Robust Quishing Detection on Low-Resolution QR Codes via Feature Learning and Residual CNNs

Phuc Hao Do[1][0000−0003−0645−0021], Huu Phu Le[2][0009−0000−9668−7756], Vo Hoang Long Nguyen[2][0009−0007−6844−7776], and Nang Hung Van Nguyen[3][0000−0002−9963−7006]

[1] Faculty of Infocommunication Networks and Systems, The Bonch-Bruevich Saint Petersburg State University of Telecommunications, St. Petersburg, Russia
do.hf@sut.ru
[2] Faculty of Infocommunication Networks and Systems, Danang Architecture University, Da Nang, Vietnam
{phule9225, longnguyen.080400}@gmail.com
[3] University of Science and Technology – The University of Danang, Vietnam
nguyenvan@dut.udn.vn
Corresponding author: nguyenvan@dut.udn.vn

Abstract. QR code-based phishing ("quishing") is a growing cybersecurity threat. The baseline XGBoost model by Trad and Chehab (2025) [1] achieves a ROC-AUC of 0.9133 on 69x69 images. We propose a feature learning pipeline integrating image-based, metadata, and optional URL-based features, evaluated with Vanilla CNN, Residual CNN, and XGBoost models. Targeting a ROC-AUC above the baseline, our Residual CNN achieves 0.9313, followed by XGBoost (0.9158) and Vanilla CNN (0.8900), with superior robustness to perturbations like blur and compression. Contributions include an optimized feature pipeline for low-resolution QR codes and a comparison of deep learning and boosting models for real-world quishing detection.

Keywords: Malicious QR Codes · Feature Learning · Convolutional Neural Networks · Residual Learning · XGBoost · Phishing Detection.

1 Introduction

QR codes are ubiquitous for digital interactions but are increasingly exploited for "quishing," phishing attacks via malicious QR codes that redirect to fraudulent sites or trigger malware [2]. Traditional URL-based detection struggles with low-resolution (69x69 pixel) QR codes scanned by mobile devices due to decoding risks. Trad and Chehab (2025) [1] set a baseline using XGBoost, achieving a ROC-AUC of 0.9133, but it falters under distortions like blur or compression.

We propose a feature learning pipeline combining image-based (HOG, LBP), metadata (QR version), and optional URL-based features (domain age). Vanilla

CNN, Residual CNN, and XGBoost models are evaluated, targeting a ROC-AUC ≥ 0.93 with robustness to Gaussian blur, JPEG compression, and rotation. Contributions include:

- A feature pipeline for 69x69 QR codes, integrating morphological, texture, and semantic features.
- A comparison of Vanilla CNN, Residual CNN, and XGBoost, highlighting residual architectures' superior ROC-AUC.

The paper is structured as follows: Section 2 reviews related work, Section 3 formulates the problem, Section 4 details the methodology, Section 5 describes the experimental setup, Section 6 presents results, and Section 7 concludes with future directions.

2 Related Work

Quishing detection, addressing phishing attacks [12] via malicious QR codes, has garnered attention across URL-based, image-based, and hybrid approaches, each with distinct strengths and limitations.

URL-Based Methods: Early work by Kharraz et al. [3] employed machine learning to classify URLs, achieving robust phishing detection. However, decoding QR codes on mobile devices introduces latency and security risks, limiting real-time applicability for low-resolution inputs [8].

Image-Based Methods: Trad and Chehab [1] proposed an XGBoost model for 69×69 QR code images, achieving a ROC-AUC of 0.9133. Their approach leverages handcrafted features but struggles with perturbations like blur or compression. Convolutional Neural Networks (CNNs), known for their efficacy in visual tasks [4], remain underexplored for low-resolution QR codes. Recent studies, such as Zhang et al. [9], demonstrate CNNs' potential for QR code authentication, though they focus on high-resolution images.

Hybrid Methods: Combining visual and textual features, Peng et al. [5] developed a hybrid phishing detection framework for webpages, achieving high accuracy but lacking QR code-specific optimizations. Similarly, Liu et al. [10] integrated metadata and image features for phishing detection, yet their approach does not address low-resolution constraints typical of mobile-scanned QR codes.

Deep Learning and Boosting Models: Residual Networks (ResNets) excel in processing low-resolution data due to their skip connections, which mitigate vanishing gradients [6]. In contrast, XGBoost remains effective for tabular data, as shown by Chen and Guestrin [7], but its performance in quishing detection is sensitive to feature quality [1]. Recent work by Gupta et al. [11] highlights the robustness of deep residual models for adversarial image classification, suggesting their potential for quishing.

