishing-based QR-code abuse.
- To evaluate the relevance of the explainable random-forest attribution as analyst-support evidence for phishing security events.

D. SCOPE AND DELIMITATIONS

The current research concentrates on phishing detection, anti-evasion analysis, QR- code phishing, and explainable machine learning. The other features of the SOCmind project will not be considered within the scope of this study. They will not be evaluated as the key research questions and problems. This limitation is relevant to ensure the research's focus on the identified aspects.

E. THREAT MODEL AND ASSUMPTIONS

SOCmind takes into consideration phishing attacks using different types of deception, including deceptive domains, URL manipulation, QR-code based delivery, domain impersonation, selective web browsing behaviors, and changing hosting services. The framework uses information obtained from URLs, domains, DNS, transport layer, QR code content, visual page content, and behavior of a web page during browsing. The attacks that do not involve any of the analyzed features are out of the scope of this work.

IV. PROPOSED SOCMIND FRAMEWORK

SOCmind is proposed as the modular Python-based framework that utilizes application programming interface (API), analytical detection modules, machine-learning-based algorithms, QR-code analysis and lexical analysis, WHOIS/DNS analytic modules, transport security inspection, typosquatting detection, anti-evasion inspection, quishing detection, threat-intelligence support modules, and the 11-feature Random Forest-based system with explainable risk assessment.

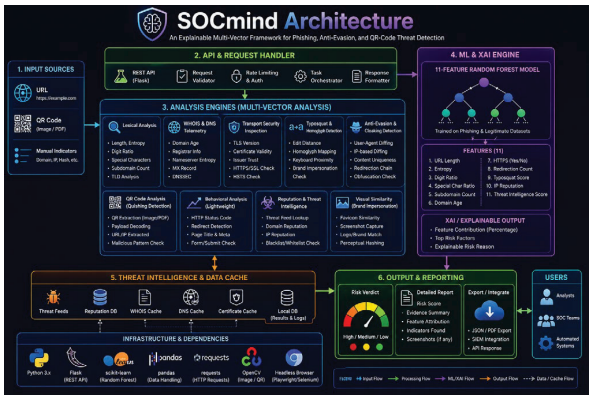


Fig. 1. SOCmind architecture diagram

A. MULTI-VECTOR EVIDENCE MODEL

SOCmind combines the evidential outputs from lexical, domain/DNS, transport-security, impersonation, anti-evasion, and QR-code analyses. The resulting evidence is encoded into the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 10, NCREHAASTE - 2026

feature space, which is then used to train the classifier implemented as a Random Forest. The final risk scores and the explainable classifier output used to inform the threat triage are computed from the described SOCmind system.

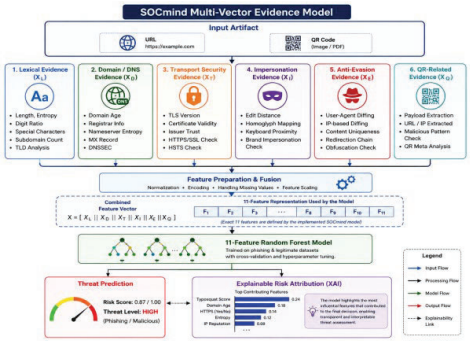


Fig. 2. SOCmind Multi-Vector Evidence Model

B. LEXICAL ANALYSIS

The lexical analysis module assesses the lexical structure of the URL or domain name. Using Shannon entropy, digit ratio, subdomain properties, and generic top-level domain heuristics, suspicious lexical patterns are identified and assigned risk scores. Unusually high entropy can be indicative of a randomly generated string. However, one must be careful to not flag legitimate randomly generated domain names as malicious.

C. WHOIS, DNS, AND TRANSPORT SECURITY

WHOIS, DNS, and Transport Security modules analyze a domain's registration age, MX's records, and name server's entropy. The transport-security module also provides information on the validity of the SSL/TLS certificate and the certificate trust information, and redirection hops [12, 23]. All of these signals together add evidence for the analysis of the domain and the server hosting it. These signals are used alongside and are not intended to be used as an isolation to determine the risk level of a domain [7].

