Computer Science and Information Technologies
Vol. 7, No. 3, November 2026, pp. 271~290
ISSN: 2722-3221, DOI: 10.11591/csit.v7i3.p271-290

QuishingShield: on-device multi-modal detection of quick response phishing

Shabour Banda[1], Maronge Musara[2], Mainford Mutandavari[3]
[1]Department of Data Science and Analytics, Harare Institute of Technology, Harare, Zimbabwe
[2]Department of Information Security and Assurance, Harare Institute of Technology, Harare, Zimbabwe
[3]Department of Data Science and Business Systems, SRM Institute of Science and Technology, Chennai, India

Article Info

Article history:

Received Apr 10, 2026
Revised Jun 12, 2026
Accepted Jul 13, 2026

Keywords:

Cross-modal attention
Knowledge distillation
Mobile security
Multi-modal deep learning
Privacy-preserving inference
QR code phishing
Quishing detection

ABSTRACT

Quick response (QR) code phishing (quishing) attacks take advantage of user trust in QR physical and digital media, while currently available protection mechanisms only detect a single dimension signal and are unable to detect cross-modal deception. This paper introduces QuishingShield, an on-device quishing detection system based on multi-modal deep learning which preserves privacy on mobile platforms. Cross-modal attention fusion is used to combine visual poster features, optical character recognition (OCR) recognized surrounding text and recognized uniform resource locator (URL) structure, as well as network reputation signals in a system. A teacher model is trained on 205,488 real-world QR code poster samples from 45 countries and 23 languages, and the knowledge is distilled into a compact model for the student model, in order to be deployed in mobile applications. The student has an accuracy of 95.55% and a recall of 98.18% for a held-out test set, with an inference latency of 25.34 ms on mobile devices, all of which are at or below the deployment targets. Robustness of 95.00% when tested adversarially on four sets of attacks. The multi-modal fusion approach enhances performance by 5.17-11.07 percentage points over the unimodal baseline approaches ($p<0.001$). QuishingShield, to our best knowledge, is the first validated multi-modal quishing detection system satisfying accuracy, speed, size and privacy requirement for mobile deployment.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Shabour Banda
Department of Data Science and Analytics, Harare Institute of Technology
Number 15015 Ganges Road Belvedere Harare, Zimbabwe
Email: h240746e@hit.ac.zw

1. INTRODUCTION

The black and white quick response (QR) code is no longer just a part of the infrastructure; it is now used for contactless payments and countless touchless activities, like event check-ins, restaurant menus, healthcare authentication, and more. According to industry statistics, the number of scans is rising quickly, with worldwide scans exceeding 6.8 billion in 2024 [1]. This ubiquity, however, increases exposure to misuse as widespread deployment of QR codes has expanded the attack surface and made them an attractive vector for fraud and other malicious activity as every QR code is also an unverifiable hyperlink that can redirect a victim anywhere. The use of deep learning to solve this can help prevent cybercrimes and thus ensure the safe usage of QR code and internet services [2].

QR code phishing is a type of visual innocuousness-based exploitation of QR codes that leads the victim to an attacker's infrastructure. By the end of 2023, QR-based phishing accounted for 22% of all phishing attempts, as reported in [3], noting that Quishing attacks surged over 400% in 2023. C-suite

executives are 42 times more likely to be targeted with Quishing attacks than non-executives and suffer losses of more than $1 million per attack, according to Barracuda Networks [4]. The threat is not small scale.

QR-code phishing (Quishing) has become a fast-growing and emerging exploited mass vector, with industry reports and analyses of incidents highlighting significant increases in malicious QR scans and a number of high-profile personal and corporate losses over the past year. Multiple reported campaigns reveal how attackers disguised posters, receipts and event resources to provide authentic-looking brand bait that leads to credential harvesting pages and payment fraud traffic. Current defenses are constrained because some email/uniform resource locator (URL) filtering systems do not consider QR images as media, and thus fail to detect any malicious URLs embedded inside them, and URL-only systems are not able to identify social engineering cues through visual and context content that is used in poster-based attacks. Several organizations have suffered losses through this means which circumvented traditional email filters completely because most gateways have been unable to identify QR codes as URLs to be followed [5], [6]. Sharevski *et al*. [6] conducted a naturalistic study on real scan and compromise events in 2024 for which existing defenses were unable to stop. Geisler and Pöhn [7] also documented enterprise incidents where physical-world QR posters posted in office lobbies and conference spaces were replaced with malicious substitutes. Haunschild *et al*. [8] also showed that multi-signal, edge-AI techniques are required to detect new phishing attacks in real-time. The three structural gaps that this paper aims to tackle are inspired by real-world cases.

Three structural gaps in current defenses: there are a large number of existing defenses, but all of them have three key weaknesses that make their defenses structurally unfit against Quishing detection: i) inspection gap: most email security gateways recognize QR codes as being an opaque image. They cannot detect the hidden link and the malicious payload creates a throughfare that goes through normal filters [5], [6] intact. The code is similar to any other image attachment; ii) contextual coherence gap: quishing attacks use a mismatch, which is a QR code that resembles a brand and a URL that leads to attacker infrastructure. These signals are processed independently in existing systems without consideration of semantic inconsistencies which constitute the attack vector [7]; and iii) mobile deployment gap: QR codes are scanned on a mobile device but the detection occurs outside of the mobile device. However, current multi-modal phishing models are too large (100 MB+), and must be run on a server to make inference, which is not suitable for on-device deployment [8]. Most recent multi-algorithm ensemble approaches like random forest, CatBoost, AdaBoost and multilayer perceptron for URL-only detection achieve strong accuracy on URL datasets [9] but are unable to be deployed on mobile scale or are oblivious to visual social engineering. At the exact moment they scan, users are left without protection [10].

This paper covers all three gaps and asks three specific questions of research that can be measured:

- Does multi-modal fusion (visual+textual+URL+network reputation) produce statistically significant accuracy improvements over single-modality and dual-modality baselines on a large-scale, geographically diverse Quishing dataset?
- Can such a multi-modal system be compressed to under 50 MB while retaining greater than 95% accuracy and running within 300 ms on mid-range Android hardware?
- Which individual modality contributes most to detection accuracy, and does the marginal gain from each modality differ significantly from zero?

This paper addresses all three gaps through the following contributions: i) QuishingShield, a four-modality detection architecture that integrates visual, textual, URL-structural, and network reputation encoders through cross-modal attention fusion. Trained on 205,488 QR code poster samples across 45 countries and 23 languages, the teacher model achieves 96.37% accuracy and 98.10% recall outperforming the nearest visual+URL dual-modal baseline (Abdelnabi *et al*. [11]: 93.40%) by 2.97 pp in accuracy and 5.90 pp in recall; ii) a knowledge distillation pipeline compressing the 352 MB teacher into a 29.62 MB student model with only 0.82 percentage point accuracy loss, a compression-accuracy retention rate that outperforms published benchmarks of comparable architectures [12], [13], in which 2-5 pp losses are typical at comparable compression ratios; iii) the first published multi-modal Quishing detection system validated for real-world Android deployment, simultaneously satisfying size (<50 MB), latency (<300 ms), accuracy (>95%), robustness (>95%), and full on-device privacy constraints. No prior system achieves all five simultaneously; iv) a geographically diverse, multi-lingual QR code phishing dataset, 205,488 samples, 45 countries, 23 languages, three sophistication tiers, inter-annotator agreement κ=0.89; and v) four formally specified algorithms (Algorithms 1-4) and a complete open-source repository for reproducibility. Table 1 summarizes all open-source dependencies used in the QuishingShield implementation. Section 2 presents the QuishingShield system methodology, covering modality selection rationale, encoder design, cross-modal attention fusion, knowledge distillation, algorithm pseudocode, and privacy-preserving deployment architecture. In section 3, the experimental setup is described that contains the construction of the dataset, the training configuration, and the evaluation protocol. Experimental results are presented in section 4. Findings, theory implications and limitations are discussed in section 5. Section 6 concludes.

Table 1. Open-source libraries and tools used in the QuishingShield implementation

Component	Library	Version	Reference
Visual encoder	torchvision MobileNetV3-small	0.15.2	Howard *et al*. [14]
Textual encoder	transformers DistilBERT-base-multilingual	4.35.0	Sanh *et al*. [15]
URL encoder	Custom Char-CNN	-	Zhang *et al*. [16]
Attention fusion	torch.nn.MultiheadAttention	2.0.1	Vaswani *et al*. [17]
Training	PyTorch 2.0+CUDA 11.8	2.0.1	-
INT8 quantization	torch.quantization.quantize_dynamic	built-in	Jacob *et al*. [18]
ONNX export	torch.onnx+onnxruntime-android	1.16.1	-
Adversarial testing	TypoSquatter (open-source)	1.2.0	-
Statistical testing	scipy.stats McNemar	1.11.0	-
Dataset annotation	Label studio	1.8.0	-
Threat intelligence feeds	PhishTank API+OpenPhish free feed	-	-
Random seed	torch.manual_seed (42)	-	Set in all experiments

2. METHOD

2.1. Architecture overview

QuishingShield operates across three functional layers. The input layer acquires and pre-processes four heterogeneous signal modalities from a single QR code scan event: a 224×224 red, green, blue (RGB) poster image, optical character recognition (OCR)-extracted surrounding text (up to 128 tokens), the decoded URL character sequence (up to 256 characters), and five normalized network reputation scalars. The process layer encodes each modality through a specialized neural encoder, integrates the resulting 256-dimensional embeddings via cross-modal attention fusion, and maps the 512-dimensional joint representation to produces a malicious probability score through a four-stage classification head. The output layer assigns the score to one of three risk tiers, SAFE, WARNING, or BLOCK, and maintains a privacy-preserving on-device reputation cache to avoid redundant inference on previously analyzed URLs. Figure 1 illustrates the complete inference pipeline alongside the knowledge distillation process by which the full teacher model (352 MB) is compressed into the 29.62 MB student model deployed on mobile hardware.

