WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

Journal Homepage: www.ijrpr.com ISSN 2582-7421

# Threat Guard: AI Threat Detector for Emails, Links, QR Code & Text Messages: Safe or Malicious. Using Machine Learning

***K M Keerthi Kumar[1], Suresh H J[2], Ananth Kumar K[3], Pavan Kumar M[4]; Ms. Pooja M V[5]***

[1,2,3,4] Department of Information Science & Engineering, GMIT Davanagere, Karnataka, India

[5]Guide, Assistant Professor, Dept. of Information Science & Engineering, GMIT Davanagere

**ABSTRACT:**

Phishing attacks have evolved from monolithic email lures to omni-channel campaigns delivered through obfuscated URLs, SMS (smishing), and QR codes (quishing). Static blocklists react too slowly to short-lived domains, deep neural detectors are opaque and costly, and automated scanners that fetch suspect links can be abused through server-side request forgery (SSRF). To address these gaps, this paper proposes ThreatGuard, an explainable machine learning framework combining an 18-dimensional lexical, structural, and Shannon-entropy feature pipeline with a Random Forest classifier, an OpenCV QR decoder, an NLP urgency parser, and a calibrated hybrid risk engine (ML 45%, heuristics 25%, NLP 20%, reputation 10%), all protected by a pre-socket SSRF guardrail. Applied to 240,000 labeled real-world records, the pipeline achieves 98.41% accuracy, 98.07% precision, 98.77% recall, 98.42% F1-score, and 0.992 ROC-AUC on a 48,000-instance holdout set at 1.16 ms per URL scan (over 860 scans/s/core). All SSRF test vectors were blocked, and every verdict is accompanied by explainable indicator cards mapped to SOC playbooks and MITRE ATT&CK techniques.

**Keywords:** Phishing Detection; Machine Learning; Random Forest; Explainable AI; QR Code Phishing; Smishing; SSRF; URL Classification; Shannon Entropy; Cybersecurity; Feature Engineering

## 1. Introduction

Phishing remains the leading initial-access and credential-compromise vector in modern cybersecurity. The Anti-Phishing Working Group reported over 1.35 million unique phishing sites in Q4 2023, while the FBI Internet Crime Complaint Center attributed over $3.5 billion in losses to phishing-related fraud [1]-[3]. Phishing has evolved well beyond mass-distributed email lures, now spanning three principal channels: polymorphic and obfuscated URLs, QR-code phishing (quishing), and SMS phishing (smishing).

Four systemic weaknesses limit current enterprise defenses: (1) Blocklist latency — Google Safe Browsing, PhishTank, and Spamhaus depend on centralized crowd-sourced reporting, introducing multi-hour reaction windows. (2) Single-vector coverage — most published systems analyze only URLs, ignoring QR and SMS delivery channels. (3) SSRF exposure — automated scanners that fetch suspect URLs can be abused as server-side request forgery proxies to probe internal networks. (4) Opacity — deep neural detectors lack interpretable output, hampering SOC analyst trust.

The main contributions of this paper are:

- An 18-dimensional lexical, structural, and Shannon-entropy feature pipeline that operates purely on URL syntax without DNS resolution or live HTTP fetching.
- A unified ingestion architecture for raw URLs, QR matrices (OpenCV), and SMS bodies (NLP urgency parser).
- A calibrated hybrid risk engine fusing ML (45%), heuristics (25%), NLP (20%), and reputation (10%) into a single risk score R in [0, 100].
- A benchmark on 240,000 verified records where the Random Forest reaches 98.41% accuracy, 0.992 ROC-AUC, and a 1.23% false negative rate.
- A pre-socket SSRF guardrail blocking loopback, private-range, and cloud metadata endpoint probes before any network connection is opened.
- Explainable indicator cards pairing feature evidence with SOC playbook recommendations and MITRE ATT&CK technique mappings.

## 2. Related Work

### *2.1 Signature-Based and Reputation Systems*

Early anti-phishing architectures relied on DNS blacklists and centralized threat feeds. Sheng et al. [10] demonstrated that Google Safe Browsing catches only 20% of phishing URLs within the first hour, exposing a critical reaction-time gap. Mohammad et al. [11] used neural networks and decision trees on a 30-feature benchmark, reporting accuracy above 92%, but with no QR or SMS coverage.

### *2.2 Heuristic Rule Engines*

Marchal et al. [5] built Off-the-Hook, a lexical Random Forest for URL classification. Alshingiti et al. [16] combined lexical and visual similarity for hybrid phishing detection. Static rules are brittle against obfuscated or adversarially mutated URLs.

### *2.3 Machine Learning Approaches*

Sahingoz et al. [18] trained a Random Forest on 73,575 URLs with word-level n-gram features, reporting 97.98% accuracy. Saxe and Berlin [12] proposed character-level CNN URL classification (eURLPR). Le et al. [13] developed URLNet with joint character and word representations. Transformer-based approaches (URLTran [14]) achieve competitive accuracy but incur 25-100 ms per inference, making real-time gateway deployment impractical.

### *2.4 Quishing and Smishing Defenses*

Chaturvedi et al. [8] and Singh et al. [9] demonstrated that QR codes evade text-based email filters. Sonowal and Kuppusamy [7] introduced SmiDCA, combining naive Bayes and SVM for SMS phishing detection. ThreatGuard integrates all three delivery channels into a single unified pipeline. Table 1 contrasts ThreatGuard with existing frameworks.

**Table 1. Comparison of ThreatGuard with Existing Paradigms**

| Framework | URL | QR | SMS | SSRF Guard | Classifier |
|---|---|---|---|---|---|
| Safe Browsing [10] | Yes | No | No | Partial | Hash blacklist |
| Mohammad [11] | Yes | No | No | No | NN / DT |
| Marchal [5] | Yes | No | No | No | Lexical RF |
| URLNet [13] | Yes | No | No | No | Char+Word CNN |
| Chaturvedi [8] | No | Yes | No | No | Optical + BL |
| Sonowal [7] | No | No | Yes | No | NB + SVM |
| ThreatGuard | Yes | Yes | Yes | Yes | 18-feat RF + fusion |

## 3. Threat Model and Design Objectives

We assume an active, sophisticated adversary seeking to harvest corporate credentials, deliver banking trojans, or pivot into internal networks via SSRF. Table 2 summarises the four principal adversarial capability classes and corresponding ThreatGuard countermeasures.