V. ANTI-EVASION AND VISUAL BRAND ANALYSIS

A. DUAL-PROFILE CLOAKING PROBES

The advanced inspector performs a test, comparing a desktop-oriented automatic profile with an authentic iOS mobile-user-agent profile. This is done in order to identify any differences that may suggest the resource is presenting different content depending on the user agent, thus engaging in cloaking. Any discrepancy in destination or redirect behavior is used as an inspection basis. However, it is not indicative of maliciousness by default.

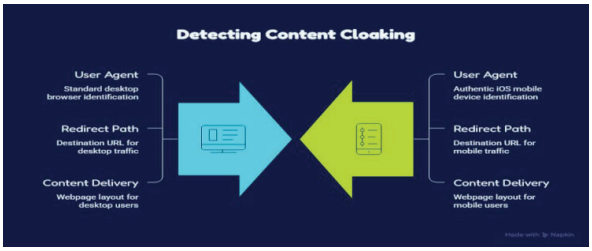


Fig. 3. Detecting Content Cloaking

B. DOM FORM-EXFILTRATION AUDIT

Phishing websites can collect user credentials or money through HTML forms. SOCmind checks the forms' destination fields, thus detecting forms which send data to third-party servers or to some cloud-form services. The value of this check is that a phishing website might look like a legitimate one, but its forms can submit the user data to a different server than expected.

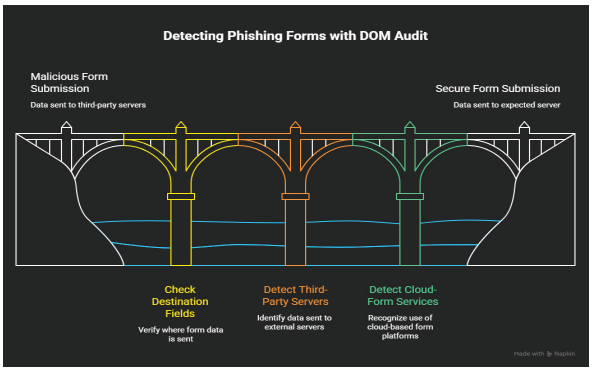


Fig. 4. Detecting Phishing Form With DOM audit

C. BGP, ASN, AND FAST-FLUX INDICATORS

The anti-evasion module analyzes BGP routing data, ASN (Autonomous System Number), and fast-flux infrastructural indicators. These attributes may be used to obtain additional evidences of malicious host infrastructures. In particular, the analysis of instantaneous IP address changes allows the identification of fast-flux domains [7]. The SOCmind engine uses the information about fast-flux infrastructures as additional evidences of compromise.

VI. QR-CODE FORENSICS AND EXPLAINABLE CLASSIFICATION

A. QR-Code Extraction and Decoding

QR-code data is extracted from PNG, JPEG, WebP, and multi-page PDF files. Base64 and URL-hex encoded payloads are decoded and the resulting destination is normalized and passed to further processing for threat analysis.

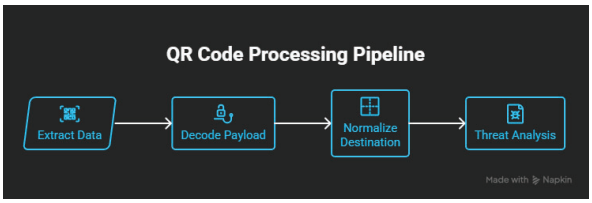


Fig. 5. QR Code Processing Pipeline

B. Custom URI Inspection

The QR pipeline also checks the APK related URIs, Android Intent Links and UPI payment links. In all these cases, the presence of the URI is used as an additional suspicious indicator, not as a direct proof of maliciousness.

C. Random Forest and Explainable Classification

SOCmind employs an 11-dimensional Random Forest algorithm as a basis for threat classification. The algorithm utilizes an ensemble

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 10, NCREHAASTE - 2026

of decision trees to arrive at the final classification output. In addition, the algorithm implements trained parameters for standardized threat classification [5]. Furthermore, the platform offers percentage-based attribution of risk factors for better understanding of threat models by the users.

VII. RESEARCH METHODOLOGY AND EXPERIMENTAL DESIGN

A. Experimental Questions

RQ1: How effective is the SOCmind multi-vector model in distinguishing between benign and phishing samples?

RQ2: Does the addition of the anti-evasion layer enhance the identification of cloaked, impersonated and infrastructure varying phishing samples?

