

A Comparative Study of QR-Code and URL Features for Machine Learning-Based Phishing Detection

Samiksha Sumesh Kunnathully¹, Yash Dashrath Pandit², Soham Sunil Pawar³, Ms. Swati Kadu⁴

¹²³⁴Department of MCA, D. Y. Patil University, School of Humanities and Sciences, Navi Mumbai, India

ABSTRACT - QR-code phishing, often called quishing, uses QR codes to conceal links that lead users toward deceptive or malicious destinations. This study presents a comparative machine-learning analysis of QR structural information and decoded URL characteristics for identifying potentially phishing QR codes before navigation. A public dataset of 9,987 QR samples was used, including 5,005 benign and 4,982 phishing samples. The experiments considered 21 QR features and 38 lexical or structural URL features. Three representations QR-only, URL-only, and QR + URL were tested with Logistic Regression and Random Forest. In five-fold stratified cross-validation, the URL-only Random Forest gave the strongest average results, reaching 91.65% accuracy, 91.45% F1-score, and 97.16% ROC-AUC. The corresponding mean accuracy for QR-only was 76.35%, while the combined representation reached 90.59%. SHAP analysis highlighted suspicious keyword count, URL length, subdomain count, and domain-label count as influential variables; QR structural variables also affected individual predictions. The findings indicate that lightweight, static, and interpretable analysis can support QR-code phishing detection, while the evaluated dataset did not show an aggregate benefit from feature combination.

Key Words: QR-code phishing, quishing, URL features, machine learning, random forest, explainable AI, SHAP

I. INTRODUCTION

Phishing remains a significant cybersecurity problem because attackers manipulate users into disclosing sensitive information or visiting harmful resources. The widespread use of QR codes for payments, authentication, information sharing, and web access has created another route through which such deceptive destinations can be delivered.

QR-based phishing, commonly described as quishing, introduces an additional challenge because the destination address is hidden inside the visual code. A person can scan the code and follow the resulting link without first seeing or evaluating the URL in the same way they might with an ordinary hyperlink.

Trad and Chehab examined whether properties of QR structure and pixel patterns could be used directly by machine-learning models for quishing detection. Their results provide evidence that the QR representation itself can contain information useful for distinguishing suspicious codes [1].

The two feature groups describe different stages of the QR-to-web process. QR features characterize the encoded

visual structure, whereas URL features describe properties of the decoded destination string. It is therefore useful to test them separately as well as together instead of assuming in advance that combining them will improve classification.

Accordingly, this study evaluates QR-only, URL-only, and combined QR + URL representations with lightweight classifiers. The decoded URL is treated only as text during analysis; the destination itself is never visited or contacted.

1.1 Problem Statement

The central problem is to determine whether QR structural characteristics, decoded URL properties, or both together can support accurate and interpretable identification of potentially phishing QR codes before a user navigates to the destination.

1.2 Objectives

1. Investigate lightweight machine learning techniques for QR-code phishing detection.
2. Extract structural features from QR-code representations.
3. Extract lexical and structural features from decoded URLs.
4. Compare QR-only, URL-only, and combined QR + URL feature sets.
5. Evaluate Logistic Regression and Random Forest classifiers.
6. Analyze important features using Random Forest feature importance and SHAP.
7. Assess computational requirements.

1.3 Research Questions

RQ1: How effectively can lightweight machine learning models detect QR-code phishing using QR and URL features?

RQ2: Does combining QR and URL features provide additional detection capability compared with either feature group independently?

RQ3: Which QR and URL features contribute most strongly to phishing predictions?

II. RELATED WORK

Research on QR-code phishing has expanded alongside the use of QR codes as convenient gateways to online content. Existing approaches can focus on the QR representation itself or examine characteristics of the destination encoded by the QR code.

Trad and Chehab presented a machine-learning method based on QR structure and pixel-level information without relying on extraction of the embedded web content [1]. Their study supports the practical possibility of detecting phishing from QR-centric information alone.