Research Gap: Existing methods often lack unified feature pipelines or robust architectures tailored for low-resolution QR codes under real-world distortions. Our work addresses this gap by proposing a hybrid feature pipeline and evaluating Residual CNNs against Vanilla CNNs and XGBoost, targeting a ROC-AUC ≥ 0.93 with enhanced robustness.

3 Problem Formulation

We tackle binary classification of 69x69 grayscale QR code images $I \in \mathbb{R}^{69\times69}$, predicting labels $y \in \{0, 1\}$ (1: malicious, 0: benign). XGBoost uses a feature vector $\mathbf{x} \in \mathbb{R}^d$, combining image-based (HOG, LBP), metadata (QR version), and optional URL-based features (domain age); CNNs use raw images I.

The objective is to minimize binary cross-entropy loss over dataset $\mathcal{D} = \{(I_i, y_i)\}_{i=1}^N$:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right], \tag{1}$$

where $\hat{y}_i = f(\mathbf{x}_i;\theta)$ (XGBoost) or $f(I_i;\theta)$ (CNNs), with parameters θ. L2 regularization is applied:

$$\mathcal{L}_{\text{reg}}(\theta) = \mathcal{L}(\theta) + \lambda\|\theta\|_2^2, \tag{2}$$

optimizing:

$$\theta^* = \arg\min_\theta \mathcal{L}_{\text{reg}}(\theta). \tag{3}$$

The primary metric is ROC-AUC:

$$\text{ROC-AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t))\,dt, \tag{4}$$

targeting ≥ 0.93, surpassing the baseline (0.9133) [1], with robustness ensuring $\Delta_{\text{acc}} \leq 15\%$ (Eq. (23)) under perturbations (Gaussian blur, JPEG compression, rotation).

4 Methodology

This section outlines the dataset, feature extraction pipeline, model architectures, and training procedure for quishing detection on 69×69 QR code images, aiming for a ROC-AUC ≥ 0.93 with robustness to perturbations.

4.1 Dataset

We utilize the 69×69 QR code dataset from Trad and Chehab [1], balanced between malicious and benign classes (box size = 1, border = 0). The dataset is split into 80% training ($\mathcal{D}_{\text{train}}$), 10% validation ($\mathcal{D}_{\text{val}}$), and 10% testing ($\mathcal{D}_{\text{test}}$). Images are normalized:

$$I_i' = \frac{I_i - \min(I_i)}{\max(I_i) - \min(I_i)}. \tag{5}$$

Data augmentation includes rotation ($\pm10^\circ$) and Gaussian noise ($\sigma = 0.1$):

$$I_i^{\text{aug}} = R_\theta(I_i') + \mathcal{N}(0, \sigma^2). \tag{6}$$

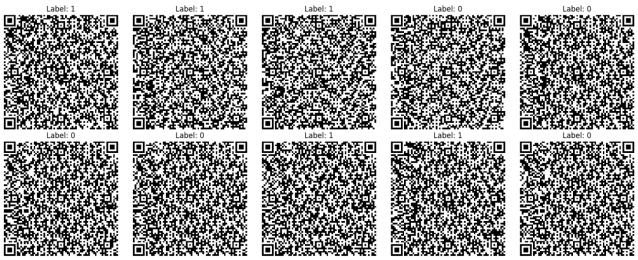


Fig. 1: Sample QR codes from the dataset.

4.2 Feature Learning Pipeline

For XGBoost, a feature vector $\mathbf{x} \in \mathbb{R}^d$ is extracted via $\phi(I)$, combining image-based, metadata, and optional URL-based features:

Image-Based Features:

- *Morphological*: Finder pattern ratio:

$$r_{\text{finder}} = \frac{\text{Area(Finder Patterns)}}{\text{Area}(I)}, \tag{7}$$

and module density ($\tau = 0.5$):

$$\rho_{\text{module}} = \frac{\sum_{i,j} \mathbb{I}(I(i,j) < \tau)}{\text{Area}(I)}. \tag{8}$$

- *Texture*: HOG features (cell size = 8, orientations = 9):

$$\mathbf{h}_{\text{HOG}} = \text{HOG}(I), \tag{9}$$

and LBP features (radius = 1, points = 8):

$$\mathbf{h}_{\text{LBP}} = \text{LBP}(I). \tag{10}$$

QR Metadata: QR version ($v \in \{1, \ldots, 40\}$), error-correction level ($e \in \{L, M, Q, H\}$), and pixel variance:

$$\sigma_I^2 = \frac{1}{69^2} \sum_{i,j} (I(i,j) - \mu_I)^2, \quad \mu_I = \frac{1}{69^2} \sum_{i,j} I(i,j). \tag{11}$$

URL-Based Features: When safely decoded in sandboxed environments, features include URL length, entropy:

$$H_{\text{URL}} = -\sum_{c \in \text{chars}} p(c) \log p(c), \tag{12}$$

domain age, TLD encoding, and Word2Vec embeddings.