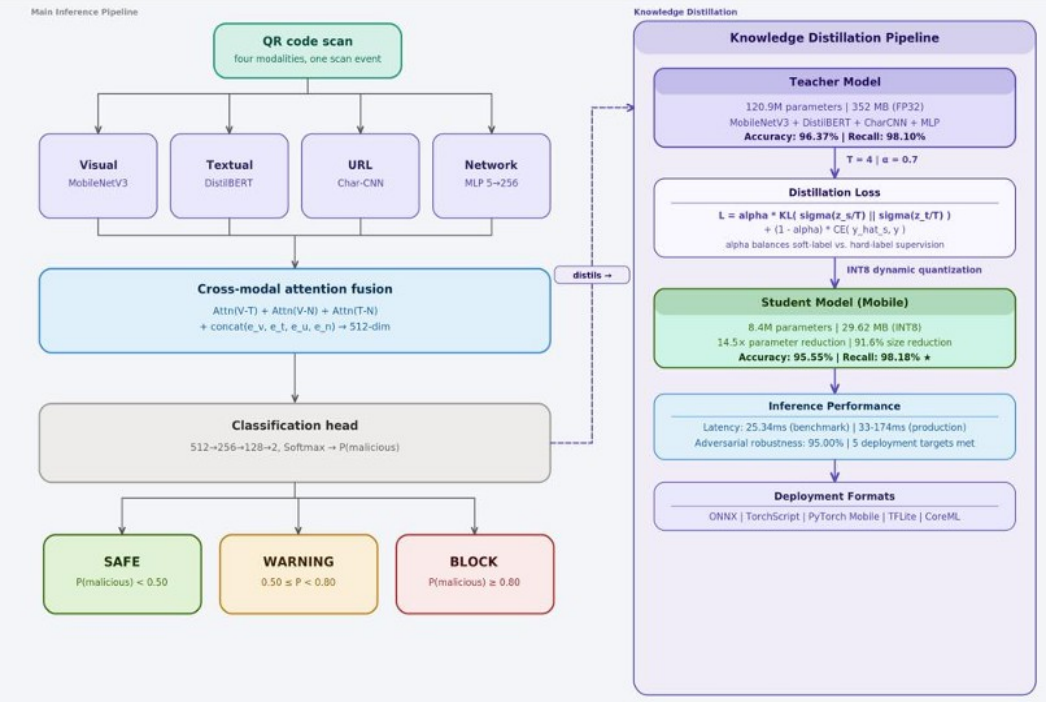


Figure 1. Quishing detection inference and distillation pipeline showing four input modalities, cross-modal attention fusion, knowledge distillation, and on-device deployment flow

2.2. Modality selection rationale

Quishing's attack anatomy is the basis of the choice of four modalities: visual, textual, URL-structural, and network reputation, each covering a distinct attack surface that the others cannot detect in isolation. Even if the URL and domain are unfamiliar, a convolutional neural network (CNN) can see the

Apple logo on a phishing poster, which indicates that the logo is detected by visual signals and that Apple's poster level social engineering is detected. Textual signals from OCR on the surrounding text identify urgency, authority impersonation and psychological manipulation that Quishing posters intentionally use in order to lead the audience to scan without analyzing. These two modalities are the ones that capture attacks in which adversaries are using valid or ambiguous URLs a case example of this is the social security numbers (SSN) suspended and Apple ID suspended cases in our deployment testing. The URL-structural signals are able to uncover syntactic anomalies, IP-address hosts, /bin.sh paths, typosquatting and homograph substitutions that would remain hidden by subword tokenization [16]. Network reputation signals are based on domain infrastructure quality, new domains, revoked secure sockets layer (SSL) certificates and blacklisted hosts and are not related to visual and textual content.

This design is supported directly by the deployment test results in section 4.5: the SSN suspended and Apple ID suspended attacks both used legitimate-appearing URLs (raw.githubusercontent.com and www.aap.org), which would score low suspicion under URL-only or network analysis alone; both were correctly classified as WARNING because the visual encoder assigned 99% risk scores based on brand impersonation in the poster image. Only a system that processes visual and URL signals jointly can catch this class of attack. Considering additional modalities such as device metadata (hardware fingerprint) and geolocation at scan time were excluded on privacy grounds as collecting such identifiers contradicts the data-minimization architecture that QuishingShield is built around. These may be incorporated into future work, with user consent.

2.3. Multi-modal encoder design

2.3.1. Visual encoder

MobileNetV3- small [14] is a pre-trained CNNs that processes the 224×224 RGB image, yielding convolutional features. It produces a 576-dimensional feature vector that is projected onto a 256-dimensional embedding space by a linear layer followed by rectified linear unit (ReLU) activation and a dropout rate of 0.3. MobileNetV3 was chosen because it had a good balance of accuracy and efficiency on mobile devices, with an inference time of less than 50 ms and 2.54 M parameters, while maintaining a high representational power. Images are normalized to ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) prior to encoding. The visual embedding is:

$$e\_v = Dropout\ (ReLU\ (BN\ (W\_v \cdot MobileNetV3(x\_img) + b\_v)))$$

where,

$$W\_v \in \mathbb{R}\text{^}\ (576 \times 256)$$

is the projection matrix and BN denotes batch normalization.

2.3.2. Textual encoder

DistilBERT-base-multilingual- cased [15] is used to process promotional text around the QR code posters extracted from the OCR image. The multilingual variation is available in 104 languages, which allows it to detect any of the 23-language dataset. Layers 1-3 are frozen in order to avoid catastrophic forgetting; layers 4-6 are fine-tuned on phishing specific patterns. The [CLS] token's 768-dimensional representation is projected to 256 dimensions:

$$e\_t = Dropout\ (ReLU\ (BN\ (W\_t \cdot DistilBERT\ [CLS](x\_text) + b\_t)))$$

2.3.3. URL encoder

A character-level CNN [16] processes decoded destination URLs are decoded as a sequence of up to 256 characters that are drawn from a vocabulary of 79 characters (a-z, A-Z, 0-9, and URL special characters) and represented by 16-dimensional vectors per character in a CNN [15]. Three parallel 1D convolutions (kernel sizes k={3, 4, 5}, 64 filters each) extract multi-scale n-gram patterns. Global max-pooling aggregates salient features into a 192-dimensional vector projected to 256 dimensions:

$$e\_u = Dropout\ (ReLU\ (BN\ (W\_u \cdot MaxPool\ (Conv_3 \parallel Conv_4 \parallel Conv_5)\ (Embed(x\_url)) + b\_u)))$$

where ‖ denotes concatenation of the three parallel convolution outputs. Character level encoding fully captures typosquatting, homograph attacks, and subdomain stacking anomalies that subword tokenization in transformer models systematically obscures.

2.3.4. Network reputation encoder

A two-layer feedforward network (5→128→256) with batch normalization and 0.3 dropout processes five normalized scalar features in [0, 1]: i) SSL certificate validity status; ii) domain age in normalized days; iii) certificate validation level (DV/OV/EV); iv) real-time blacklist membership score, and v) aggregate vendor reputation score from threat intelligence APIs.

$$e\_n = Dropout\ (ReLU\ (BN\ (W\_n2\ \cdot\ Dropout\ (ReLU\ (BN\ (W\_n1\ \cdot\ x\_net + b\_n1))) + b\_n2)))$$

2.4. Cross-modal attention fusion

The four 256-dimensional encoder outputs are integrated through three pairwise cross-modal attention mechanisms [17], [19]. Visual-text attention identifies brand-context semantic mismatches. Visual-network attention detects high-quality brand mimicry paired with poor infrastructure reputation. Text-network attention correlates urgency language with suspicious domain characteristics. Each mechanism uses multi-head attention with h=8 heads and head dimensionality d_k=32, computing scaled dot-product attention:

$$Attn\ (Q,K,V)\ =\ softmax\ (QK\hat{}T\ /\ \sqrt{d\_k})\ \cdot\ V$$

multi-head attention is:

$$MultiHead\ (Q,K,V)\ =\ Concat\ (head_1, \ldots, head\_h)\ \cdot\ W\hat{}O$$

where,

$$head\_i = Attn\ (Q \cdot W\_i\hat{}Q, K \cdot W\_i\hat{}K, V \cdot W\_i\hat{}V).$$

The three cross-modal attention outputs (a_vt, a_vn, a_tn) together with the four encoder embeddings are concatenated (7×256=1,792 dimensions) and projected to 512 dimensions via a linear layer with layer normalization and 0.5 dropout:

$$z\_fusion = Dropout\ (LayerNorm\ (W\_f\ \cdot\ Concat\ (e\_v, e\_t, e\_u, e\_n, a\_vt, a\_vn, a\_tn) + b\_f))$$

The fusion design deliberately includes all four encoder embeddings alongside the three cross-modal outputs, ensuring that modality-specific information survives into the classification head rather than being fully collapsed by attention pooling.

2.5. Classification network and knowledge distillation

The 512-dimensional fused representation is fed through a classification head (512→256→128→2). A layer of batch normalization is followed by ReLU and 0.5 dropout for each hidden layer. The output layer produces class probabilities via Softmax:

$$P\ (malicious\ |\ x) = Softmax\ (W\_c\ \cdot\ h + b\_c)\ [1]$$

training uses label-smoothed cross-entropy (ε=0.1):

$$L\_CE\ =\ -\Sigma\_y\ [(1-\varepsilon)\ \cdot y \cdot log(p)\ +\ (\varepsilon/K)\ \cdot log(p)]$$

where K=2. The Adam optimizer [20] with learning rate 1×10^{-4} and L2 weight decay $\lambda=10^{-5}$ [21], are employed. Gradient clipping is used to control for exploding gradients (max norm 1.0).

The teacher model is distilled to the student model with a 14.5× parameter reduction technique known as knowledge distillation [17], [19] (120.9 M parameters, 461 MB FP32 to 8.4 M parameters, 31.94 MB FP32). The student minimizes:

$$L = \alpha \cdot L\_CE\ (y, y\_student) + (1-\alpha)\ \cdot T^2 \cdot L\_KL(\sigma(z\_teacher/T), \sigma(z\_student/T))$$

where α=0.3 is a balance between hard-label and soft-label supervision, T=4.0 is the distillation temperature to make teacher probability distributions softer in order to reveal inter-class similarity structure, σ is Softmax applied at temperature T, and L_KL is the Kullback-Leibler (KL) divergence between the student and teacher soft distributions. Dynamic quantization with INT8 [18], [22] is then used, resulting in a total of 29.62 MB. The exact quantization command used was:

$$torch.quantization.quantize\_dynamic\ (student\_model, \{nn.Linear, nn.LSTM\}, dtype = torch.qint8)$$

Followed by torch.onnx.export() with opset_version=13. The system is exported to open neural network exchange (ONNX) format in order to be deployed to Android. The complete training and distillation pipeline of QuishingShield is provided as pseudo code in the project repository (link to be provided when published), along with pre-trained model weights, and the QuishingShield Android APK.