URL-oriented detection is also relevant because deceptive links frequently contain measurable lexical and structural patterns. Examples include unusual URL length, multiple subdomains, numerical or special characters, and terms associated with authentication, accounts, payments, or security.

For cybersecurity applications, model accuracy alone may not be sufficient; analysts also need to understand why a model reaches a decision. SHAP, introduced by Lundberg and Lee, provides a systematic way to attribute model outputs to input features [2].

2.1 Research Gap

The research gap considered here is the absence, within this experimental setup, of a single lightweight comparison in which QR structural features, decoded URL features, and their combined representation are evaluated under the same data split and models while also examining feature-level explanations. The analysis deliberately tests feature fusion rather than presuming that it is beneficial.

III. PROPOSED FRAMEWORK

QR Dataset → QR Reconstruction → QR Decoding → QR Feature Extraction + URL Feature Extraction → Feature Configuration → Preprocessing → Machine Learning → Evaluation → Explainability

The QR representation is reconstructed from the supplied binary matrix and decoded locally. The resulting URL is never opened, browsed, or sent to its destination. Three feature configurations are then compared: QR-only with 21 features, URL-only with 38 features, and QR + URL with 59 features.

IV. RESEARCH METHODOLOGY

4.1 Dataset

A publicly available QR-code dataset associated with the work of Trad and Chehab [1] was used. The dataset

contains 9,987 samples, with each QR code represented as a binary matrix of size 69×69 .

Table 1. Dataset distribution

Class	Samples	Percentage
Benign	5,005	50.12%
Phishing	4,982	49.88%
Total	9,987	100%

4.2 QR-Code Reconstruction and Decoding

For every sample, the binary matrix was transformed into a black-and-white image and decoded on the local system. The decoding procedure succeeded for all 9,987 samples. No decoded URL was visited, queried, or otherwise contacted during this process.

4.3 QR-Code Feature Extraction

The QR representation was summarized through 21 structural variables. These cover properties such as code size and inferred version, black-to-white proportions, entropy, horizontal and vertical transition behavior, row and column density statistics, central density, and symmetry-related measures.

4.4 URL Feature Extraction

The decoded URLs were described with 38 lexical and structural variables. The feature set includes overall URL length, hostname and path properties, character frequencies, subdomain information, protocol indicators, IP and port indicators, Punycode, suspicious and brand-related keyword counts, URL-shortener indicators, entropy measures, top-level-domain length, and the number of domain labels.

4.5 Feature-Set Construction

Table 2. Feature configurations

Feature Set	Features
QR-only	21
URL-only	38
QR + URL	59

4.6 Data Preprocessing

The data were divided using an 80:20 stratified split with random seed 42, giving 7,989 samples for training and 1,998 for testing. Median imputation was applied where necessary. Inputs for Logistic Regression were

standardized, whereas Random Forest operated without feature scaling.

4.7 Machine Learning Models

Logistic Regression served as the baseline classifier and was allowed up to 2,000 iterations. The Random Forest configuration contained 300 trees, used random state 42, applied balanced class weights, and enabled parallel processing.

4.8 Training and Testing Procedure

For each of the three feature configurations, the classifiers were trained independently. The fixed 80:20 split supplied the main hold-out evaluation, while Random Forest was additionally assessed using five-fold stratified cross-validation to examine performance consistency across folds.

4.9 Evaluation Metrics

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

ROC-AUC was included to assess how well the classifier separates benign and phishing samples over different decision thresholds rather than at only one operating point.

4.10 Explainability Using SHAP

For model interpretation, SHAP was calculated for 300 randomly selected observations from the test set. Global feature rankings were obtained from the mean absolute SHAP values, which indicate the average magnitude of each feature's contribution to model output [2].