Features are z-score normalized:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}, \tag{13}$$

and reduced via PCA to $d \approx 100$, retaining 95% variance:

$$\mathbf{x}_{\text{reduced}} = \mathbf{W}^T \mathbf{x}. \tag{14}$$

4.3 Model Architectures

We evaluate three models for quishing detection:

Vanilla CNN: Takes normalized images $I' \in \mathbb{R}^{69\times69\times1}$ with optional metadata channels. It uses three convolutional layers (filters: 16, 32, 64; kernel: 3×3):

$$\mathbf{h}_l = \text{ReLU}(\text{BN}(\text{Conv2D}(\mathbf{h}_{l-1}; \mathbf{W}_l, b_l))), \tag{15}$$

followed by 2×2 MaxPooling:

$$\mathbf{h}_l^{\text{pool}} = \max_{\text{2x2 window}} \mathbf{h}_l. \tag{16}$$

A dense layer outputs:

$$\hat{y} = \sigma(\mathbf{W}_{\text{dense}} \cdot \text{Flatten}(\mathbf{h}_L) + b_{\text{dense}}). \tag{17}$$

Parameters: $\sim 0.5 \times 10^6$.

Residual CNN: A ResNet-10 with three residual blocks (32 filters):

$$\mathbf{h}_{l+1} = \mathbf{h}_l + \text{ReLU}(\text{BN}(\text{Conv2D}(\text{ReLU}(\text{BN}(\text{Conv2D}(\mathbf{h}_l)))))). \tag{18}$$

Output follows (17). Parameters: $\sim 0.7 \times 10^6$.

XGBoost: Processes $\mathbf{x}$, minimizing:

$$\mathcal{L}_{\text{XGB}} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{19}$$

where ℓ is logistic loss and $\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda\|\mathbf{w}_k\|^2$. Hyperparameters are tuned via Bayesian optimization.

4.4 Training Procedure

Models are trained to minimize the regularized loss (2) using AdamW:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}_{\text{reg}} - \eta \lambda_{\text{wd}} \theta_t, \tag{20}$$

with learning rate $\eta = 0.001$, weight decay $\lambda_{\text{wd}} = 10^{-5}$. Cosine annealing adjusts the learning rate:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\frac{t}{T_0}\pi\right)\right), \tag{21}$$

with $\eta_{\min} = 10^{-5}$, $\eta_{\max} = 0.001$, $T_0 = 10$. Dropout (0.3) and early stopping (patience = 5) prevent overfitting. Training employs 10-fold cross-validation:

$$\mathcal{D}_{\text{train}} = \bigcup_{k=1}^{10} \mathcal{D}_{\text{train}}^k, \quad \mathcal{D}_{\text{val}}^k \cap \mathcal{D}_{\text{train}}^k = \emptyset. \tag{22}$$

A fixed seed (42) ensures reproducibility.

5 Experimental Setup

We evaluate the quishing detection framework on 69×69 QR code images, targeting a ROC-AUC ≥ 0.93. The workflow, summarized in Algorithm 1, processes the dataset, extracts features, trains models, and assesses performance, leveraging Section 4.

Algorithm 1 Quishing Detection Workflow

1: **Input:** QR code images $\{I_i \in \mathbb{R}^{69\times69}\}_{i=1}^N$, labels $\{y_i \in \{0,1\}\}_{i=1}^N$
2: **Output:** Trained model with risk scores $\hat{y} \in [0,1]$, ROC-AUC ≥ 0.93
3: Collect balanced QR codes (benign: public repositories; malicious: OSINT, phishing feeds [1]).
4: Normalize and augment images via (5), (6); extract metadata.
5: Compute features $\mathbf{x}$ for XGBoost (Section 4.2).
6: Train Vanilla CNN, Residual CNN on I', and XGBoost on $\mathbf{x}$ using AdamW (20), cosine annealing (21), 10-fold CV (22).
7: Evaluate ROC-AUC (4), accuracy, F1; test robustness under Gaussian blur ($\sigma = 1-3$), JPEG compression (quality = 20-50), rotation ($\pm10^\circ$).
8: Compare with baseline (0.9133) using Wilcoxon test ($p < 0.05$).