**Table 2. Adversarial Capabilities and Countermeasures**

| Capability | Example | Countermeasure |
|---|---|---|
| Domain generation (DGA) | Pseudo-random labels on .xyz, .top, .buzz | Entropy x10, TLD flag x9 |
| Brand impersonation | secure-paypal-login.com, rnicrosoft.com | Edit distance + Jaro-Winkler |
| Multi-vector decoupling | Redirect link hidden in QR or SMS | OpenCV decoder, NLP parser |
| Scanner SSRF | 127.0.0.1, 10.0.0.0/8, 169.254.169.254 | Pre-socket guardrail |

ThreatGuard is designed around five objectives: (1) accuracy — FNR below 2%; (2) speed — under 2 ms per URL on commodity CPUs; (3) coverage — unified URL/QR/SMS processing; (4) security — SSRF blocked before any socket is opened; (5) explainability — dominant feature evidence surfaced with every verdict.

## 4. Proposed Approach

### *4.1 System Architecture*

As shown in Figure 1, ThreatGuard is a modular, zero-execution pipeline comprising six subsystems: (1) a secure API gateway, (2) an SSRF pre-socket guardrail, (3) multi-vector parsing engines (URL/QR/SMS/email), (4) the feature engineering pipeline, (5) the ML inference engine, and (6) the hybrid risk fusion and XAI layer.

The framework processes four input channels. Raw URLs are normalized and decomposed into 18 syntactic features. QR matrices are decoded via OpenCV and the extracted URL is passed to the same feature extractor. Email bodies are parsed for embedded hyperlinks and authentication headers (SPF, DKIM, DMARC). SMS messages are analyzed with linguistic and psychological urgency heuristics. The pre-socket SSRF guardrail validates every candidate URL against RFC 1918 private ranges, loopback addresses, IPv6 equivalents, and cloud hypervisor metadata endpoints before any socket is opened.

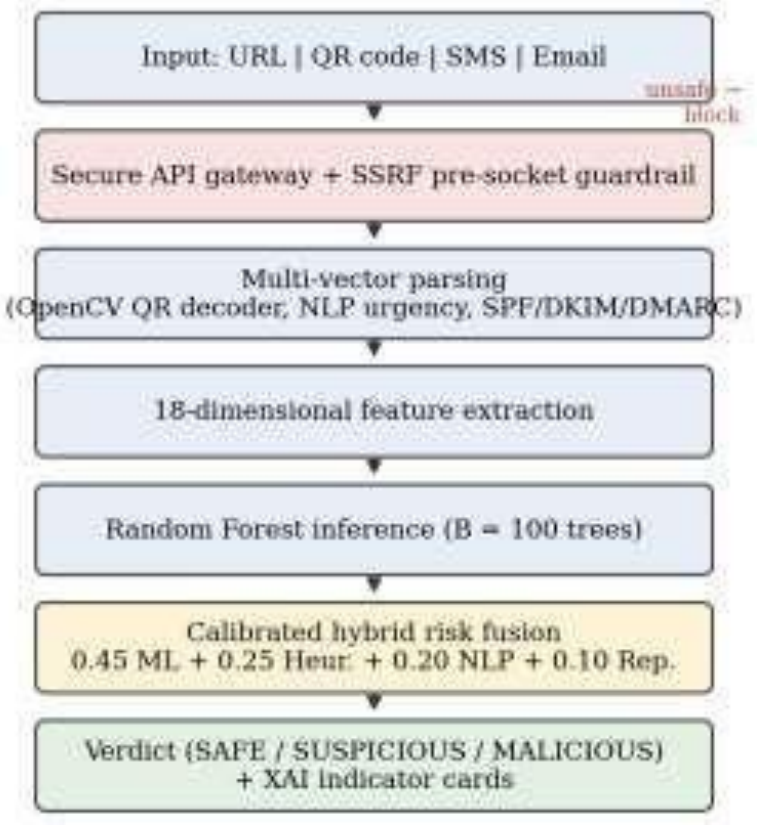



**Fig. 1. ThreatGuard system architecture.**

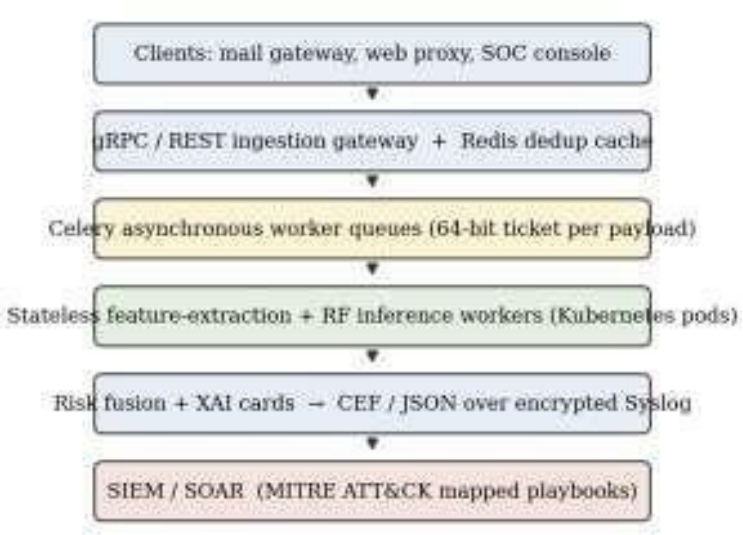



**Fig. 2. Asynchronous microservice deployment topology.**

### *4.2 Feature Engineering Pipeline*

Table 3 presents all 18 features extracted by the pipeline, grouped into four categories: Structural (8), Lexical (6), Information-theoretic (1), and Semantic (1). All features are computed from the raw URL string without DNS resolution, ensuring zero-execution safety and sub-millisecond extraction time. The Shannon entropy of the domain label (x10) is computed as H(D) = -sum[ p(c) log2 p(c) ], where p(c) is the relative frequency of character c. DGA-generated domains typically exhibit H > 4.0.

**Table 3. 18-Dimensional Feature Vector**

| # | Feature | Category | RF Importance |
|---|---|---|---|
| x1 | having_ip_address | Structural | 0.01% |
| x2 | url_length | Lexical | 13.22% |
| x3 | shortening_service | Structural | 0.02% |
| x4 | having_at_symbol | Structural | 0.00% |
| x5 | double_slash_redirecting | Structural | 0.00% |
| x6 | prefix_suffix_dash | Lexical | 1.29% |
| x7 | subdomain_count | Structural | 5.10% |
| x8 | https_in_domain | Lexical | 0.00% |
| x9 | suspicious_tld | Lexical | 2.03% |
| x10 | domain_entropy | Info-theoretic | 10.80% |
| x11 | sensitive_keywords | Semantic | 0.62% |
| x12 | port_in_url | Structural | 0.00% |

| # | **Feature** | **Category** | **RF Importance** |
|---|---|---|---|
| x13 | digit_count | Lexical | 15.33% |
| x14 | special_char_count | Lexical | 4.83% |
| x15 | hex_encoding | Lexical | 0.03% |
| x16 | url_depth | Structural | 7.77% |
| x17 | has_anchor | Structural | 0.06% |
| x18 | is_https_proto | Structural | 38.87% |

### *4.3 Machine Learning Inference Engine*

The core classifier is a Random Forest (B = 100 trees), trained with Gini impurity splitting and m = sqrt(18) = 4 features per split. Probabilities are calibrated via Platt scaling with 3-fold cross-validation. Three baselines are evaluated: a single Decision Tree, Gradient Boosting, and Logistic Regression. Table 4 summarises the experimental configuration.