4.11 Computational Efficiency

In the reported experimental environment, the hybrid Random Forest required approximately 2.8 seconds for training, occupied about 42.69 MB after serialization, and processed 1,000 predictions in approximately 0.056 seconds. These figures describe the observed setup and are not intended as hardware-independent benchmarks.

V. RESULTS AND DISCUSSION

5.1 Classification Performance

Table 3. Classification results on the 80:20 test set

Feature Set	Model	Accuracy	Precision	Recall	F1	ROC-AUC
QR-only	Logistic Regression	72.82%	77.02%	64.89%	70.44%	78.04%
QR-only	Random Forest	76.93%	82.21%	68.61%	74.79%	84.06%
URL-only	Logistic Regression	87.39%	90.10%	83.95%	86.92%	95.03%
URL-only	Random Forest	91.09%	92.97%	88.87%	90.87%	97.08%
QR + URL	Logistic Regression	87.79%	90.35%	84.55%	87.36%	95.13%
QR + URL	Random Forest	90.74%	92.92%	88.16%	90.48%	96.29%

On the fixed test set, the URL-only Random Forest was the strongest configuration. The QR + URL model performed similarly, but its score remained below the URL-only model rather than providing an improvement.

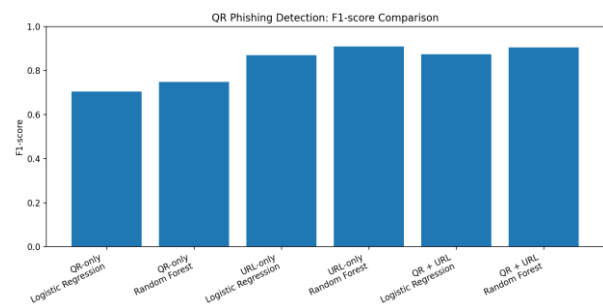


Fig. 1. F1-score comparison of the evaluated models.

5.2 Five-Fold Cross-Validation

Table 4. Five-fold stratified cross-validation results for Random Forest

Feature Set	Accuracy Mean $\pm$ SD	F1 Mean $\pm$ SD	ROC-AUC Mean $\pm$ SD	Precision	Recall
QR-only	76.35% $\pm$ 0.82%	73.75% $\pm$ 1.07%	83.19% $\pm$ 1.03%	82.64%	66.62%

Feature Set	Accuracy Mean $\pm$ SD	F1 Mean $\pm$ SD	ROC-AUC Mean $\pm$ SD	Precision	Recall
URL-only	91.65% $\pm$ 0.55%	91.45% $\pm$ 0.58%	97.16% $\pm$ 0.16%	93.46%	89.52%
QR + URL	90.59% $\pm$ 0.68%	90.32% $\pm$ 0.71%	96.39% $\pm$ 0.20%	92.77%	88.00%

The fold-to-fold standard deviations were relatively small, suggesting stable performance under the five stratified splits. Among the Random Forest configurations, URL-only again produced the highest average results.

5.3 QR and URL Feature Comparison

The QR-only Random Forest reached 76.35% mean accuracy, confirming that structural information in the QR representation contains useful predictive signal. The URL-only model was substantially stronger, with 91.65% mean accuracy and a 91.45% mean F1-score, while the combined representation reached 90.59% mean accuracy. One plausible explanation is that the QR samples are derived from URL-associated content, allowing some information captured by QR structure to overlap with information represented explicitly by the decoded URL.