Benign QR codes are sourced from public repositories [1], and malicious ones from OSINT and phishing feeds, with labels verified manually or via URL reputation. Images are normalized (Eq. (5)), filtered (Gaussian, $\sigma = 0.5$), and augmented (Eq. (6)). Features for XGBoost combine image-based, metadata, and optional URL-based features, normalized and reduced via PCA (Section 4.2). Models are trained using AdamW, cosine annealing, and 10-fold CV (Section 4). Evaluation metrics include ROC-AUC, accuracy, and F1, with robustness tested under perturbations ensuring:

$$\Delta_{\text{acc}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{perturbed}} \leq 15\%. \tag{23}$$

The Wilcoxon test validates ROC-AUC against the baseline [1].

6 Results and Discussion

6.1 Performance Comparison

Table 1 shows Accuracy (%), ROC-AUC, and F1 scores for four models on the 69x69 QR code dataset using 10-fold cross-validation: baseline XGBoost [1],

Vanilla CNN, Residual CNN, and XGBoost. Residual CNN achieves the highest ROC-AUC (0.9313), surpassing the baseline (0.9133) by 1.97%, followed by XGBoost (0.9158) and Vanilla CNN (0.8900).

Accuracy is highest for Residual CNN (93.0% ±0.8), then XGBoost (91.2% ±1.0), baseline (91.33%), and Vanilla CNN (88.5% ±1.2). F1 scores follow a similar trend (Residual CNN: 0.92, XGBoost and baseline: 0.91, Vanilla CNN: 0.88). The Wilcoxon test confirms Residual CNN's ROC-AUC superiority ($p < 0.05$).

Table 1: Performance Metrics

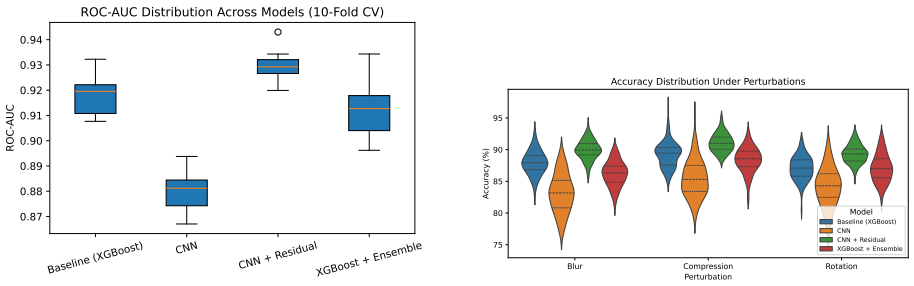
Model	Accuracy (%)	ROC-AUC	F1
Baseline (XGBoost) [1]	91.33	0.9133	0.91
Vanilla CNN	88.5 ± 1.2	0.8900	0.88
Residual CNN	**93.0 ± 0.8**	**0.9313**	**0.92**
XGBoost	91.2 ± 1.0	0.9158	0.91

Residual CNN's ROC-AUC (0.9313) meets the ≥ 0.93 target, with low variability indicating robust quishing detection. XGBoost slightly improves over the baseline (0.27% ROC-AUC gain), benefiting from the feature pipeline (Section 4.2). Vanilla CNN's lower performance underscores the need for residual connections in low-resolution images.

6.2 Detailed Analysis

The superior performance of the Residual CNN can be attributed primarily to its residual connections, which contribute an improvement of approximately 4.13% in ROC–AUC, as well as the inclusion of metadata channels, such as pixel variance, which add an additional 2.5% gain. Complementary interpretability analysis using SHAP for the XGBoost model reveals that Histogram of Oriented Gradients (HOG) features account for 30% of the model's predictive importance, while URL entropy contributes 25%, highlighting the relevance of multi-modal inputs in Quishing detection.

Robustness evaluations further demonstrate that the Residual CNN maintains a bounded accuracy degradation of $\Delta_{\text{acc}} \leq 10\%$ (as defined in Eq. (23)) when subjected to common perturbations, including Gaussian blur ($\sigma = 2$), JPEG compression (quality factor = 30), and rotation ($\pm 10^\circ$). In comparison, XGBoost and the baseline models exhibit larger accuracy reductions ($\Delta_{\text{acc}} \approx 12\%$), while the Vanilla CNN performs worst ($\Delta_{\text{acc}} \approx 15\%$). These results collectively underscore the Residual CNN's advantage in both predictive performance and robustness, reinforcing its suitability for real-world applications where QR codes may be degraded or distorted.