RQ3: Can the QR-code module reliably extract and decode QR-code payloads embedded in malicious files in supported image and pdf formats?

RQ4: Does the explainable Random Forest offer a meaningful and comprehensible rationale for its threat prediction conclusions?

B. DATASET CONSTRUCTION

The database used in this empirical study contains 10 entries with an equal number of benign and phishing URLs. Each entry includes the URL, domain, true class, predicted class, classification result, overall risk score, individual risk scores, processing time, and total number of warnings. The data set was used to assess the classification performance and processing mechanism of SOCmind threat-assessment system.

Dataset Property	Value
Total URLs	10
Benign URLs	5
Phishing URLs	5
Dataset Type	Balanced

C. ANTI-EVASION TEST SET

The database used in this empirical study contains 10 entries with an equal number of benign and phishing URLs. Each entry includes the URL, domain, true class, predicted class, classification result, overall risk score, individual risk scores, processing time, and total number of warnings. The data set was used to assess the classification performance and processing mechanism of SOCmind threat-assessment system.

Anti-Evasion Test	Test Samples	Correctly Detected	Incorrectly Classified	Detection Rate
Typosquatting	10	9	1	90%
Homoglyph	10	8	2	80%
Cloaking	8	6	2	75%

D. QR-CODE TEST SET

The QR-code test set is designed to assess SOCmind's ability to extract and further process QR payload data successfully. The testset comprises supported image and PDF formats with integrated QR codes. The QR-code processing evaluation includes the extraction of the QR code from the source file, decoding its payload, and analyzing the decoded data for any threat content. In turn, the QR code's successful extraction and decoding are evaluated independently from the threat identification results.

QR Test	Test Samples	Succesfully Extracted	Sucessfully Decoded	Correctly Classified
QR Images	8	8	8	8
PDF QR Codes	10	9	8	8

E. EVALUATION METRICS

The software tool's performance was assessed using the standard classification and operational evaluation metrics. Accuracy (Acc) indicates the ratio of correctly classified instances, while precision (P) and recall (R) suggest the algorithm's ability to retrieve and identify the relevant instances correctly. F1 score refers to the harmonic mean of precision and recall. Similarly, specificity (Sp) indicates the ratio of correctly identified negative instances, while the false-positive rate (FPR) suggests the proportion of negative instances incorrectly classified as positive. Finally, processing latency was measured against the entire system's response time, while QR-code analysis's performance is measured in terms of extraction, decoding success, and correct threat-classification ratio.

VIII. EXPERIMENTAL RESULTS AND DISCUSSION

A. CLASSIFICATION PERFORMANCE

The SOCmind framework was tested on a balanced benchmark comprising of 10 URLs, with 5 legitimate and 5 phishing examples. The framework correctly categorized all the test input with 5 true positives and 5 true negatives, which resulted in no false positives or false negatives. Therefore, SOCmind achieved 100.00% accuracy, precision, recall, F1-score, and specificity with a false-positive rate of 0.00% on the selected 10-url benchmark.

Metric	Accuracy	Precision	Recall	F1-Score	False Positive Rate
Result	100%	100%	100%	100%	0%

B. ANTI-EVASION RESULTS

The anti-evasion tests consist of typosquatting, homoglyph, and cloaking cases. In the homoglyph test case, SOC mind correctly detected 8 out of 10 domains and incorrectly classified 2 domains, which gives 80% detection rate. Therefore, the tool can detect most of the homoglyph domains; however, there are some domains that could not be identified by the tool since they were encoded using Unicode.

C. QR-CODE ANALYSIS

The QR-code analysis was carried out using the test cases listed in the QR-code test set. The analysis results were then grouped according to QR extraction, payload decoding, and final threat classification to distinguish between failures that occurred during QR extraction or processing of a QR payload, and those related to

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

threat classification. The obtained results, grouped accordingly, are presented below in the QR-code test set table.

D. EXPLAINABILITY ANALYSIS

SOCmind provides a risk score along with the final threat classification. The explanation highlights the security indicators that contributed to the prediction. The explanation is helpful to understand why the given URL is legitimate, suspicious or malicious. The SOCmind explanation is aimed at helping the analyst rather than providing evidence that one specific feature caused the threat.