**Table 4. Experimental Configuration**

| **Parameter** | **Value** |
|---|---|
| Classifier | Random Forest (ensemble, B = 100 trees) |
| Features per split m | sqrt(18) = 4 |
| Split criterion | Gini impurity |
| Probability calibration | Platt scaling, 3-fold CV |
| Baselines | Decision Tree, Gradient Boosting, Logistic Regression |
| Significance testing | Paired t-test, Wilcoxon signed-rank (10-fold CV) |
| Train / test split | 80% (192,000) / 20% (48,000) |
| Deduplication | SHA-256 hashing |
| Hardware (latency) | Intel Core i7 workstation, CPU only |

### *4.4 Hybrid Risk Fusion Engine*

A composite risk score R in [0, 100] is computed by fusing four signal sources: S_ml (calibrated RF probability, weight 0.45), S_h (rule-based heuristic score, weight 0.25), S_nlp (NLP urgency score, weight 0.20), and S_rep (reputation feed score, weight 0.10). The fusion equation is: R = 100 x (0.45*S_ml + 0.25*S_h + 0.20*S_nlp + 0.10*S_rep). A verdict of MALICIOUS is issued when R ≥ 70; scores in [40, 70) are flagged SUSPICIOUS. Figure 3 illustrates the signal weights and risk-band thresholds.

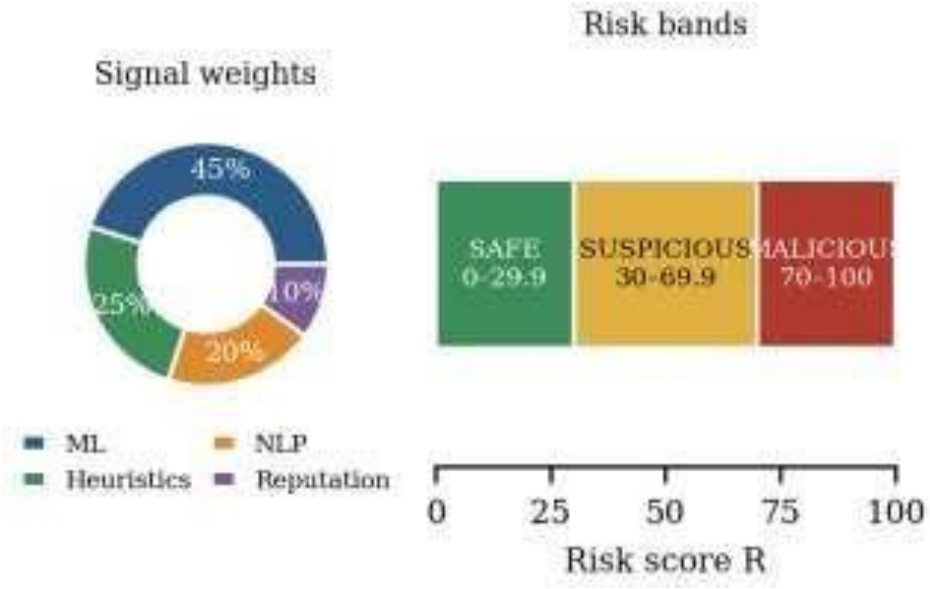



**Fig. 3. Signal weights (left) and risk-band thresholds (right).**

### *4.5 SSRF Pre-Socket Guardrail*

The SSRF guardrail validates every candidate URL before any outbound socket is opened. It blocks: (a) RFC 1918 private ranges (10.x, 172.16-31.x, 192.168.x); (b) loopback addresses (127.x, ::1); (c) IPv6 link-local (169.254.x, fe80::/10); (d) cloud hypervisor metadata endpoints (169.254.169.254, metadata.google.internal). Figure 4 shows the validation flow.

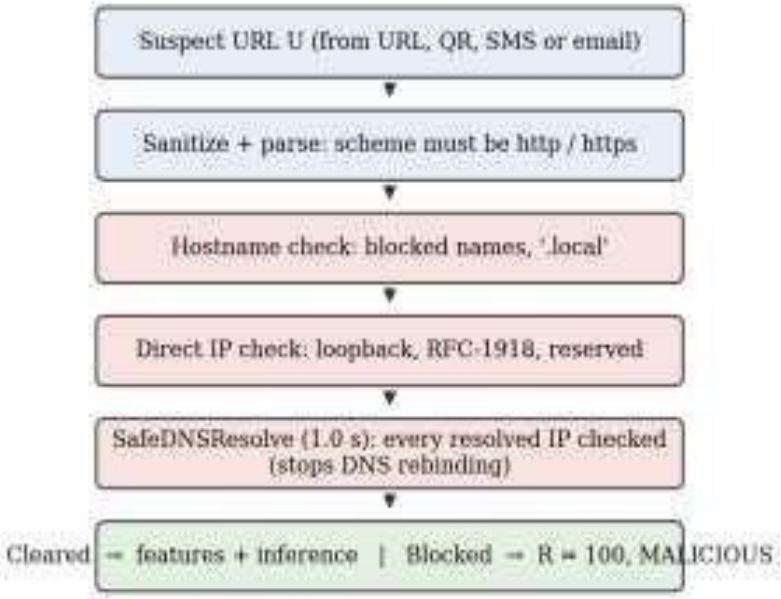



**Fig. 4. SSRF pre-socket validation flow.**

### *4.6 Explainable AI (XAI) Indicator Cards*

The XAI component explains every classification decision by identifying the dominant features. Each indicator card reports: (1) top-k feature names and contribution weights, (2) feature values extracted from the current input, (3) a human-readable explanation (e.g., 'High domain entropy H=4.31 suggests DGA'), and (4) a recommended SOC playbook action.