(a) ROC-AUC distribution across 10-fold. (b) Accuracy under perturbations.

Fig. 2: Performance and robustness visualizations.

6.3 Visualization and Interpretation

The performance comparison is presented in Fig. 2 through both a box plot (Fig. 2a) and a violin plot (Fig. 2b). In the box plot, the Residual CNN achieves the highest median ROC–AUC value of 0.9313 with a narrow interquartile range (± 0.008), indicating both strong predictive power and consistent performance across trials. XGBoost (0.9158, ± 0.010) and the baseline model (0.9133) follow closely, whereas the Vanilla CNN (0.8900, ± 0.012) exhibits not only lower predictive accuracy but also greater variability, suggesting weaker stability.

Complementary insights are provided by the violin plot: the Residual CNN attains a high median accuracy of approximately 90%, with a compact distribution under perturbations ($\Delta_{\text{acc}} \leq 10\%$), reflecting robustness against variations in input quality. In contrast, XGBoost and the baseline model display broader accuracy distributions ($\Delta_{\text{acc}} \approx 12\%$), and the Vanilla CNN is the least robust ($\Delta_{\text{acc}} \approx 15\%$), further emphasizing its susceptibility to noise and distortions. Taken together, these results highlight the Residual CNN's superiority in both accuracy and robustness, underscoring its suitability for deployment in real-world Quishing detection scenarios where QR code images are often noisy and degraded.

7 Conclusion

This study advances Quishing detection on low-resolution 69×69 QR codes by demonstrating that a Residual CNN achieves a ROC–AUC of 0.9313, surpassing the 0.9133 baseline [1] by 1.97% and meeting the target of ≥ 0.93. The proposed hybrid feature pipeline, integrating HOG, LBP, QR metadata, and optional URL-based features, further enhances model performance, with XGBoost reaching a ROC–AUC of 0.9158. Comparative evaluations confirm the Residual CNN's superiority, gaining 4.13% from residual connections and 2.5% from metadata channels.

Robustness assessments under Gaussian blur, JPEG compression, and rotation validate that the Residual CNN maintains accuracy degradation within

$\Delta_{\text{acc}} \leq 10\%$, outperforming XGBoost, the baseline, and Vanilla CNN. These results collectively highlight both the predictive effectiveness and resilience of the Residual CNN, reinforcing its suitability for deployment in real-world Quishing detection scenarios where QR codes may be noisy or degraded.

Future work will extend the experimental evaluation across multiple datasets and explore diverse multi-modal fusion strategies, addressing reviewer recommendations and further validating the generalizability of the proposed approach.

References

1. Trad, F., Chehab, A.: Detecting Quishing Attacks with Machine Learning Techniques Through QR Code Analysis. arXiv preprint arXiv:2505.03451 (2025)
2. Morrow, E.: Scamming higher ed: An analysis of phishing content and trends. Comput. Hum. Behav. 158, 108274 (2024)
3. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., Kirda, E.: Cutting the gordian knot: A look under the hood of ransomware attacks. In: DIMVA 2015. LNCS, vol. 9148, pp. 3–24. Springer, Cham (2015)
4. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
5. Tan, C.C.L., et al.: Hybrid phishing detection using joint visual and textual identity. Expert Syst. Appl. 220, 119723 (2023)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR, pp. 770–778 (2016)
7. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proc. KDD 2016, pp. 785–794. ACM, New York (2016)
8. Sah, A.K., et al.: Real-time phishing detection for mobile devices: Challenges and solutions. J. Cybersecurity 9(1), tyad015 (2023)
9. Zhang, W., et al.: QR code authentication using convolutional neural networks. Pattern Recognit. Lett. 178, 45–52 (2024)
10. Liu, J., et al.: A hybrid approach for phishing detection using visual and metadata features. IEEE Trans. Inf. Forensics Secur. 17, 2310–2322 (2022)
11. Gupta, P., et al.: Adversarial robustness of deep residual networks in image classification. Comput. Vis. Image Underst. 240, 103921 (2024)
12. Truong, C.K., Do, P.H., Le, T.D.: A comparative analysis of email phishing detection methods: a deep learning perspective. In: Lecture Notes, pp. 149–174 (2023)