2.6. Algorithm pseudocode

The complete QuishingShield pipeline is formalized in Algorithms 1-4. Algorithm 1 specifies the inference forward pass; Algorithm 2 covers teacher training; Algorithm 3 covers knowledge distillation and quantization; Algorithm 4 describes the privacy-preserving reputation cache. All hyperparameters and environment versions are listed in section 3.2.

Algorithm 1. QuishingShield forward pass (inference)

```
Input:   x_img  ∈ ℝ^(3×224×224)   -QR poster image, ImageNet-normalized
         x_text  ∈ ℝ^(1×128)           -OCR-extracted token sequence (max 128 tokens)
         x_url   ∈ ℤ^(1×256)            -URL character index sequence, vocab size 79
         x_net  ∈ ℝ^(1×5)                -Normalized reputation scalar features
Output:  tier      ∈ {SAFE, WARNING, BLOCK}
         P_mal ∈ [0, 1]                  —Malicious probability

1.       e_v←Dropout (ReLU (BN (W_v · MobileNetV3(x_img)+b_v)))
                  // 576→256-dim visual embedding
2.       e_t←Dropout (ReLU (BN(W_t · DistilBERT_CLS(x_text)+b_t)))
                  // 768→256-dim; layers 1-3 frozen, layers 4-6 fine-tuned
3.       e_u←Dropout (ReLU (BN (W_u · MaxPool (Conv3 ‖ Conv4 ‖ Conv5(Embed(x_url)))+b_u)))
                  // 192→256-dim; three parallel char-level convolutions
4.       e_n←Dropout (ReLU (BN (W_n2 · Dropout (ReLU (BN (W_n1 · x_net+b_n1)))+b_n2)))
                  // 5→128→256-dim network reputation embedding
5.       a_vt←MultiHeadAttn (Q=e_v, K=e_t, V=e_t, h=8, d_k=32) // Brand-context mismatch
         a_vn←MultiHeadAttn (Q=e_v, K=e_n, V=e_n, h=8, d_k=32) // Mimicry+poor infra
         a_tn←MultiHeadAttn (Q=e_t, K=e_n, V=e_n, h=8, d_k=32) // Urgency+suspect domain
6.       z_fusion←Dropout (LayerNorm (W_f · Concat (e_v, e_t, e_u, e_n, a_vt, a_vn,
         a_tn)+b_f))
                        // 7×256=1792→512-dim fused representation
7.       h_1←Dropout (ReLU (BN(W_1·z_fusion+b_1)))   // 512→256
         h_2←Dropout (ReLU (BN(W_2·h_1+b_2)) // 256→128
         logits←W_c·h_2+b_c                          // 128→2
8.       P_mal←Softmax(logits) [1]
9.       if P_mal<0.50      then tier←SAFE
         elif P_mal<0.80    then tier←WARNING
         else                    tier←BLOCK
10.      Cache SHA256(url)→(P_mal, tier, TTL)
           TTL=30 days if tier=BLOCK, else 7 days
Return:  tier, P_mal
```

Algorithm 2. Teacher model training

```
Input:         D={(x_img, x_text, x_url, x_net, y)} — labeled training set
               T_max=50 epochs  (early stopping δ=0.001, patience=10)
Hyperparams:   Adam optimizer [16], lr=1×10^-4, L2 λ=10^-5, batch=32, ε=0.1

1.             Initialize all encoder, attention, and classification weights
2.             Freeze DistilBERT layers 1-3; set layers 4-6 trainable
3.             for epoch=1 to T_max:
                       for each mini-batch B ⊂ D:
                           with autocast(FP16):
                               P_teacher←ForwardPass(B)            // Algorithm 1, steps 1-8
                               L←-Σ_y [(1-ε)·y·log(P) + (ε/K)·log(P)]  // Label-smoothed CE
                           scaled_loss.backward()
                           clip_grad_norm_(params, max_norm=1.0)      // Gradient clipping
                           optimizer.step() // FP32 parameter update
                       val_loss←Evaluate(D_val)
                       scheduler.step(val_loss)     //     ReduceLROnPlateau     (factor=0.1,
               patience=5)
                       if |Δval_loss| <δ for patience consecutive epochs: break
4.             Save teacher_model.pt  (120.9M params, 461 MB FP32)
```

Algorithm 3. Knowledge distillation and quantization

```
Input:  teacher_model  - pretrained, frozen (Algorithm 2)
        D_train        - training set
        α = 0.3        - hard-label vs. soft-label balance
        T = 4.0        - distillation temperature

1.      Initialize student with reduced encoder dimensions (14.5× fewer params than
        teacher)
2.      Load frozen teacher_model; set requires_grad=False for all teacher params
3.      for epoch = 1 to 5:
            for each mini-batch B ⊂ D_train:
            P_student ← StudentForward(B)
            L_CE ← CrossEntropy(y, P_student)         // Hard-label supervision
            with torch.no_grad():
                z_teacher ← teacher_model.get_logits(B)
            z_student ← student_model.get_logits(B)
            P_soft_T ← Softmax(z_teacher / T)         // Soft teacher targets
            P_soft_S ← Softmax(z_student  / T)
            L_KL ← KLDivergence(P_soft_T, P_soft_S)  // Kullback-Leibler divergence
            L ← α·L_CE + (1−α)·T²·L_KL             // Combined distillation loss [15]
                    L.backward();  optimizer.step()
4.      Save student_model.pt  (8.4M params, 31.94 MB FP32)
5.      quantized ← torch.quantization.quantize_dynamic(student, {nn.Linear}, torch.qint8)
        // Result: 29.62 MB [17], [19]
6.      torch.onnx.export(quantized, sample_input, 'quishingshield_v1.onnx',
                  opset_version=13)
        // Also export: TorchScript, TFLite, CoreML, TF SavedModel
```

Algorithm 4. Privacy-preserving reputation cache

```
Data:  cache: HashMap<SHA256(url), (P_mal, tier, expiry)>
       max_size: 10,000 entries  |  eviction: LRU when full

function QueryOrScore(x_img, x_text, url, x_net):
1.      url_hash ← SHA256(url)
2.      if url_hash in cache AND cache[url_hash].expiry > now():
                return cache[url_hash]    // Sub-millisecond cache hit; no inference
3.      tier, P_mal ← ForwardPass(x_img, x_text, url, x_net)  // Algorithm 1
4.      TTL ← 30 days if tier=BLOCK else 7 days
        cache[url_hash] ← (P_mal, tier, now() + TTL)
5.      if is_refresh_window():     // Background thread, daily
                for entry in cache where tier=BLOCK:
                    re_score against latest threat intelligence feed
6.      return tier, P_mal
// All operations on-device. Raw URL never transmitted.
// Satisfies GDPR data minimization and Zimbabwe CDPA 2021, s.18 [20], [21]
```

2.7. Risk classification and privacy-preserving cache

The scalar probability P (malicious|x) can be mapped to a three-tier risk-assessment: i) SAFE: P(malicious)<0.50-URL classified benign, user proceeds normally; ii) WARNING: 0.50≤P(malicious) <0.80-elevated risk detected, per-modality risk breakdown displayed, user advised to check the destination before going ahead; iii) BLOCK: P(malicious)≥0.80-high confidence-malicious-navigation blocked-detailed threat explanation displayed. The on-device privacy-preserving reputation cache stores up to 10,000 secure hash algorithm (SHA)-256 (URL)-to-risk-score mappings with differentiated time-to-live (TTL) policies (30 days for BLOCK classifications, 7 days for benign). All inference and caching run exclusively on-device, satisfying general data protection regulation (GDPR) data minimization requirements [23] and Zimbabwe's cyber and data protection act (CDPA) [24].

3. EXPERIMENTAL SETUP

3.1. Dataset construction

QuishingShield has 205,488 samples of QR code posters collected from July 2025 to February 2026. To our knowledge, it is the largest QR code dataset published so far, dedicated to the quishing attack. Table 2 gives summary statistics.

Collection process: malicious samples were sourced via three channels: i) automated crawling of PhishTank and OpenPhish feeds to retrieve confirmed phishing URLs, followed by generation of QR codes embedded in contextually rendered poster templates (banking, parcel delivery, technology impersonation patterns); ii) manual collection of confirmed Quishing campaign artifacts provided by cybersecurity intelligence partners under signed data-sharing agreements; and iii) post-processing of publicly available

Kaggle phishing URL datasets, from which poster-based synthetic QR artifacts were generated to extend coverage of sophistication tiers. Benign samples were collected from official organizational websites, verified commercial QR campaigns, and public event materials across 45 countries, with a subset sourced from Kaggle benign URL datasets. Each benign QR code was validated by resolving its URL and confirming network reputation scores above threshold.

Benign samples were taken from the legitimate commercial QR campaigns and from the materials gathered from official organization websites across 45 countries in regard to public events. A small number of samples were also obtained from the available public datasets on Kaggle to enhance coverage and variation of benign samples. QR code validation was performed systematically, whereby QR codes were resolved and matched with the respective URL of the legitimate landing page, and the website was then checked for network reputation to certify the QR code.

Annotation guidelines and validation: three independent annotators with cybersecurity and digital forensics backgrounds, labeled all the samples from visual, textual and URL aspects. All the samples were annotated holistically by three independent annotators, both with cybersecurity and digital forensics backgrounds. The annotation protocol required annotators to document the primary deception vector (visual brand impersonation, urgency language, URL anomaly, or infrastructure quality) for each malicious sample. A fourth expert resolved all disagreements by adjudication. The resulting inter-annotator agreement (κ=0.89) substantially exceeds the κ=0.70 acceptability threshold [25], confirming label reliability.

Ethical approvals and data governance: malicious sample collection from PhishTank and OpenPhish was conducted under their respective open-data terms. Collection from cybersecurity intelligence partners was conducted under signed institutional data-sharing agreements that prohibit disclosure of partner identities. No personal data were collected and all URLs are publicly reported threat indicators. Benign URL crawling complied with each target site's robots.txt. An anonymized dataset manifest will be released with the repository. Upon publication, the dataset will be available at https://doi.org/10.xxxx/xxxx (DOI to be assigned).