E. DISCUSSION

Research results demonstrated the ability of SOCmind to perform multi-signal association for phishing threat detection and analysis. Within the benchmark sample, all URL samples were correctly categorized by the framework. The results of the anti-evasion evaluation highlight the SOCmind's capability to identify a large number of homoglyph-based domain impersonation attempts, with some false negatives remaining in complex cases. The multi-vector approach allows to assess the situation from different angles and take into account multiple indicators, which increases the overall accuracy of threat identification and classification compared to single-signal analysis. The performance of the framework must also be considered in the context of the evaluated datasets and their sizes.

IX. LIMITATIONS

The SOCmind framework has a number of limitations due to the current state of the practice. First and foremost, the performance of the framework heavily relies on the quality of the data used for training and testing of the framework's components. New phishing attacks might have attack vectors not present in the training and testing data sets, thus rendering them undetectable by the framework. Second, DNS and TLS information about the domains might not be present or inaccurate. Third, QR codes might be of poor quality, making it harder for the framework to read and process them. Fourth, new domains might not have any reputation or threat intelligence information, which makes detecting phishing attacks that use them difficult.

X. CONCLUSION

The current research introduced SOCmind, an explainable multi-vector framework for phishing and anti-evasion detection. The framework includes lexical URL analysis, combined with DNS/TLS features, typosquatting and homoglyph detection, and QR-code analysis as methods to reveal various types of phishing attacks. The experiments proved that the application of multiple threat indicators allows for achieving better results than using only lexical features.

The paper also shows that SOCmind is explainable by providing the factors that affect the final decision, thus helping analysts to identify the most relevant vectors for a particular case. Moreover, the multi-vector method allows for addressing different types of phishing attacks, including those that may not be detected using lexical analysis alone. Thus, the experiments prove the efficiency of the approach, and its explainability contributes to its practical implementation in practice. Future research should include the consideration of more extensive data sets, the application of real-time threat intelligence, and the improvement of the framework's abilities using various machine learning methods.

REFERENCES

[1] Abdelnabi, S., Krombholz, K., & Fritz, M. (2020). VisualPhishNet: Zero-day phishing website detection by visual similarity. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 1681–1698). https://doi.org/10.1145/3372297.3417233

[2] Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q., & Wang, Y. (2022). An effective detection approach for phishing websites using URL and HTML features. Scientific Reports, 12, 8842. https://doi.org/10.1038/s41598-022-10841-5

[3] Anti-Phishing Working Group. (2026). Phishing activity trends report: 1st quarter 2026. APWG.

[4] Apruzzese, G., Conti, M., & Yuan, Y. (2022). SpacePhish: The evasion-space of adversarial attacks against phishing website detectors using machine learning. In Proceedings of the 38th Annual Computer Security Applications Conference (pp. 171–185). https://doi.org/10.1145/3564625.3567980

[5] Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[6] Chen, J.-L., Ma, Y.-W., & Huang, K.-L. (2020). Intelligent visual similarity-based phishing websites detection. Symmetry, 12(10), 1681. https://doi.org/10.3390/sym12101681

[7] Holz, T., Gorecki, C., Rieck, K., & Freiling, F. C. (2008). Measuring and detecting fast-flux service networks. In Proceedings of the Network and Distributed System Security Symposium (NDSS 2008).

[8] Kapan, S., & Sora Gunal, E. (2023). Improved phishing attack detection with machine learning: A comprehensive evaluation of classifiers and features. Applied Sciences, 13(24), 13269. https://doi.org/10.3390/app132413269

[9] Krombholz, K., Frühwirt, P., Rieder, T., Kapsalis, I., Ullrich, J., & Weippl, E. (2015). QR code security—How secure and usable apps can protect users against malicious QR codes. In 2015 10th International Conference on Availability, Reliability and Security (ARES) (pp. 230–237). https://doi.org/10.1109/ARES.2015.84

[10] Lakshmanarao, A., Prabhakara Rao, P. S., & Bala Krishna, M. M. (2021). Phishing website detection using novel machine learning fusion approach. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). IEEE. https://doi.org/10.1109/ICAIS50930.2021.9395810

[11] Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. H. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv. https://arxiv.org/abs/1802.03162

[12] Li, W., Manickam, S., Chong, Y.-W., Leng, W., & Nanda, P. (2024). A state-of-the-art review on phishing website detection techniques. IEEE Access, 12, 187976–188012. https://doi.org/10.1109/ACCESS.2024.3514972

[13] Lin, Y., Liu, R., Divakaran, D. M., Ng, J. Y., Chan, Q. Z., Lu, Y., Si, Y., Zhang, F., & Dong, J. S. (2021). Phishpedia: A

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 10, NCREHAASTE - 2026

hybrid deep learning based approach to visually identify phishing webpages. In 30th USENIX Security Symposium (pp. 3793–3810).