5.4 Confusion Matrix

Table 5. Confusion matrix of the QR + URL Random Forest

Actual / Predicted	Benign	Phishing
Benign	934	67
Phishing	118	879

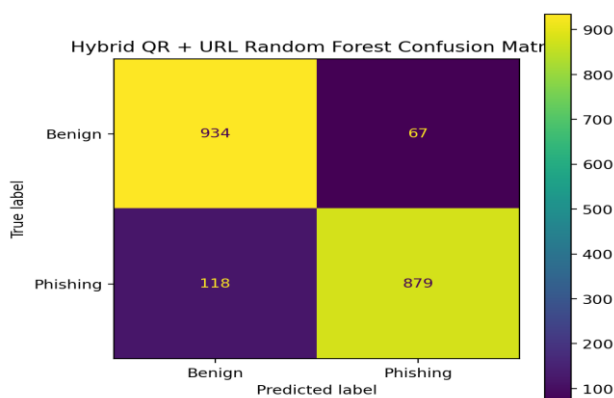


Fig. 2. Confusion matrix of the QR + URL Random Forest model.

For the QR + URL Random Forest, 934 benign samples and 879 phishing samples were classified correctly. The remaining errors consisted of 67 benign samples predicted as phishing and 118 phishing samples predicted as benign.

5.5 Feature Importance

Table 6. Top Random Forest feature-importance rankings

Rank	Feature
1	suspicious_keyword_count
2	url_length
3	num_letters
4	num_subdomains
5	qr_vertical_transition_std
6	domain_label_count
7	url_entropy
8	path_entropy
9	path_length
10	num_digits

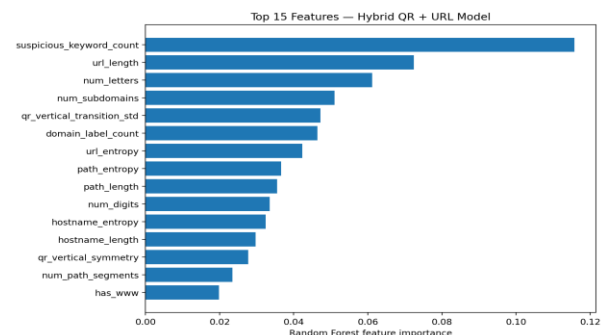


Fig. 3. Top Random Forest feature importance for the QR + URL model.

The largest Random Forest importance was assigned to suspicious_keyword_count. Most of the highest-ranked variables describe URL properties, although qr_vertical_transition_std also appeared within the top-ranked features, indicating that QR structure still supplied useful information to the combined model.

5.6 SHAP Analysis

Table 7. Top SHAP feature contributions

Rank	Feature	Mean Absolute SHAP
1	suspicious_keyword_count	0.0875
2	url_length	0.0483

Rank	Feature	Mean Absolute SHAP
3	num_subdomains	0.0428
4	domain_label_count	0.0390
5	num_letters	0.0327
6	num_digits	0.0241
7	hostname_length	0.0231
8	has_www	0.0228
9	tld_length	0.0228
10	qr_vertical_transition_std	0.0206

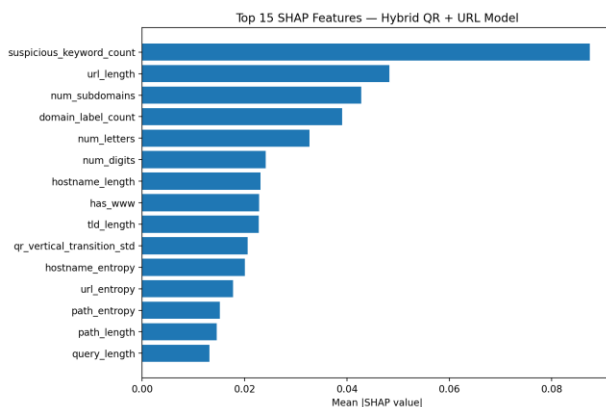


Fig. 4. SHAP-based global feature importance for the QR + URL Random Forest model.

The SHAP ranking was broadly consistent with the Random Forest importance results. In particular, URL-related variables were prominent, while QR structural variables also contributed to individual model decisions. Thus, the absence of an aggregate performance gain for QR + URL does not mean that every QR feature was irrelevant to prediction.