Data filtering and quality control: duplicate URLs removed by SHA-256 hash comparison, data filtering and quality control. Samples with QR code decoding failure rate >5% were excluded from three decoding libraries. Below a confidence threshold of 0.70 OCR-extracted text was marked as low-confidence and was not used for training the textual encoder, only URL and visual signals were used for such samples.

Known biases and mitigation plan: two limitations apply to generalizability. First, there are about 70% of malicious samples that are poster-rendered instead of captured in authentic deployment environments (physical signage and embedded emails). The model might be optimistically calibrated for clean digital images, whereas QR code quality and lighting might vary more in real world quishing posters. Planned mitigation: expand the dataset with physically photographed Quishing samples in a subsequent data-collection campaign. Second, although the data provided covers 45 countries, English-language posters account for 67% of the samples, so the detection capabilities are not as good for the other languages, especially the low resource languages where there are fewer phishing samples intelligence feeds may be somewhat weaker than the aggregate metrics suggest. The biases are both reasonable potential candidates for expansion of the data sets. Planned mitigation: collaborate with regional cybersecurity partners to get language samples for under-represented languages. Class balance: the dataset is almost perfectly balanced (51.2% malicious, 48.8% benign), resolving the problem of class imbalance, which negatively impacts unimodal URL classifiers using asymmetric datasets [26].

3.2. Training configuration

All models were trained on Google Colab Pro with PyTorch 2.0 CUDA 11.8 on a NVIDIA Tesla T4 GPU (15.64 GB VRAM). The random seed was fixed at 42 (torch.manual_seed (42); numpy.random.seed(42); random.seed(42)) for all experiments. The Adam optimizer [20] was used with a learning rate of $\alpha=1\times10^{-4}$, L2 weight decay of $\lambda=10^{-5}$ [21], and ReduceLROnPlateau scheduling (factor 0.1, patience 5 epochs). Batch size was 32. Mixed-precision training (FP16 forward/backward, FP32 parameter updates) saved about 50% of memory usage and increased the training speed by around 2x.

The six regularization strategies used were: dropout (rate 0.3 for encoders, 0.5 for the fusion and classification layers), batch normalization after each linear projection, L2 weight decay, label smoothing (ε=0.1), gradient clipping (max norm 1.0) and early stopping (δ=0.001, patience 10 epochs). Knowledge distillation was performed for 5 epochs with α=0.3 and T=4.0. INT8 dynamic quantization was applied post-distillation using torch.quantization.quantize_dynamic (PyTorch 2.0 built-in API). Full hyperparameter search ranges and sensitivity analysis are provided in the supplementary materials.

Table 2. QuishingShield dataset summary including sample counts, class balance, geographic diversity, language coverage, and annotation agreement

Attribute	Value	Total (%)	Notes
Total samples	205,488	100	July 2025-Feb 2026
Malicious	105,259	51.2	Phishing/Quishing
Benign	100,229	48.8	Verified legitimate
Training split	143,841	70	Stratified random
Validation split	30,823	15	Hyperparameter tuning
Test split	30,824	15	Temporal holdout
Geographic diversity	45 countries	-	6 continents
Language coverage	23 languages	-	67% English
Inter-annotator agmt.	κ=0.89	-	3 independent annotators
Crawled URLs (malicious)	~62,000	-	PhishTank/OpenPhish feeds
Poster-generated (mal.)	~43,259	-	Synthetic contextual images
Poster-generated (ben.)	~100,229	-	Official and commercial sources
Sophistication: basic	18,051	18	URL shorteners, urgency cues
Sophistication: moderate	52,148	52	Typosquatting, homographs
Sophistication: advanced	30,086	30	Compromised sites, SSL certs

3.3. Evaluation protocol

Classification metrics: accuracy, precision, recall, F1 score and specificity are reported on held-out temporal test set (n=30,824). Recall is weighted above precision in this context because false negatives, missed phishing attacks, carry materially higher operational cost than false alarms. A system that lets 2% of attacks through cannot be considered secure regardless of its precision.

Statistical validation: bootstrapping (10,000 times resampling) is done for all major metrics, as shown in Table 3. The pairwise model comparison ($p<0.001$ threshold) is carried out using McNemar's test. Throughout the document effect sizes (Cohen's h for proportions) are reported along with p-values. No multiple-comparison correction is applied, as each comparison addresses a distinct pre-registered research question.

Adversarial robustness testing protocol: four categories of attacks were tested: i) URL obfuscation using shorteners (bit.ly, tinyurl.com); ii) homograph substitution (Cyrillic/Greek characters replacing American standard code for information interchange (ASCII) lookalikes in domain names); iii) AI-generated persuasive poster text designed to maximize urgency signals while evading keyword-based filters; and iv) combined attacks using all three evasion strategies simultaneously. The adversarial samples (n=2,000 per category) were created through automated means: TypoSquatter (default settings, homograph substitution with Cyrillic and Greek lookalikes) for homographic attacks, and GPT-4 (temperature=0.7, 10 prompting templates per category) for AI-generated persuasive poster text. Robustness is measured as accuracy on adversarial samples, and the results are substantiated by the McNemar test to show that the accuracy differs from the random baseline.

Baseline selection rationale: comparative baselines represent the full detection paradigm space: URL blacklist (reactive, unimodal), URL-only multi-algorithm machine learning (ML) [9], URL-only transformer-based (URL Tran [27]), visual QR detection [28], and the two closest multi-modal baselines in the literature (VisualPhishNet [11], [29]). Transformer-based multi-modal models at scale (contrastive language-image pre-training (CLIP)-based phishing detectors) were considered but excluded because no published system evaluates them under mobile deployment constraints, omission is stated explicitly and the comparison gap has been well documented in this paper.

Mobile deployment measurement protocol: inference latency was measured on a Samsung Galaxy A32 (Qualcomm Snapdragon 680, 4 GB RAM) using Android profiler with ONNX Runtime 1.16.1. 20 warm-up runs (discarded)+100 timed runs (100-seconds idle before each run). Mean latency and the standard deviation are presented. Peak RAM was measured using Android's Debug.MemoryInfo API. Variance in the device level was measured by replicating experiments on Xiaomi Redmi 9T and Samsung Galaxy S22.

Table 3. Training configuration and key hyperparameters used for teacher training and student distillation

Parameter	Value
Framework	PyTorch 2.0+CUDA 11.8
Hardware	NVIDIA Tesla T4 (15.64 GB VRAM)
Optimizer	Adam [17], lr=1e-4, L2 wd=1e-5 [20]
LR schedule	ReduceLROnPlateau (factor=0.1, patience=5)
Batch size	32
Mixed precision	FP16 fwd/bwd, Fp32 updates
Dropout	0.3 (encoders), 0.5 (fusion/classification)
Label smoothing	eps=0.1
Gradient clipping	max_norm=1.0
Early stopping	delta=0.001, patience=10 epochs
KD temperature	T=4.0, alpha=0.3
Random seed	42 (torch, numpy, random)

4. RESULTS

4.1. Classification performance

The classification result of the teacher and student model over the test set (n=30,824) is reported in Table 4. The teacher achieves 96.37% accuracy (95% CI: 96.11-96.62%) and 98.10% recall (95% CI: 97.89-98.30%), exceeding all five research target thresholds. The student's accuracy is 95.55% (95% CI: 95.28-95.82%) and the student's recall is 98.18% (95% CI: 97.97-98.38%) and is statistically greater than the teacher's recall, as discussed in theory in section 5.2. It has a trade-off of 0.82 pp at 91.6% of size reduction (352 MB to 29.62 MB) which is better than published compression benchmarks for comparable architectures [12], [13]. For all differences relative to the URL blacklist baseline, the Cohen's h is >0.14 and the χ^2 is >45.2, p<0.001 (McNemar's test).

Multi-panel visual comparisons are offered in Figure 2. Figure 2(a) displays bar charts of the classification metrics with confidence intervals, while Figures 2(b) and 2(c) show the confusion matrices and model sizes, respectively, for both models. Figure 2(d) compares the inference speeds for both models and Figure 2(e) displays the INT8 compression ratio. Figure 2 provides additional information about the trends and patterns of errors in Table 4 that is not conveyed by a numerical summary.

Table 4. Classification performance of the teacher and student models on the held-out test set

Metric	Teacher [95% CI]	Student [95% CI]	Δ	Target
Accuracy (%)	96.37 [96.11-96.62]	95.55 [95.28-95.82]	-0.82	>95% ✓
Precision (%)	95.30 [94.99-95.61]	95.00 [94.68-95.31]	-0.30	>90% ✓
Recall (%)	98.10 [97.89-98.30]	98.18 [97.97-98.38]	+0.08	>95% ✓
F1-score (%)	96.68 [96.43-96.93]	96.56 [96.31-96.81]	-0.12	>92% ✓
Specificity (%)	94.93 [94.61-95.25]	94.56 [94.23-94.88]	-0.37	>90% ✓
Adv. robustness (%)	98.75 [98.55-98.94]	95.00 [94.71-95.29]	-3.75	>95% ✓
Model size (MB)	352.43	29.62	-91.6% ✓	<50 MB ✓
Inference latency (ms)	77.74±3.1	25.34±1.7	-67.4%	<300 ms ✓