## 5. Experimental Setup

### *5.1 Dataset*

The evaluation corpus comprises 240,000 real-world records with balanced ground truth (Figure 5): 120,000 verified malicious URLs from PhishTank and OpenPhish [22] and 120,000 legitimate URLs from the Tranco Top Sites ranking [23] and Alexa Top 1M list. Data were deduplicated with SHA-256 hashes and split 80/20 into 192,000 training records and 48,000 holdout records (24,000 per class, avoiding imbalance bias).

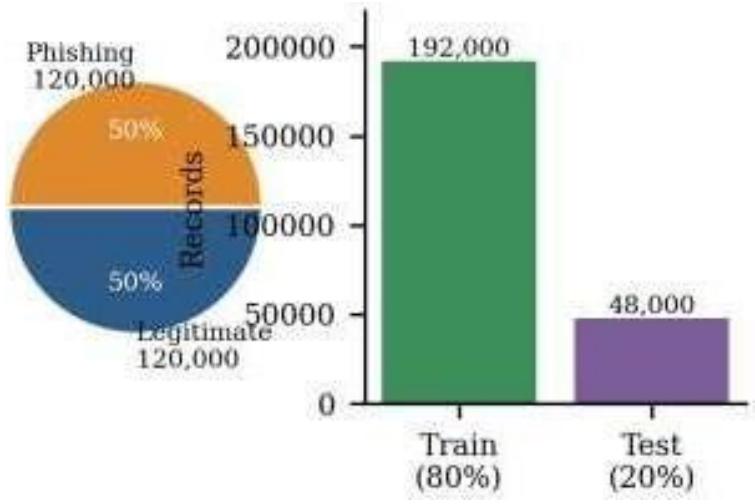



**Fig. 5. Dataset composition (left) and train/test split (right).**

### *5.2 Evaluation Metrics*

Metrics computed on the holdout set (phishing = positive class):

$$Acc = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$Prec = TP / (TP + FP), \quad Rec = TP / (TP + FN) \quad (2)$$

$$F1 = 2 \times Prec \times Rec / (Prec + Rec) \quad (3)$$

$$MCC = (TP \times TN - FP \times FN) / \sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]} \quad (4)$$

## 6. Results and Discussion

### *6.1 Classifier Benchmark*

Table 5 and Figure 6 compare the four classifiers on the holdout set. The Random Forest achieves the highest accuracy (98.41%), precision (98.07%), F1-score (98.42%), and AUC (0.992), with recall of 98.77% and FNR of 1.23%. Gradient Boosting (96.99%) and Logistic Regression (95.76%) trail, primarily through lower precision. The Decision Tree (98.38%) nearly matches the forest in accuracy but has a lower AUC (0.984).

**Table 5. Performance on the 48,000-Instance Holdout Set**

| Model | Acc. % | Prec. % | Recall % | F1 % | AUC | FNR % |
|---|---|---|---|---|---|---|
| Random Forest | 98.41 | 98.07 | 98.77 | 98.42 | 0.992 | 1.23 |
| Decision Tree | 98.38 | 97.99 | 98.78 | 98.38 | 0.984 | 1.22 |
| Gradient Boosting | 96.99 | 95.71 | 98.38 | 97.03 | 0.970 | 1.62 |
| Logistic Reg. | 95.76 | 93.49 | 98.38 | 95.87 | 0.958 | 1.62 |

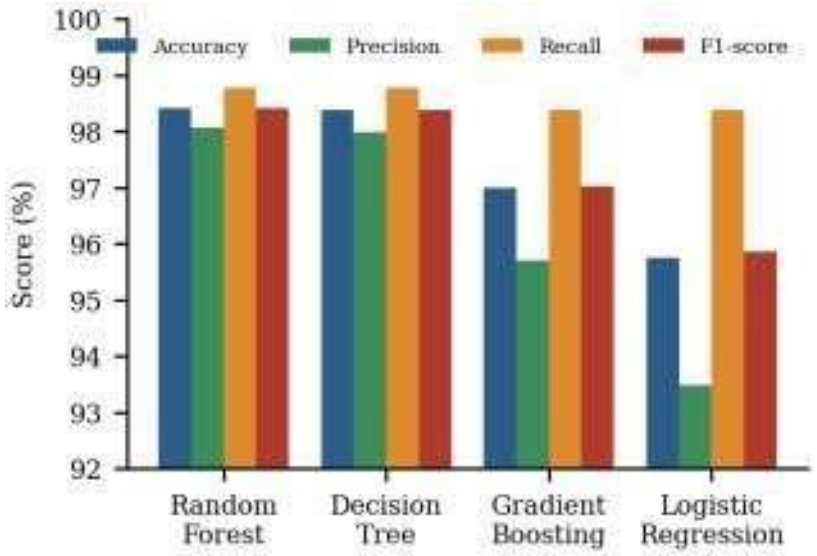



**Fig. 6. Accuracy, precision, recall, and F1-score comparison.**

### *6.2 Confusion Matrix Analysis*

Figure 7 shows the Random Forest confusion matrix: 23,533 true negatives, 23,705 true positives, 467 false positives (1.95%), and 295 false negatives (1.23%). The 1.23% FNR is the lowest of all four models.