5.7 Computational Performance

Table 8. Measured computational performance

Measure	Observed Value
Hybrid Random Forest training time	Approximately 2.8 seconds
Serialized model size	Approximately 42.69 MB
Prediction time for 1,000 samples	Approximately 0.056 seconds

5.8 Discussion

- The QR representation alone contains predictive information that can support phishing classification.

- For the evaluated dataset, URL-based variables produced substantially stronger aggregate performance than QR-only variables.
- Adding the QR feature group to the URL features did not increase the overall Random Forest scores.
- Feature-importance measures and SHAP add decision-level interpretability that is not visible from accuracy and related metrics alone.

5.9 Limitations

1. The evaluation relied on a single primary QR dataset.
2. The samples were supplied as binary QR matrices rather than photographs captured from real environments.
3. Because the QR representations are associated with URL information, the two feature groups may contain overlapping signals that influence the comparison.
4. The proposed analysis is static and does not examine the content or behavior of the destination webpage.
5. The experimental comparison was limited to the selected lightweight classifiers.
6. No independent external dataset was used to validate generalization beyond the primary dataset.

VI. CONCLUSION, FUTURE WORK, AND FINAL CONTRIBUTIONS

6.1 Conclusion

The study evaluated a lightweight and interpretable approach to pre-navigation detection of potentially phishing QR codes by comparing structural QR information with characteristics of decoded URLs.

Across the five-fold Random Forest evaluation, the URL-only configuration delivered the best average performance, with 91.65% accuracy, 91.45% F1-score, and 97.16% ROC-AUC. QR-only obtained 76.35% mean accuracy, whereas the QR + URL representation obtained 90.59%.

These results do not support a claim that combining the two feature groups improves performance on the evaluated dataset. Nevertheless, the analysis shows that QR structural variables can still influence predictions and therefore should not be considered useless solely because the combined model has a lower aggregate score.

Both Random Forest importance and SHAP highlighted suspicious keyword count, URL length, subdomain count, and domain-label count as influential variables, with

additional contribution from selected QR structural features.

Taken together, the experiments show that a relatively lightweight static pipeline can provide useful and explainable QR-code phishing classification, while also demonstrating why the choice of feature representation should be validated empirically rather than assumed to benefit from fusion.

6.2 Future Work

1. Test the approach on independent QR-code datasets from different sources and distributions.
2. Extend evaluation from binary QR matrices to photographs and scanned QR images captured under realistic conditions.
3. Explore additional structural, visual, and image-quality characteristics of QR codes.
4. Apply feature-selection methods to determine whether a smaller representation can retain comparable performance.
5. Benchmark the current classifiers against other lightweight machine-learning methods.
6. Expand SHAP analysis with both global rankings and case-level explanations for individual predictions.
7. Prototype and evaluate the detection pipeline inside a real-time QR-scanning workflow.
8. Assess robustness under distribution shifts and against changing phishing strategies.

6.3 Final Contributions

1. A controlled comparison of QR-only, URL-only, and QR + URL feature representations.
2. Experimental evaluation of Logistic Regression and Random Forest as lightweight classifiers.
3. Five-fold stratified cross-validation of Random Forest for the three feature configurations.
4. Model interpretation using Random Forest feature importance together with SHAP-based attribution.
5. Measurement of training time, serialized model size, and prediction time for the evaluated pipeline.
6. Evidence from the evaluated dataset that URL features produced stronger aggregate performance than the combined QR + URL representation.

ACKNOWLEDGMENT

The authors sincerely thank the Department of MCA, D Y Patil University School of Humanities and Sciences, for the academic support, facilities, and research environment provided during this work.

REFERENCES

- [1] F. Trad and A. Chehab, "Detecting Quishing Attacks with Machine Learning Techniques Through QR Code Analysis," arXiv preprint arXiv:2505.03451, 2025.
- [2] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems* 30, 2017.
- [3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [4] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [5] P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [6] W. McKinney, *Python for Data Analysis*, 3rd ed., O'Reilly Media, 2022.