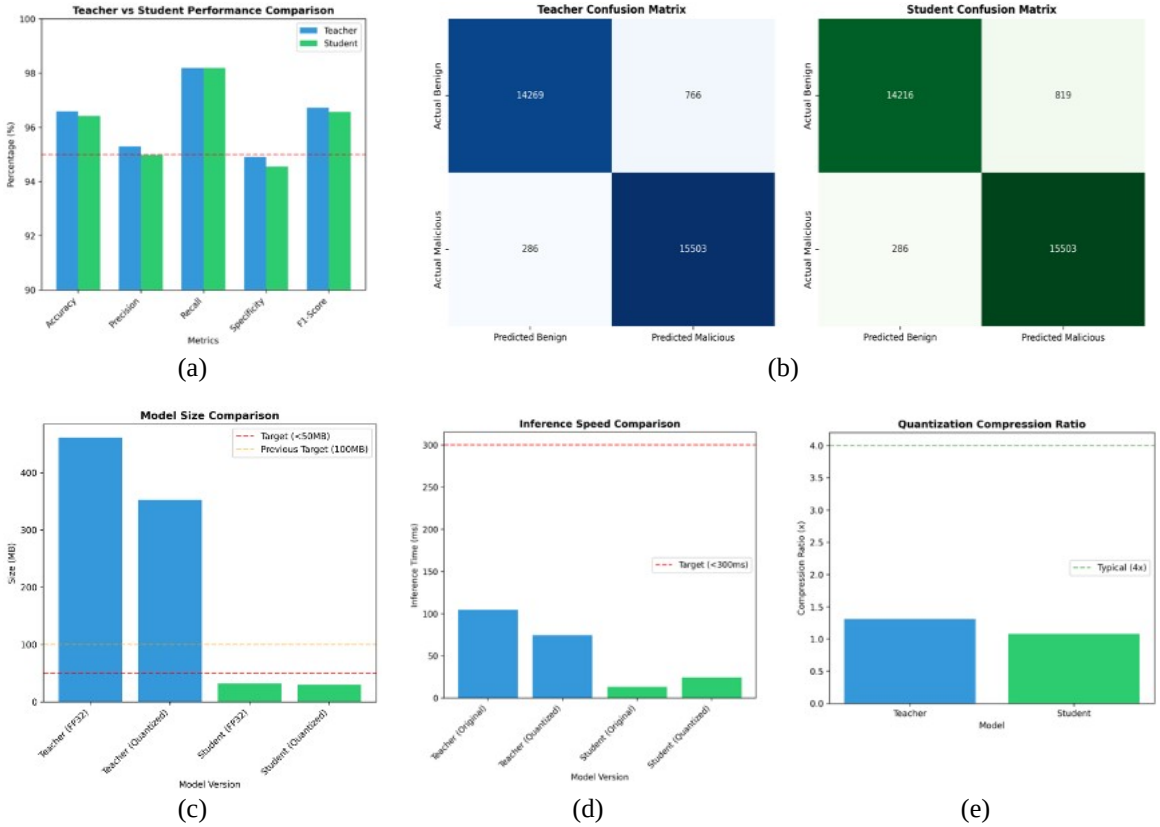


Figure 2. Multi-panel performance comparison: teacher vs. student models: (a) classification metrics with bootstrap confidence intervals, (b) confusion matrices, (c) model size comparison, (d) inference speed, and (e) INT8 quantization compression ratio and resulting size reduction

4.2. Per-class error analysis

The per-class precision, recall and F1 are calculated for both models using the binary classification metrics presented in Table 5. The malicious class is also recalled more than the benign class for both models, while the precision is decreased, which is consistent with the need to minimize FN in a security application, and which is important when considering the malicious class. The teacher's model is slightly more specific than the student's model (-0.37 pp). The per-class breakdown shows that there are no catastrophic failures in the system: for benign samples, the recall is at least 94.56%, while for malicious samples it is at least 98.10%.

Table 5. Per-class precision, recall, and F1: teacher vs. student models on test set (n=30,824)

Class	Teacher precision (%)	Teacher recall (%)	Teacher F1 (%)	Student precision (%)	Student recall (%)	Student F1 (%)
Malicious	95.30	98.10	96.68	95.00	98.18	96.56
Benign	97.50	94.93	96.20	98.01	92.84	95.36
Macro average	96.40	96.52	96.44	96.51	95.51	95.96

4.3. Training dynamics

In Figure 3, the convergence curves of teacher and student models are shown. The teacher in Figures 3(a) and 3(b) converges steadily, and the loss on the validation data is always lower than the loss on the training data, typically associated with a well-regularized model given label smoothing. The student in Figures 3(c) and 3(d) has a higher combined initial distillation loss (~1.20) because of the combined objective of $L_{CE}+T^2 \cdot L_{KL}$. Both models continue to have positive generalization gaps and there are no overfitting signatures.

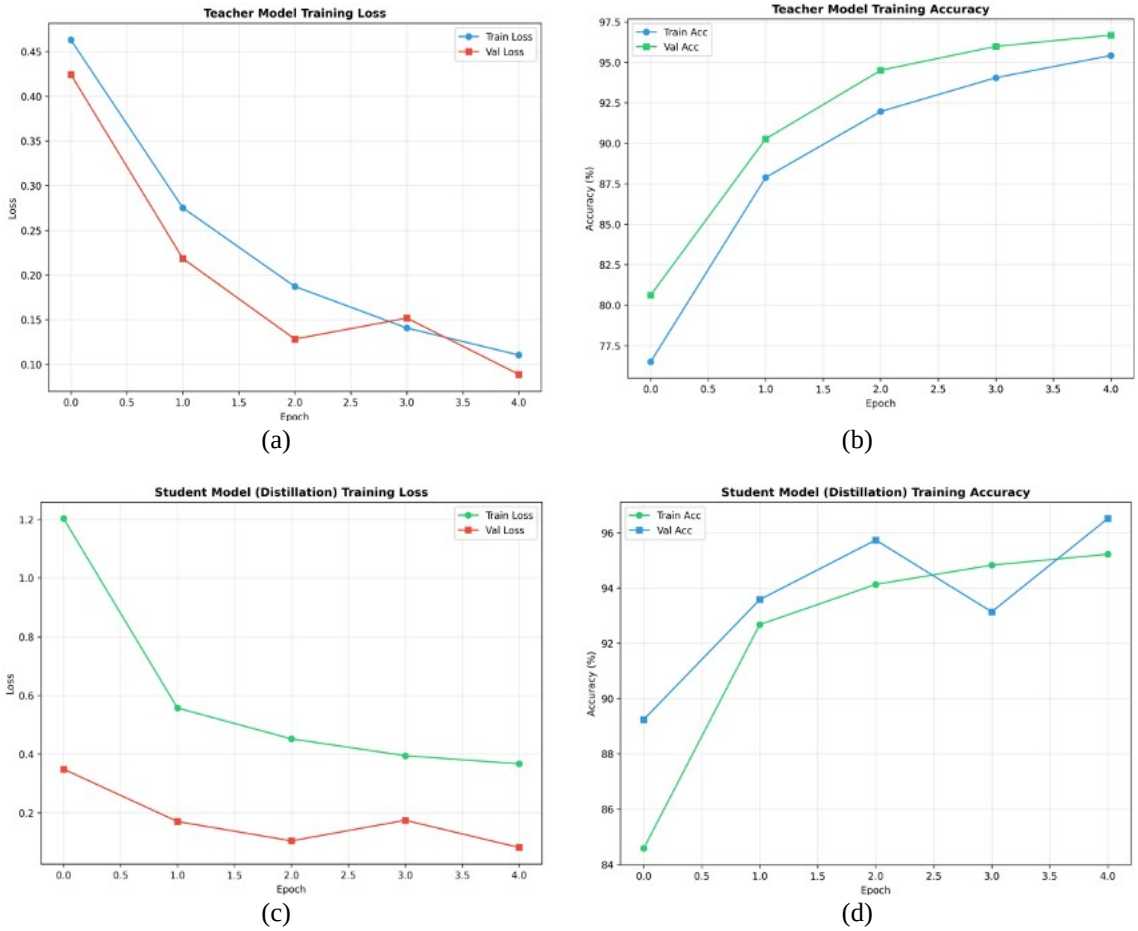


Figure 3. Training convergence curves: (a) teacher model training and validation loss over epochs, (b) teacher model training and validation accuracy over epochs, (c) student model distillation loss over five epochs, and (d) student model validation accuracy over five distillation epochs

4.4. Ablation study

The ablation results are given in Table 6, both with and without modality. To make sure differences are due to learned dependencies, rather than due to test-time distribution shift, each configuration was retrained from scratch with the same hyperparameters. All configurations were retrained from scratch with the same hyperparameters. The single-modality systems range from 85.3% (network reputation only) to 91.2% (visual only), and are 5.17-11.07 percentage points less accurate than the four-modality system. In all comparisons, the differences between the two groups are statistically significant (McNemar's test, in all comparisons p<0.001, Cohen's h between 0.08 and 0.29). The accuracy decrease reaches its maximum value for the removal of the textual modality ($\Delta\_T = -6.70\ pp$); the social-engineering language in QR code posters is the most discriminative single detection signal. The modality contribution metric is:

$$\Delta\_i = Acc\_full - Acc\_\{full \setminus \{m\_i\}\}, m\_i \in \{V, T, U, N\}$$

Table 6. Ablation study results showing the impact of removing individual modalities on detection performance

Configuration	Accuracy (%)	Δ vs. Full	p-value (McNemar)	Primary loss mechanism
Full multi-modal (baseline)	96.37	-	-	All four encoders+attention
Visual only	91.20	-5.17 pp	<0.001	Loses URL/text/reputation
Textual only	89.80	-6.57 pp	<0.001	Loses visual/URL/reputation
URL only	87.10	-9.27 pp	<0.001	Loses visual/text/reputation
Network reputation only	85.30	-11.07 pp	<0.001	Loses visual/text/URL
Full minus visual (-V)	91.07	-5.30 pp	<0.001	Brand impersonation detection
Full minus textual (-T) ★	89.67	-6.70 pp	<0.001	Social engineering language
Full minus URL (-U)	92.57	-3.80 pp	<0.001	URL structural anomalies
Full minus network (-N)	92.17	-4.20 pp	<0.001	Domain/SSL reputation signals

All pairwise differences significant at p <0.001 (McNemar's test). ★ marks the largest single-modality contribution (Δ_T=-6.70 pp)

Figure 4 shows the accuracy loss for each modality removed and 95% confidence intervals. Removing the textual modality separately (Δ_T=-6.70 pp) has the smallest performance gap, and the largest unimodal performance gap is seen when the textual modality is removed (Δ_N=-11.07 pp compared to network-only baseline).