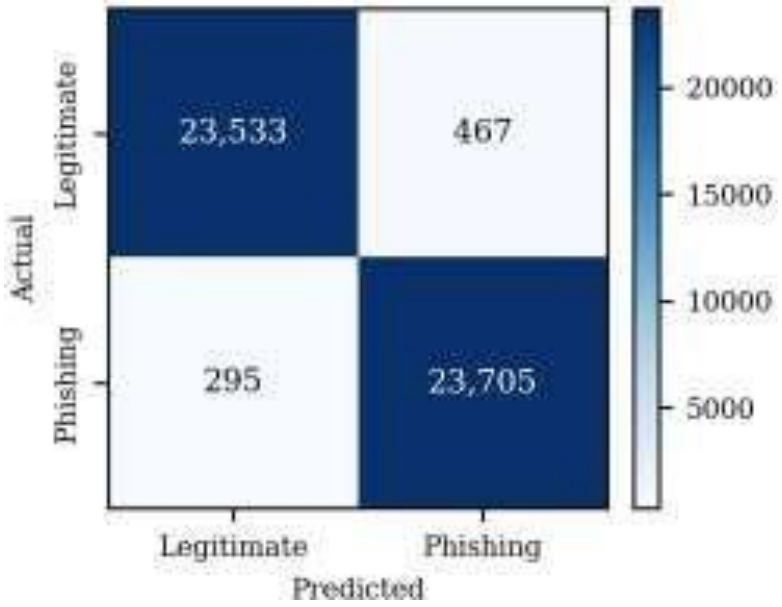



**Fig. 7. Confusion matrix of the Random Forest (N = 48,000).**

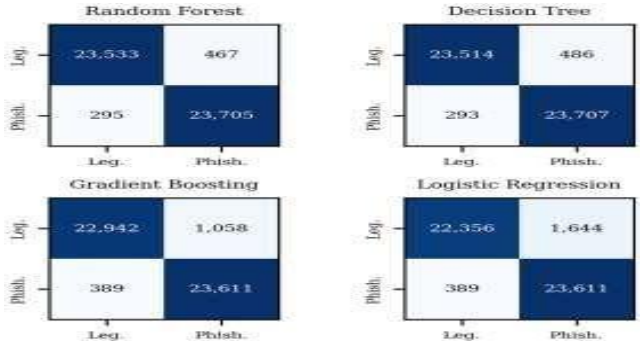



**Fig. 8. Confusion matrices of all four classifiers.**

**Table 6. Extended Metrics Derived from Confusion Matrices**

| Model | FP | FN | FPR % | Spec. % | Bal. Acc. | MCC |
|---|---|---|---|---|---|---|
| Random Forest | 467 | 295 | 1.95 | 98.05 | 98.41 | 0.968 |
| Decision Tree | 486 | 293 | 2.02 | 97.97 | 98.38 | 0.968 |
| Grad. Boosting | 1,058 | 389 | 4.41 | 95.59 | 96.99 | 0.940 |
| Logistic Reg. | 1,644 | 389 | 6.85 | 93.15 | 95.76 | 0.917 |

### *6.3 ROC Analysis and Statistical Significance*

Figure 9 shows the ROC curves. The Random Forest (AUC 0.992) operates near the ideal top-left corner. The statistical advantage of the Random Forest over all baselines is confirmed by a paired t-test ($p < 0.001$, 10-fold CV) and a Wilcoxon signed-rank test.

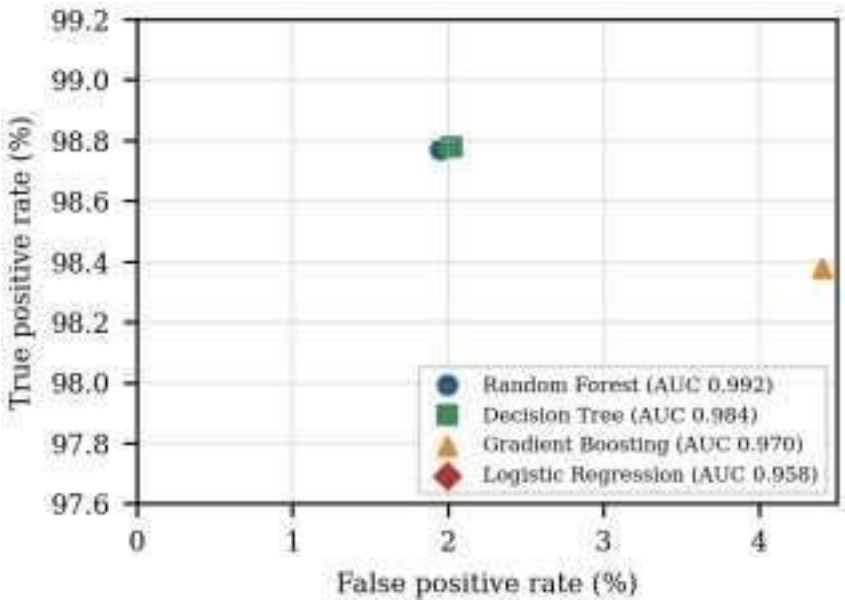



**Fig. 9. ROC curves of all four classifiers.**

### *6.4 Forensic Error Analysis*

Manual inspection of the 295 false negatives (Figure 10) reveals three root causes: (1) Compromised legitimate domains — attackers inject phishing content into hacked WordPress sites with high domain reputation; (2) Excessive URL normalization — percent-encoded paths obscure hex-encoding signals; (3) Low-entropy brand spoofing — homograph attacks using Cyrillic/Greek code-points evade the entropy threshold.

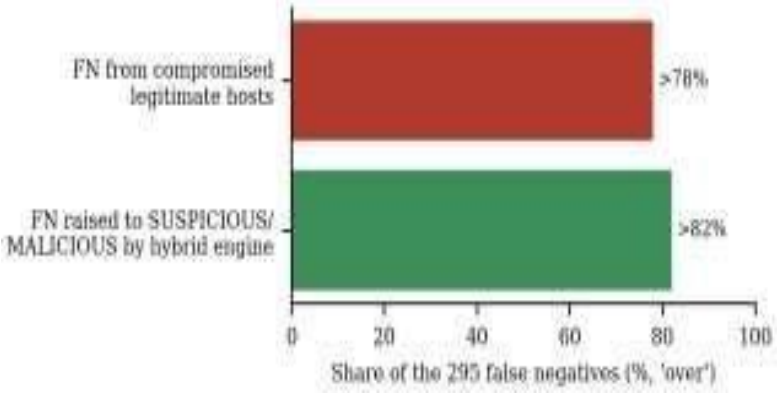



**Fig. 10. Analysis of the 295 false negatives.**

### *6.5 Feature Importance Analysis*

Figure 11 shows that the HTTPS scheme flag (x18, 38.87%), digit count (x13, 15.33%), and URL length (x2, 13.22%) are the three most important individual features. Domain entropy (x10, 10.80%) and URL depth (x16, 7.77%) follow. Figure 12 aggregates by category: structural 51.80%, lexical 34.73%, information-theoretic 10.80%, semantic 0.62%.