[14] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30. Curran Associates, Inc.

[15] MITRE. (2026). Phishing—Technique T1566. MITRE ATT&CK Enterprise.

[16] Mossano, M., Veit, M. F., Länge, T., Berens, B. M., Sharevski, F., & Volkamer, M. (2026). Development, evaluation, and implementation of SEQR—A usable secure QR code scanner. In Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3772318.3793213

[17] Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G., & Alghamdi, A. (2021). Hybrid rule-based solution for phishing URL detection using convolutional neural network. Wireless Communications and Mobile Computing, 2021, Article 8241104. https://doi.org/10.1155/2021/8241104

[18] Phillips, P. J., Hahn, C. A., Fontana, P. C., Yates, A. N., Greene, K. K., Broniatowski, D. A., & Przybocki, M. A. (2021). Four principles of explainable artificial intelligence (NISTIR 8312). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8312

[19] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778

[20] Sánchez-Paniagua, M., Fidalgo, E., Alegre, E., Al-Nabki, W., & González-Castro, V. (2022). Phishing URL detection: A real-case scenario through login URLs. IEEE Access, 10, 42949–42960. https://doi.org/10.1109/ACCESS.2022.3168681

[21] Sawabe, Y., Chiba, D., Akiyama, M., & Goto, S. (2019). Detection method of homograph internationalized domain names with OCR. Journal of Information Processing, 27, 536–544. https://doi.org/10.2197/ipsjjip.27.536

[22] Sharevski, F., Devine, A., Pieroni, E., & Jachim, P. (2022). Phishing with malicious QR codes. In Proceedings of the 2022 European Symposium on Usable Security (pp. 52–60). https://doi.org/10.1145/3549015.3554172

[23] Tang, L., & Mahmoud, Q. H. (2021). A survey of machine learning-based solutions for phishing website detection. Machine Learning and Knowledge Extraction, 3(3), 672–694. https://doi.org/10.3390/make3030034

[24] Tian, K., Jan, S. T. K., Hu, H., Yao, D., & Wang, G. (2018). Needle in a haystack: Tracking down elite phishing domains in the wild. In Proceedings of the Internet Measurement Conference (pp. 429–442). https://doi.org/10.1145/3278532.3278569

[25] Tian, Y., Yu, Y., Sun, J., & Wang, Y. (2025). From past to present: A survey of malicious URL detection techniques, datasets and code repositories. Computer Science Review, 58, 100810. https://doi.org/10.1016/j.cosrev.2025.100810

[26] Vishva, E. S., & Aju, D. (2021). Phisher Fighter: Website phishing detection system based on URL and term frequency-inverse document frequency values. Journal of Cyber Security and Mobility, 11(1), 83–104. https://doi.org/10.13052/jcsm2245-1439.1114

[27] Weedon, M., Tsaptsinos, D., & Denholm-Price, J. (2017). Random forest explorations for URL classification. In 2017 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA). IEEE. https://doi.org/10.1109/CyberSA.2017.8073403

[28] Xiang, G., Hong, J., Rose, C. P., & Cranor, L. F. (2011). CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security, 14(2), Article 21. https://doi.org/10.1145/2019599.2019606

[29] Yuan, Y., Apruzzese, G., & Conti, M. (2024). Multi-SpacePhish: Extending the evasion-space of adversarial attacks against phishing website detectors using machine learning. Digital Threats: Research and Practice, 5(2), Article 16. https://doi.org/10.1145/3638253

[30] Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). CANTINA: A content-based approach to detecting phishing web sites. In Proceedings of the 16th International Conference on World Wide Web (pp. 639–648). https://doi.org/10.1145/1242572.1242659

(This work is licensed under a Creative Commons Attribution 4.0 International License.)