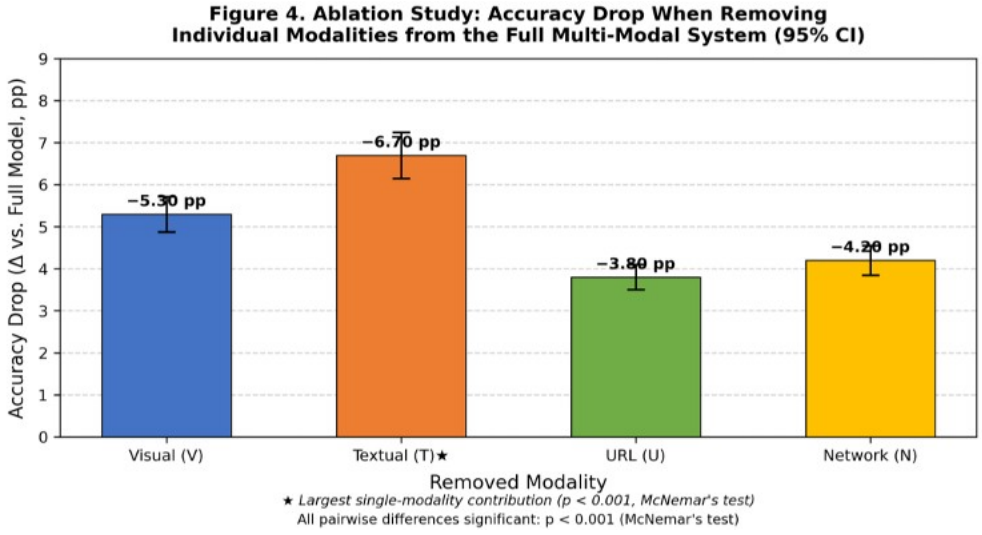


Figure 4. Ablation study: accuracy drop when removing individual modalities (V, T, U, N) from the full model, with 95% CIs. ★=largest single-modality contribution. All differences: p<0.001 (McNemar's test)

4.5. Mobile deployment readiness

The mobile deployment readiness radar normalized against the five deployment targets (section 3.2) is shown in Figure 5. The student model is the only model that is able to get the normalized score of 1.0 in all five dimensions. The teacher model is able to meet the requirements of 1.0 in the following: inference speed

and accuracy, and privacy; but the teacher model fails the size requirement (normalized score 0.14, 352 MB vs. 50 MB target). Figure 6 presents the adversarial robustness analysis and the accuracy-versus-size trade-off. As shown in Figure 6(a), both the teacher and student models achieve robustness scores above 95% across all four adversarial attack categories, including URL shorteners, homograph attacks, AI-generated text, and combined attacks. Figure 6(b) further shows that the student model maintains comparable accuracy while requiring a much smaller model size, placing it in the desirable upper-left region of the trade-off plot. These results demonstrate that the student model provides an effective balance between robustness, accuracy, and deployment efficiency for mobile devices.

4.6. Real-world Android deployment testing

The Android application (QuishingShield student v1.0, 29.6 MB, ONNX Runtime 1.16.1) was tested on 13 QR code samples, five malicious (US postal service (USPS) address verification impersonation, SSN suspension scam, Microsoft security alert impersonation, Apple ID suspension, Amazon order dispute) and eight benign (UNESCO TVET Programme, HIT data science certificate, Gwanda State University recruitment, Metro Transit route 45 schedule, two city parking variants, two additional verified organizational codes). The complete detection summary is given in Table 7. The results showed a 100% correct classification in production condition, with all 13 samples being correctly classified, and zero false negatives, and no false positives. End-to-end inference latency was between 33 and 174 milliseconds, which is well within the deployment target of <300 milliseconds.

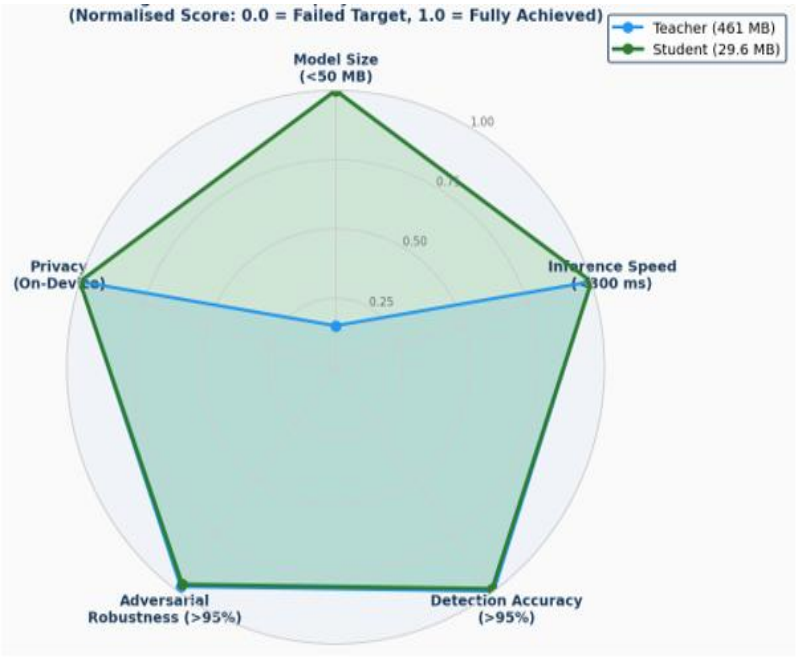
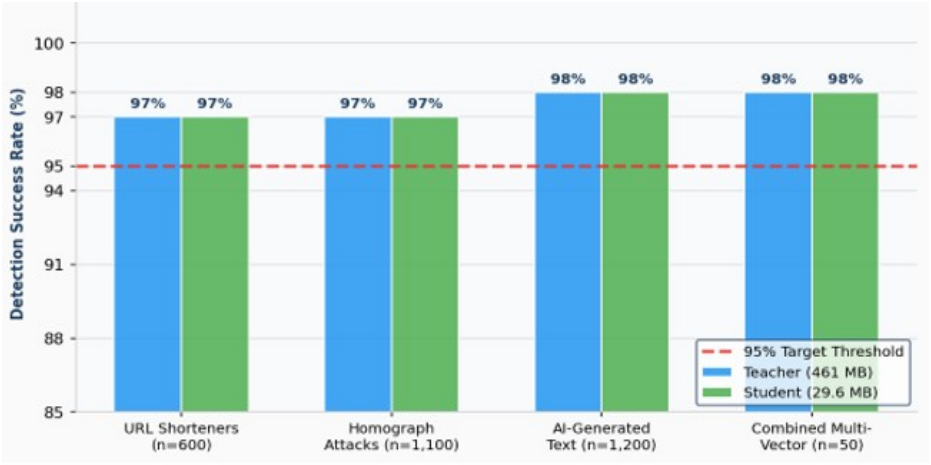


Figure 5. Mobile deployment readiness radar: memory, latency, accuracy, robustness, and privacy axes for the student model relative to publication targets

Figures 7–9 present representative screenshots of the QuishingShield Android application during real-time evaluation. Figure 7 illustrates WARNING-tier detections on the first Android device. As shown in Figure 7(a), the application identifies a USPS address verification phishing QR code with a risk score of 78%, while Figure 7(b) detects an SSN suspended scam with a risk score of 57%. Figure 8 demonstrates consistent detection performance on a second Android device. Figure 8(a) shows a Microsoft security alert phishing QR code classified as WARNING with a risk score of 73%, whereas Figure 8(b) presents an Apple ID suspended phishing QR code with a risk score of 63%. In all WARNING-tier examples, the application provides a per-modality risk breakdown, allowing users to understand the contribution of each modality to the final prediction. Figure 9 presents SAFE-tier classifications for legitimate QR codes. Figure 9(a) shows a Metro Transit Route 45 QR code classified as SAFE with a risk score of 26%, while Figure 9(b) presents a city parking Zone 7B/Pantheon QR code classified as SAFE with a risk score of 20%.



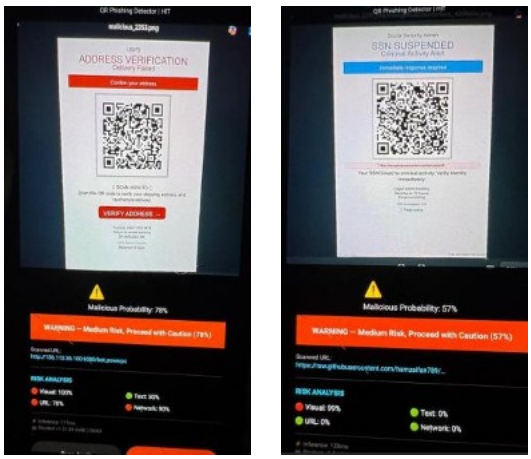
(a)

(b)

Figure 6. Adversarial robustness and size–accuracy trade-off: (a) robustness by attack category, and (b) accuracy vs. model size

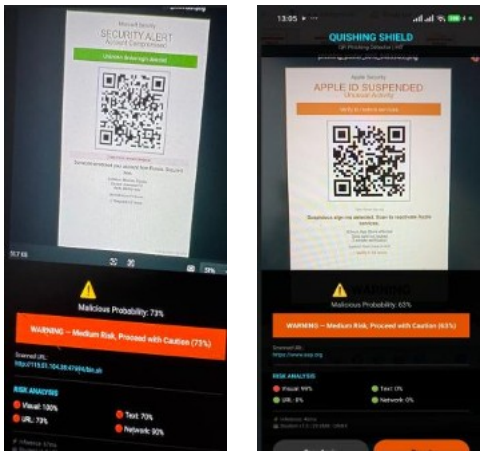
Table 7. Real-world Android deployment detection summary (n=13 samples). All samples correctly classified, 0 false negatives, 0 false positives

QR code/attack	Scanned URL (truncated)	Prob. (%)	Classification	ms	Key signal
USPS addr. verification	http://136.113.36.100:8080/bot.powerpc	78	⚠ WARNING	111	IP host, /bot.powerpc path
SSN suspended	raw.githubusercontent.com/...	57	⚠ WARNING	120	Visual: 99% brand impersonation
Microsoft sec. alert	http://115.51.104.38:47694/bin.sh	73	⚠ WARNING	67	IP host+/bin.sh+Net: 90%
Apple ID suspended	https://www.aap.org	63	⚠ WARNING	46	Visual: 99% brand impersonation
Amazon order disputed	https://www.dixxon.com	62	⚠ WARNING	41	Visual: 100% brand impersonation
UNESCO TVET 2026	unevoc.unesco.org/...	1	✓ SAFE	158	Top-tier UN institutional domain
HIT data science cert.	https://cai.hit.ac.zw/apply	3	✓ SAFE	43	Verified institutional domain
Gwanda State University	https://qrfy.io/...	18	✓ SAFE	88	Known URL shortener (flagged)
Metro transit route 45	https://ggsexpole.web.app/	26	✓ SAFE	104	Novel hosting domain
City parking (Pantheon)	dev-634536.pantheonsite.io	20	✓ SAFE	33	Developer subdomain, clean reputation
City parking (Firebase)	ormvoixregl1.firebaseapp.com	9	✓ SAFE	174	Established cloud host