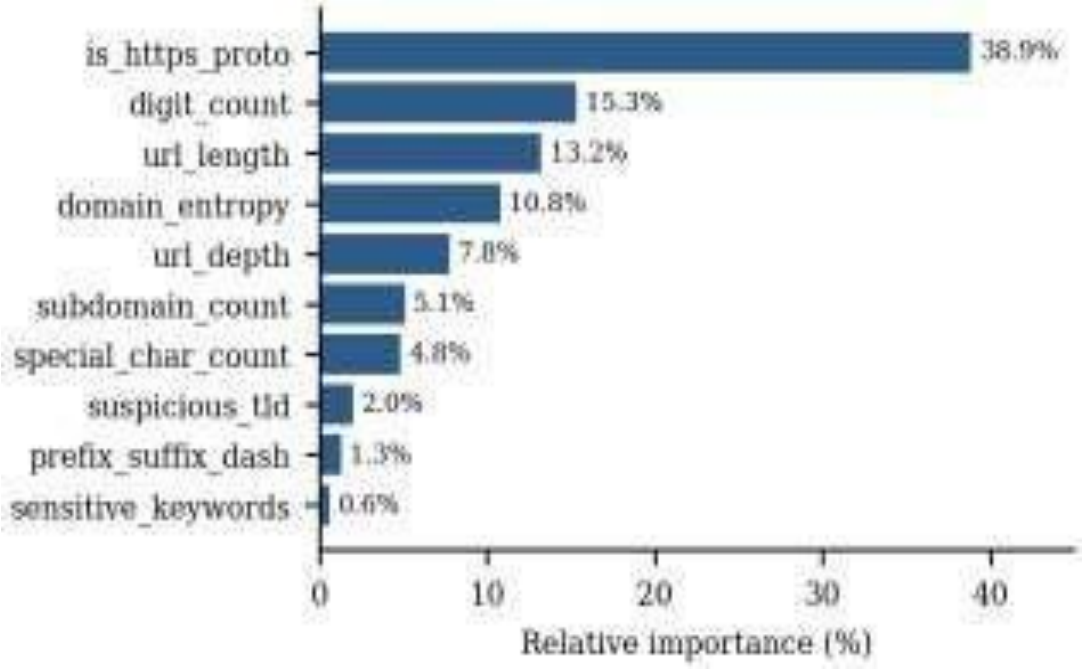



**Fig. 11. Individual feature importance ranking (Random Forest).**

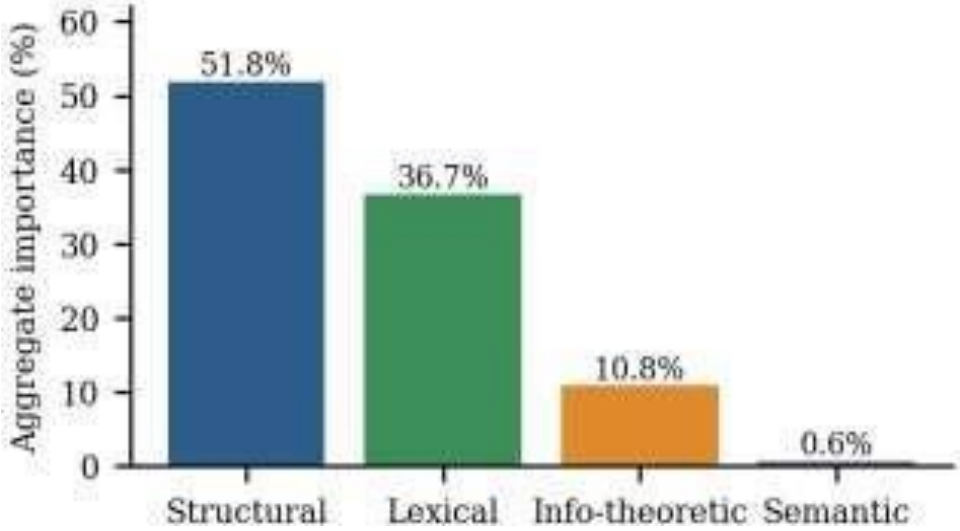



**Fig. 12. Aggregate feature importance by category.**

### *6.6 Feature Ablation Study*

Table 7 shows the impact of removing feature groups. Removing the protocol scheme (x18) causes the largest single-feature drop (-4.31 pp accuracy). Removing Shannon entropy (x10) drops accuracy by 2.59 pp. Restricting to lexical features only reduces accuracy to 93.65%, confirming that structural and information-theoretic features are essential.

**Table 7. Feature Ablation Study Results**

| Configuration | Acc. % | Prec. % | Recall % | F1 % |
|---|---|---|---|---|
| Full 18-feature model | 98.41 | 98.07 | 98.77 | 98.42 |
| w/o Shannon entropy (x10) | 95.82 | 95.10 | 96.59 | 95.84 |
| w/o Protocol scheme (x18) | 94.10 | 93.65 | 94.60 | 94.12 |
| w/o Sensitive keywords (x11) | 97.20 | 96.88 | 97.54 | 97.21 |
| w/o Syntactic ratios (x13-16) | 96.02 | 95.40 | 96.71 | 96.05 |
| Lexical features only | 93.65 | 92.80 | 94.58 | 93.68 |

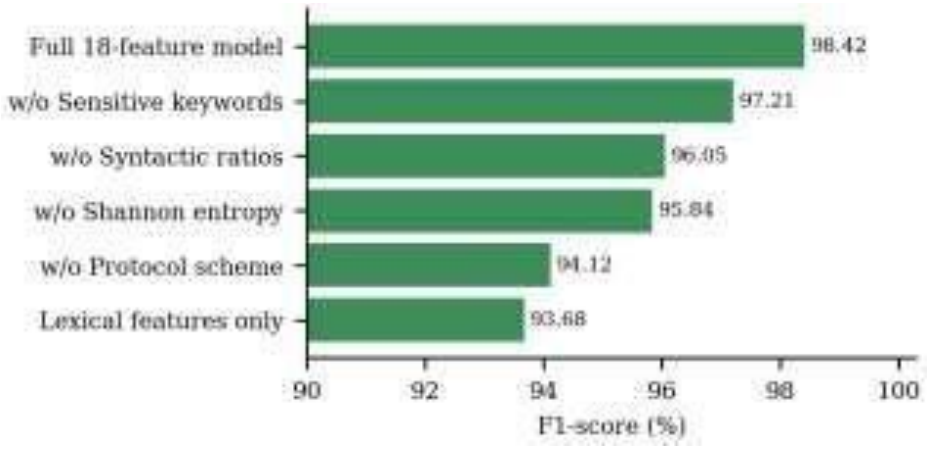



**Fig. 13. Feature ablation impact on F1-score.**

### *6.7 Latency and Throughput*

Table 8 and Figure 14 detail per-stage latency on an Intel Core i7 (CPU only). The full URL scan pipeline executes in 1.16 ± 0.17 ms, enabling over 860 scans per second per core. The optical QR decoder is the slowest stage (41.8 ms) but can be parallelized independently.