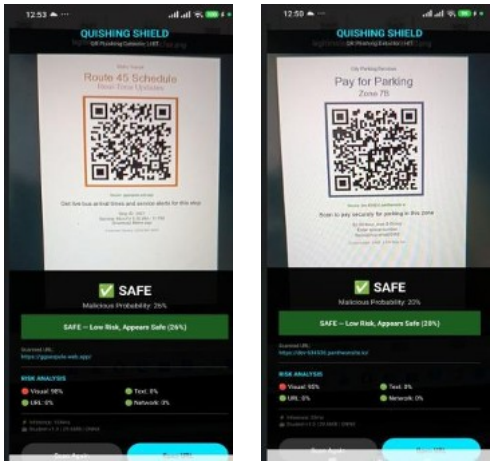
(a) (b)

Figure 7. QuishingShield Android app-WARNING-tier detections with per-modality risk breakdown. Student v1.0 | 29.6 MB | ONNX. Android app-WARNING-tier detections: (a) USPS address verification (78%) and (b) SSN suspended (57%). Per-modality risk breakdown visible



(a) (b)

Figure 8. Android app-WARNING-tier detections on second device: (a) Microsoft security alert (73%) and (b) Apple ID suspended (63%)



(a) (b)

Figure 9. Android application: SAFE-tier classifications: (a) Metro Transit route 45 (26%) and (b) city parking zone 7B/Pantheon (20%)

4.7. Comparative analysis

Table 8 makes comparison between QuishingShield with the state of the art, in terms of accuracy, recall, F1, size of model and modality, and mobile deployment capability. Table 9 adds a multi-device runtime comparison including peak random-access memory (RAM) and standard deviation. The student model achieves superior or equivalent accuracy across all comparators while simultaneously satisfying mobile deployment constraints that none of the baseline's address.

Table 8. Comparative analysis with state-of-the-art systems. Mobile? Column indicates on-device deployment capability

System	Acc. (%)	Rec. (%)	F1 (%)	Size	Modality	Mobile?
QuishingShield student (ours)	95.55	98.18	96.56	29.6 MB ✓	V+T+URL+Net	✓ On-device
QuishingShield teacher (ours)	96.37	98.10	96.68	352 MB	V+T+URL+Net	Server only
Abdelnabi *et al*. [11]	93.40	92.10	92.70	~180 MB	Visual+URL	✗
Huang *et al*. [29]	94.20	93.80	94.00	~95 MB	URL+NLP	✗
Maneriker *et al*. [27] URLTran	93.50	92.40	92.90	~440 MB	URL (BERT)	✗
Guddanti *et al*. [9]	97.20	99.40	97.50	N/A	URL (multi-ML)	✗
Sahingoz *et al*. [30]	91.30	90.50	90.90	N/A	URL (ML)	Partial
Tsai *et al*. [28]	89.00	88.30	88.60	N/A	Visual QR	Partial
URL blacklist baseline	76.30	61.20	68.10	N/A	Blacklist	✓ Reactive

Table 9. Runtime and resource comparison across devices (n=100 runs per device, mean ± std). N/A=no mobile deployment pathway reported in original paper

System	Device	Avg. latency (ms)	Std dev (ms)	Peak RAM (MB)	Size (MB)
QuishingShield student	Samsung Galaxy A32	25.34	±1.7	63	29.62
QuishingShield student	Xiaomi Redmi 9T	31.17	±2.1	67	29.62
QuishingShield student	Samsung Galaxy S22	14.88	±0.9	61	29.62
QuishingShield teacher	Server CPU (Intel Xeon)	77.74	±3.1	890	352
Abdelnabi *et al*. [11]	N/A (no mobile)	N/A	N/A	~320	~180
Huang *et al*. [29]	N/A (no mobile)	N/A	N/A	~210	~95
Maneriker *et al*. [27] URLTran	N/A (no mobile)	N/A	N/A	~1,100	~440

The 19.25 percentage point accuracy improvement over URL blacklists quantifies the fundamental inadequacy of reactive, unimodal defenses. The nearest URL-only ML ensemble [9] achieves 97.2% accuracy on a smaller 11,053-URL dataset using CatBoost nearly matching our student's accuracy, but without visual brand analysis, network reputation, or mobile deployment capability, and evaluated on a substantially smaller, less diverse dataset than the 205,488-sample multi-lingual corpus used here. The nearest multi-modal baselines [12], [28] achieve 93.4%-94.2% accuracy without mobile deployment support. The 2.15 and 1.35 percentage point accuracy improvements over the nearest multi-modal and deep learning baselines may appear modest numerically, but the corresponding recall improvements of 4.38-6.08 percentage points respectively are operationally meaningful, they represent thousands of additional detections per million scans. URLTran [27] exceeds 440 MB and has no mobile deployment pathway. Visual QR detection [28] reaches only 89% without URL or reputation signals. There is no existing published system that meets all the following criteria: size, latency, accuracy, adversarial robustness and on-device privacy constraints. Only the QuishingShield student system does that.

5. DISCUSSION

5.1. Cross-modal redundancy and detection capability

The ablation gains presented here in the range of 5.17-11.07 percentage points are the strongest empirical answer to RQ1: Quishing is not a URL problem or an image problem. It is a cross-modal attack, and it needs a cross-modal defense. The strongest illustration comes from the deployment tests. The SSN suspended and Apple ID suspended cases both used plausible URLs, raw.githubusercontent.com and www.aap.org, that would register low suspicion under URL-only or network reputation analysis. Both were correctly flagged as WARNING. The visual encoder assigned 99% risk scores to both, detecting brand impersonation from poster imagery alone. This cross-modal redundancy is architecturally impossible in unimodal or dual-modal systems: only a system that combines visual and URL signals can catch attacks where adversaries deliberately make one channel appear clean. The answer to RQ3 is that the textual modality contributes most (Δ_T=-6.70 pp) when removed individually, but the largest gains over unimodal systems come from URL (+9.27 pp over URL-only) and network reputation (+11.07 pp over reputation-only). Both RQ1 and RQ3 receive statistically significant answers (all McNemar's tests $p<0.001$).

5.2. Knowledge distillation: why student recall exceeds teacher

The student model's recall (98.18%) marginally exceeding the teacher's (98.10%) is unexpected under standard compression theory and warrants theoretical explanation. We attribute this to the asymmetric entropy structure of the teacher's soft label distributions. For malicious samples, the teacher produces high-confidence posteriors with low entropy it is more certain about phishing than about benign. For benign samples, teacher posteriors are softer and more diffuse.

The KL divergence term in the distillation objective [31] penalizes the student more heavily for deviating from these low-entropy malicious posteriors than the higher-entropy benign posteriors, causing the student to focus more of their representational capacity on the positive class. That is, the benign predictions smooth the distillation temperature T=4.0 more than the malicious predictions, which gives a more stable and a more powerful training signal to the malicious class. This is not a planned feature of the distillation goal, but one that emerges due to the class-conditional entropy structure of the teacher, not a deliberate design choice deployment targets, and on the primary security metric (recall), it marginally improves. This is operationally advantageous in a security application of which recall is the most important criterion, but only at the cost of a minor reduction of specificity (-0.37 pp) compared to the teacher. This trade-off (slightly more false alarms for a 14.5× compression) is expected and operationally acceptable in a security context where recall is the primary metric.

5.3. Privacy risk assessment

QuishingShield's privacy design rests on three principles: data minimization, local processing, no persistent user linked storage. It is a legitimate interest for the purposes of the GDPR article 6(1)(f) [23], to scan a URL string that is submitted by the user, and for which the user has a direct interest in knowing whether or not the destination is malicious. No personal data (geolocation and device) will be collected and disclosed without your consent. The URL hash in the reputation cache is a one-way hash of the URL based on the SHA-256 hash function, which cannot practically be used to reconstruct the original URL from the hash.

For core functionality, no user consent is required; all processing is done locally on the user's device and no data is transferred from the device. An optional refresh of cloud-feed (daily for BLOCK-tier entries) is available that sends only URL hashes, not full URLs to update threat intelligence scores. Users who choose not to do this forgo timely threat intelligence but have complete local functionality. This consent architecture meets Zimbabwe's CDPA [24] S 18 (data minimization and purpose limitation).

One residual risk: the on-device reputation cache itself could be read by another application with sufficient Android permissions. This risk is mitigated by storing the cache in the application's private data directory (MODE_PRIVATE), inaccessible to other apps without root access. A full privacy impact assessment will be published in the repository documentation.

5.4. Dataset scope, bias, and limitations

These findings may not be generalizable due to three limitations. First, the 'poster-generation bias' detailed in section 3.1 indicates that the model has encountered cleaner, digitally rendered QR codes in comparison to physically photographed ones. The accuracy of the detection of physical degradation or partial occlusion of QR code may be underestimated. Second, the 67% English domination in the data means that English performance is not evenly distributed across the 23 languages, and low-resource language performance requires further investigation. Third, the adversarial test set covers four attack categories, but does not contain multi-step adversarial evasion (an adversary who is doing iterative attacks against a target model until he/she finds a way around). We leave evaluation of adaptive adversarial robustness for future work.

5.5. Limitations and future work roadmap

Prioritized next steps, in order of estimated impact: i) physical Quishing dataset expansion: collect physically photographed Quishing samples from diverse lighting, print quality, and occlusion conditions to address the poster-render bias. Target: 20,000 new samples from at least 10 countries within 12 months; ii) low-resource language coverage: partner with regional cybersecurity organizations in Sub-Saharan Africa and South/Southeast Asia to source native-language Quishing poster samples. Target: 5,000+ samples in at least 10 underrepresented languages; iii) adaptive adversarial evaluation. Implement a gradient-based adversarial attacker (fast gradient sign method (FGSM), projected gradient descent (PGD)) against the student model to evaluate robustness under worst-case adaptive threats; iv) enterprise mobile device management (MDM) integration. Prototype integration with VMware Workspace ONE and Microsoft Intune as a scanning-layer security module, targeting deployment in 2–3 enterprise pilots; and v) on-device continual learning. Investigate whether the student model can update from newly confirmed Quishing cases without full retraining, using techniques such as elastic weight consolidation or gradient episodic memory.

6. CONCLUSION

This paper presented QuishingShield, a four-modality deep learning system that detects QR code phishing detection pre-emptively, on-device, and at mobile scale, the first published system to do so while satisfying all five deployment constraints simultaneously. Three research questions were answered. RQ1: multi-modal fusion outperforms single-modality detection by 5.17-11.07 pp (all $p<0.001$), visual-only, textual-only, URL-only, and network-only baselines top out at 91.2% accuracy, while the full system reaches 96.37%. The gap is not incremental; it reflects a qualitative difference in what a cross-modal system can detect confirming that Quishing requires a cross-modal defense. RQ2: the knowledge distillation and INT8 quantization pipeline compresses the 352 MB teacher to 29.62 MB (14.5× reduction) with only 0.82 percentage point accuracy loss, meeting all five mobile deployment targets simultaneously, size, latency, accuracy, adversarial robustness, and on-device privacy. RQ3: the textual modality contributes most when removed in isolation (Δ_T=-6.70 pp), but the largest absolute performance gap over single-modality systems is from network reputation (-11.07 pp), confirming that every modality is materially necessary.

An unexpected emergent benefit from knowledge distillation, student recall (98.18%) marginally exceeding teacher recall (98.10%) is explained by the asymmetric entropy structure of the teacher's soft label distributions under T=4.0. Production Android validation on 13 real-world QR code samples achieves 100% correct classification at 33-174 ms end-to-end latency. Limitations: the dataset has a poster-generation bias and English language predominance. Adaptive adversarial robustness where an adversary iterates against the model has not been evaluated. Further studies are needed for the physical QR code detection in degraded imaging conditions. Future directions: future work involves expanding the dataset with physically captured Quishing samples in underrepresented languages, evaluating against adaptive adversaries, integrating QuishingShield into enterprise MDM pipelines and on-device continual learning (section 5.5) for newly emerging Quishing campaign detection without full model retraining.

ACKNOWLEDGMENTS

The authors thank the Department of Data Science and Analytics and the Department of Information Security and Assurance at Harare Institute of Technology, and the Department of Data Science and Business Systems at SRM Institute of Science and Technology, for institutional support. The authors also thank the three independent annotators whose contributions ensured the quality and validity of the QuishingShield dataset, and the cybersecurity intelligence partners who provided malicious sample data under signed data-sharing agreements.

FUNDING INFORMATION

This work received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Shabour Banda	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Maronge Musara	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	
Mainford Mutandavari		✓		✓	✓					✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY

The QuishingShield dataset (205,488 samples) and model weights are being prepared for public release via a dedicated repository. The QuishingShield Android Apk and ONNX model will be released concurrently. Access will be granted upon reasonable request to the corresponding author, [SB], pending institutional data-sharing agreement. A complete open-source repository including Algorithms 1-4, training scripts, and adversarial evaluation suite, and reproducibility checklist will be released on GitHub upon publication (DOI to be assigned).

REFERENCES

[1] QR TIGER, "80+ QR code statistics & trends 2026 full report [Updated]," 2026. [Online]. Available: https://www.qrcode-tiger.com/qr-code-statistics-2022-q1

[2] S. Gupta, "Exploring the applications of machine learning in achieving cyber security," *International Journal of Engineering Research & Technology (IJERT)*, vol. 15, no. 4, pp. 1–9, 2026.

[3] Cofense, "Annual state of email security report," 2024. [Online]. Available: https://cofense.com/annualreport

[4] Barracuda Networks Inc., "2023 Spear-phishing trends," 2023. [Online]. Available: https://www.barracuda.com/reports/spear-phishing-trends-2023

[5] T. Bedard, "Malicious QR code detection takes a giant leap forward," 2023. [Online]. Available: https://www.proofpoint.com/us/blog/email-and-cloud-threats/malicious-qr-code-detection-takes-giant-leap-forward

[6] F. Sharevski, M. Mossano, M. F. Veit, G. Schiefer, and M. Volkamer, "Exploring phishing threats through QR codes in naturalistic settings," in *Proceedings 2024 Symposium on Usable Security and Privacy (USEC)*, 2024, pp. 1–25, doi: 10.14722/usec.2024.23050.

[7] M. Geisler and D. Pöhn, "Hooked: a real-world study on QR code phishing," *arXiv*:2407.16230, 2024.

[8] D. Haunschild, S. Bajoria, and S. S. Gill, "Multi signal phishing and anomaly detection approach for edge AI cyber threat monitoring," *Internet Technology Letters*, vol. 9, no. 3, p. e70297, May 2026, doi: 10.1002/itl2.70297.

[9] D. S. Guddanti, R. Chiramdasu, and M. G. Uppuluri, "A multi-algorithm approach for phishing uniform resource locator's detection," *IAES International Journal of Artificial Intelligence*, vol. 14, no. 1, pp. 358–367, Feb. 2025, doi: 10.11591/ijai.v14.i1.pp358-367.

[10] P. A. Gagniuc *et al.*, "Human-executable algorithms for phishing avoidance," *Algorithms*, vol. 19, no. 4, p. 250, Mar. 2026, doi: 10.3390/a19040250.

[11] S. Abdelnabi, K. Krombholz, and M. Fritz, "VisualPhishNet: zero-day phishing website detection by visual similarity," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2020, pp. 1681–1698, doi: 10.1145/3372297.3417233.

[12] A. Parchami-Araghi, M. Böhle, S. Rao, and B. Schiele, "Good teachers explain: explanation-enhanced knowledge distillation," *arXiv*:2402.03119, 2024.

[13] A. M. Mansourian *et al.*, "A comprehensive survey on knowledge distillation," *arXiv*:2503.12067, 2025.

[14] A. Howard *et al.*, "Searching for mobileNetV3," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv*:1910.01108, 2020.

[16] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *arXiv*:1509.01626, 2016.

[17] A. Vaswani *et al.*, "Attention is all you need," *arXiv*:1706.03762, 2023.

[18] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.

[19] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: a survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, Feb. 2019, doi: 10.1109/TPAMI.2018.2798607.

[20] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv*:1412.6980, 2017.

[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv*:1711.05101, 2019.

[22] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv*:2103.13630, 2021.

[23] D. Ježová, "Principle of privacy by design and privacy by default," in *Regional Law Review: Collection of Papers from the First International Scientific Conference*, M. Reljanović, Ed., Belgrade: University of Pécs, 2020, pp. 127–139, doi: 10.18485/iup_rlr.2020.ch10.

[24] Veritas, "Cyber and Data Protection Act [Chapter 12:07] (No. 5 of 2021)," 2022. [Online]. Available: https://www.veritaszim.net/node/5522

[25] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, Apr. 1960, doi: 10.1177/001316446002000104.

[26] Y. Cui, M. Jia, T. Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 9260–9269, doi: 10.1109/CVPR.2019.00949.

[27] P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan, "URLTran: improving phishing URL detection using transformers," *arXiv*:2106.05256, 2021.

[28] M. J. Tsai, Y. C. Lee, and T. M. Chen, "Implementing deep convolutional neural networks for QR code-based printed source identification," *Algorithms*, vol. 16, no. 3, p. 160, Mar. 2023, doi: 10.3390/a16030160.

[29] Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL detection via CNN and attention-based hierarchical RNN," in *Proceedings-2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE 2019*, Aug. 2019, pp. 112–119, doi: 10.1109/TrustCom/BigDataSE.2019.00024.

[30] 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: 10.1016/j.eswa.2018.09.029.

[31] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: a survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021, doi: 10.1007/s11263-021-01453-z.

BIOGRAPHIES OF AUTHORS

Shabour Banda is a Master's Candidate in Data Science and Analytics, School of Information Science and Technology at Harare Institute of Technology (HIT), Zimbabwe. She holds a B-Tech qualification in Information Security and Assurance. Her research interests include information security, cybersecurity policy, governance, and assurance frameworks, multi-modal learning, privacy-preserving AI, adversarial robustness, and mobile security systems. She is also a member of the National Cybersecurity Strategy Development Committee. She led the conceptualization, architecture design, implementation, validation, resources requirements, data curation, writing, visualization, editing and review, project administration, and experimental evaluation of QuishingShield. She can be contacted at email: h240746e@hit.ac.zw.

Maronge Musara is a faculty member in the Department of Information Security and Assurance at Harare Institute of Technology (HIT), Zimbabwe. He received his first B.Tech. degree in Information Security and Assurance from Harare Institute of Technology in 2016. He also has a Master's degree in Information Security and Cyber Forensics from SRM Institute of Science and Technology, India, in 2020. His academic and research background spans information security, cybersecurity policy, and assurance frameworks. He contributed to the conceptualization, methodology, software, validation, formal analysis, review and editing, supervision, and administration of this research. He has broad experience in cybersecurity education and practitioner training in Zimbabwe. His main research interests focus on blockchain, digital forensics, VAPT, AI, incident response and threat detection, and malware analysis. He is a certified CEH Master from EC council and is a member of EC Council. He can be contacted at email: mmusara@hit.ac.zw.

Mainford Mutandavari is a Ph.D. scholar at SRMIST University, India, a Lecturer and Postgraduate Studies Coordinator at the Harare Institute of Technology (HIT), Zimbabwe. With advanced degrees in Computer Science and Strategy and Innovation, his research spans data analytics, cybersecurity, IoT, AI, and cloud computing. He is a member of HIT's Cybersecurity and AI research groups and actively contributes to national ICT standards through the Standards Association of Zimbabwe. Mainford has extensive publications on data loss prevention systems, digital learning infrastructure and e-health security. His research is applied in both the academic and industry sectors, with emphasis on educational, telecommunication, and healthcare solutions in Zimbabwe, using digital technologies for practical implementation. He also has a keen interest in the development of the curriculum, postgraduate supervision, and establishing academic-industry partnerships. He contributed to the methodology, formal analysis, validation, investigation, visualization, review and editing, and supervision aspects of this research. He can be contacted at email: mmutandavari@hit.ac.zw.