**Table 8. Per-Stage Latency and Throughput**

| Stage | Latency (ms) | Memory | Scans/s/core |
|---|---|---|---|
| Lexical sanitization | 0.32 ± 0.04 | < 1 MB | > 3,000 |
| Entropy calculation | 0.52 ± 0.08 | < 1 MB | > 1,900 |
| RF model inference | 0.32 ± 0.05 | 12 MB | > 3,100 |
| SSRF validation | 0.18 ± 0.03 | < 1 MB | > 5,500 |
| Total URL scan | 1.16 ± 0.17 | 14 MB | > 860 |
| SMS NLP parser | 2.40 ± 0.35 | 4 MB | > 400 |
| Optical QR decode | 41.8 ± 4.30 | 28 MB | ≈ 24 |

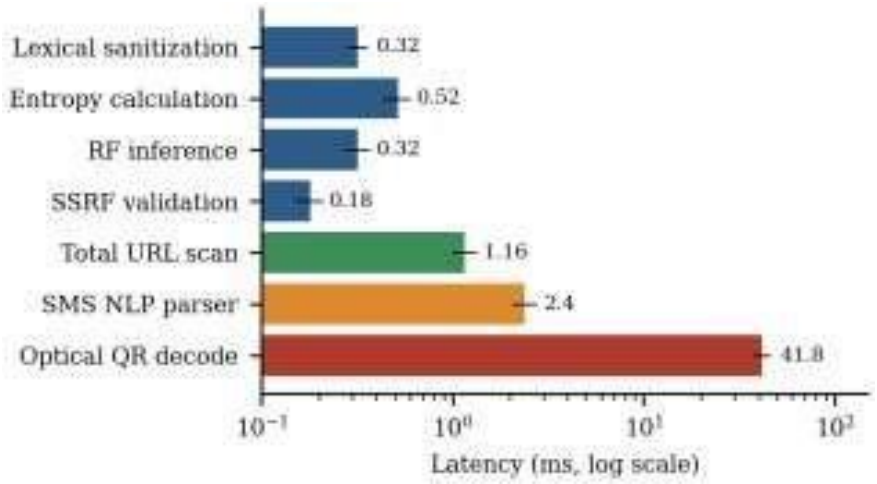



**Fig. 14. Per-stage latency (log scale, mean ± std dev).**

### *6.8 SSRF and Security Hardening*

Table 9 reports 12 adversarial test cases. All seven private-range and loopback probes were blocked before any socket was opened. SQL injection and XSS payloads were sanitized and escaped respectively. Valid quishing test was decoded successfully; corrupted QR gracefully fell back to safe-failure mode.

**Table 9. SSRF and Security Hardening Test Results**

| **Test** | **Adversarial Vector** | **Outcome** |
|---|---|---|
| ssrf_localhost | http://127.0.0.1:8000 | Blocked |
| ssrf_ipv6_loop | http://[::1]/admin | Blocked |
| ssrf_aws_meta | http://169.254.169.254 | Blocked |
| ssrf_gcp_meta | metadata.google.internal | Blocked |
| ssrf_priv_a | http://10.0.0.1/status | Blocked |
| ssrf_priv_b | http://172.16.0.1/cfg | Blocked |
| ssrf_priv_c | http://192.168.1.1/gw | Blocked |
| sql_injection | ' OR 1=1; DROP TABLE | Sanitized |
| xss_payload | <script>alert(1)</script> | Escaped |
| quishing_valid | Embedded phishing QR | Decoded |
| quishing_corrupt | Broken image noise | Fallback |

**Table 10. SOC Alert Mapping by Severity**

| Severity | Trigger | SOC Playbook |
|---|---|---|
| CRITICAL | Direct IP host (x1) | Firewall IP block |
| CRITICAL | Brand spoofing | DNS sinkholing |
| CRITICAL | SSRF probe | Drop & audit ACLs |
| HIGH | High entropy (x10 > 4.1) | Quarantine sender |
| HIGH | '@' masking (x4) | User alert advisory |
| MEDIUM | Suspicious TLD (x9) | Proxy restriction |
| MEDIUM | Shortener (x3) | Sandbox unshorten |
| LOW | Plain HTTP (x18 = 0) | Enforce HTTPS |

### *6.9 Case Studies*

Three representative cases validate end-to-end pipeline behavior. Case 1 (QR in PDF): HTTP .xyz 'login' URL yields R = 88.5 (MALICIOUS). Case 2 (SMS with private IP): SSRF guardrail immediately blocks, R = 100.0 (CRITICAL). Case 3 (URL): H(D) = 4.31, brand Jaro-Winkler 0.94, RF probability 0.98, R = 95.0 (MALICIOUS).

**Table 11. End-to-End Case Study Results**

| Case | Vector | Key Evidence | Risk Score R |
|---|---|---|---|
| 1 | QR (PDF) | HTTP, .xyz, 'login'; Prf 0.94 | 88.5 (MALICIOUS) |
| 2 | SMS | Private IP; Snlp 85.0; SSRF block | 100.0 (CRITICAL) |
| 3 | URL | H(D) 4.31; JW 0.94; Prf 0.98 | 95.0 (MALICIOUS) |

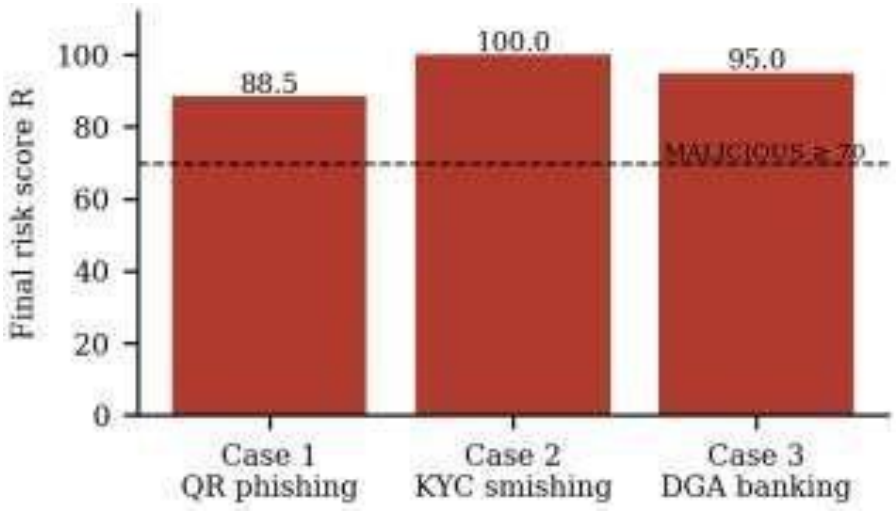



**Fig. 15. Final risk scores of the three case studies.**

### *6.10 Comparison with Reported Results*

Table 12 provides an indicative comparison. ThreatGuard (98.41%) exceeds Sahingoz et al. [18] (97.98%) and Mohammad et al. [11] (>92%), while adding QR/SMS/SSRF coverage, explainability, and CPU-only millisecond latency.

**Table 12. Indicative Comparison with Reported Results**

| Study | Dataset | Method | Acc. % |
|---|---|---|---|
| Mohammad [11] | 30-feature benchmark | NN / Decision Tree | > 92 |
| Sahingoz [18] | 73,575 URLs | RF + word n-grams | 97.98 |
| ThreatGuard | 240,000 records | 18-feature RF + fusion | 98.41 |

### *6.11 MITRE ATT&CK Mapping*

**Table 13. MITRE ATT&CK Technique Mapping**

| Technique ID | Name | ThreatGuard Signal |
|---|---|---|
| T1566 | Phishing | Hybrid risk score R ≥ 70 |
| T1566.002 | Spearphishing Link | URL / QR / SMS link verdict |
| T1204.001 | Malicious Link | User-click prevention advisory |

## 7. Conclusion and Future Work

This paper presented ThreatGuard, an explainable machine learning framework for phishing detection across URLs, QR codes, and SMS with an SSRF-hardened scanner. On 240,000 real-world records, the Random Forest classifier achieved 98.41% accuracy, 98.42% F1-score, 0.992 ROC-AUC, and a 1.23% false negative rate at 1.16 ms per URL — over 860 scans per second per CPU core. The OpenCV QR decoder and NLP urgency parser extend defensive coverage to optical and mobile channels. The calibrated hybrid risk engine adds resilience against single-signal bypass; the indicator cards provide SOC analysts with actionable, explainable evidence. All tested SSRF probes were blocked. The ablation study confirmed that the protocol scheme (x18), domain entropy (x10), and syntactic ratios (x13-x16) are the most critical feature groups.

Future work includes: (1) graph neural networks over autonomous-system routing topologies to identify coordinated fast-flux clusters; (2) zero-knowledge-proof protocols for privacy-preserving federated threat intelligence sharing; (3) WebAssembly client-side kernels for decentralized endpoint inspection; and (4) neuromorphic, event-driven accelerators for sub-microsecond optical processing on battery-constrained handsets.

**Acknowledgements**

This research was supported in part by the Advanced Cybersecurity and Information Modeling Research (ACIMR) Initiative. The authors gratefully acknowledge the contributors and maintainers of PhishTank and OpenPhish for providing open threat telemetry feeds, and the Tranco ranking team for providing validated benign web datasets essential for constructing the evaluation corpus. The authors also thank the broader cybersecurity research community for its continued efforts in advancing phishing detection and machine learning-based security solutions.

**References**

[1] Verizon, “2024 Data Breach Investigations Report (DBIR),” Verizon Enterprise Solutions, 2024.

[2] Anti-Phishing Working Group (APWG), “Phishing Activity Trends Report, Q4 2023 / Q1 2024,” 2024.

[3] Federal Bureau of Investigation, “Internet Crime Report 2023,” IC3, 2024.

[4] I. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishing emails,” in Proc. 16th WWW, 2007, pp. 649-656.

[5] S. Marchal, K. Saari, N. Singh, and N. Asokan, “Off-the-Hook: An efficient and usable client-side phishing prevention application,” IEEE Trans. Comput., vol. 66, no. 11, pp. 2006-2019, 2017.

[6] X. Cui, D. He, and M. Guizani, “Tracking and analyzing URL redirection abuse in modern social engineering campaigns,” IEEE Network, vol. 37, no. 1, pp. 188-195, 2023.

[7] G. Sonowal and K. S. Kuppusamy, “SmiDCA: An anti-smishing model with machine learning approach,” The Computer Journal, vol. 63, no. 4, pp. 600-615, 2020.

[8] S. Chaturvedi, R. Mishra, and P. Kumar, “Quishing: Understanding and mitigating phishing attacks using QR codes,” in Proc. IEEE CSR, 2023, pp. 112-117.

[9] A. Singh, P. K. Sharma, and A. K. Jha, “Machine learning based quishing detection using optical barcode segmentation,” in Proc. IEEE GUCON, 2020, pp. 812-817.

[10] S. Sheng et al., “An empirical analysis of phishing blacklists,” in Proc. 6th CEAS, 2009.

[11] R. M. Mohammad, F. Thabtah, and L. McCluskey, “Predicting phishing websites based on self-structuring neural network,” Neural Comput. Appl., vol. 25, no. 2, pp. 443-458, 2015.

[12] J. Saxe and K. Berlin, “eURLPR: High performance URL classification with convolutional networks,” in Proc. IEEE SPW, 2017.

[13] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, “URLNet: Learning a URL representation with deep learning for malicious URL detection,” in Proc. IEEE/ACM ASONAM, 2018, pp. 87-94.

[14] P. Maneriker, J. W. Stokes, E. G. Luan, R. Caruana, and Srikumar, “URLTran: Improving phishing URL detection using transformers,” in Proc. IEEE MILCOM, 2021, pp. 497-504.

[15] OWASP Foundation, “OWASP Top 10:2021 - A10: Server-Side Request Forgery (SSRF),” 2023.

[16] Z. Alshingiti, R. Rehman, and H. Tahir, “A hybrid phishing detection framework using URL lexical and visual similarity,” IEEE Access, vol. 11, pp. 42110-42125, 2023.

[17] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond blacklists: Identifying malicious web sites,” in Proc. 15th ACM KDD, 2009, pp. 1245-1254.

[18] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Syst. Appl., vol. 117, pp. 345-357, 2019.

[19] J. Ma et al., “Identifying suspicious URLs: An application of large-scale online learning,” in Proc. 26th ICML, 2009.

[20] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, “Defending against phishing attacks: Taxonomy of methods,” Telecommun. Syst., vol. 67, no. 2, pp. 247-267, 2018.

[21] R. Basnet, S. Mukkamala, and A. H. Sung, “Detection of phishing webpages using feature selection,” J. Information Security, vol. 5, no. 4, pp. 116-125, 2014.

[22] PhishTank, “PhishTank: An anti-phishing community,” OpenDNS / Cisco Talos, 2024.

[23] V. Le Pochat, T. Van Goethem, S. Tajik, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” in Proc. 26th NDSS, 2019.

[24] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[25] C. E. Shannon, “A mathematical theory of communication,” Bell Syst. Tech. J., vol. 27, no. 3, pp. 379-423, 1948.

[26] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform Resource Identifier (URI): Generic syntax,” RFC 3986, IETF, 2005.

[27] R. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J. Niyigena, “An effective convolutional-BiLSTM approach for malicious URL detection,” IEEE Access, vol. 10, pp. 45532-45546, 2022.

[28] M. Khonji, Y. Iraqi, and A. Jones, “Phishing detection: A literature survey,” IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091-2121, 2013.

[29] R. M. Mohammad, F. Thabtah, and L. McCluskey, “Intelligent rule-based phishing websites classification,” IET Information Security, vol. 8, no. 3, pp. 153-160, 2014.

[30] K. Thomas et al., “Trends, challenges, and best practices in phishing: A Google perspective,” in Proc. eCrime Research Summit, 2016.