International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Bridging Accuracy and Deployability: A Dual-Model Architecture for Real-Time Fake and Phishing Website Detection

Pruthviraj Pawar
Dept. of Computer Science
MIT-ADT University
Pune, India
pruthvirajpawar964@gmail.com

Ganesh Mule
Dept. of Computer Science
MIT-ADT University
Pune, India
ganeshmule8021@gmail.com

Sarthak Kshirsagar
Dept. of Computer Science
MIT-ADT University
Pune, India
sarthakkshirsagar6155@gmail.com

Alok Khade
Dept. of Computer Science
MIT-ADT University
Pune, India
alokkhade0024@gmail.com

Prof. Sharon Christa
Dept. of Computer Science
MIT-ADT University
Pune, India
sharon.christa@mituniversity.edu.in

***Abstract*—Fake websites that imitate trusted brands to harvest credentials, most commonly phishing sites, are difficult to stop with conventional blocklists because campaigns rotate through disposable infrastructure within hours. This paper presents a dual-model detection architecture that achieves a 96.7% F1-score with a 1.8% false positive rate while maintaining sub-80 ms response latency at the 95th percentile and throughput exceeding 2,400 classifications per second. The system combines gradient boosting for rapid primary inference with a character-level neural network for refined secondary analysis of ambiguous cases. Feature extraction spans 82 discriminative signals across URL lexical patterns, DNS resolution metadata, TLS certificate properties, and DOM structural indicators. Confidence-based early termination routes approximately 82% of requests through the fast path, while asynchronous enrichment prevents network delays from blocking decisions. Temporal validation on chronologically partitioned datasets prevents the optimistic bias that arises from campaign overlap between training and test data. Ablation experiments quantify individual feature contributions, and systematic error analysis reveals misclassification patterns that inform future enhancements. The framework supports inline deployment across browser extensions, enterprise web gateways, and email security platforms.**

***Index Terms*—Fake website detection, phishing classification, URL analysis, ensemble learning, convolutional networks, latency optimization, threat detection**

I. Introduction

Deceptive credential-harvesting websites continue to cause substantial financial and informational losses for organizations worldwide. Industry monitoring by the Anti-Phishing Working Group documents a persistent escalation in attack sophistication, with threat actors increasingly exploiting trusted cloud platforms and recognized brand identities [1]. Complaint data from the Federal Bureau of Investigation attributes billions of dollars in annual damages to phishing-enabled fraud [2]. Breach investigations by Verizon consistently identify credential theft through social engineering as a dominant initial access mechanism [3], and European threat assessments likewise rank phishing among the challenges demanding sustained defensive innovation [4].

Throughout this paper, a *fake website* is a site that imitates a legitimate brand or service in order to deceive visitors into revealing credentials, payment details, or other sensitive data. Phishing sites are the dominant and best-documented instance of this threat, and the labeled data used in our experiments (Section VIII) consists of verified phishing URLs and legitimate sites.

Blocklist-based protection was adequate when malicious infrastructure persisted for extended periods, but attackers have adapted. Domains now rotate within hours rather than weeks, fast-flux architectures obscure hosting locations, and URL shortening services mask terminal destinations [12]. Security vendors frequently list a resource only after the attacker has already migrated elsewhere [10].

The Deployment Divide. Scholarly investigations regularly report impressive benchmark performance yet struggle to translate it into operational systems. Architectures that reach peak accuracy demand computational resources incompatible with the latency budgets of inline traffic inspection, while lightweight alternatives that satisfy those budgets sacrifice detection capability. Organizations therefore face an uncomfortable trade-off between protection strength and user-experience quality.

This work bridges the deployment divide through the following contributions:

- **Multi-modal feature synthesis.** Features from four complementary modalities, selected through iterative evaluation that prioritizes discrimination per millisecond of extraction overhead.
- **Tiered inference.** Rapid gradient boosting paired with targeted neural refinement for uncertain predictions.

- **Deployment-conscious engineering.** Sustained throughput beyond 2,400 URLs per second with tail latency under 80 milliseconds.
- **Temporal partitioning.** Evaluation protocols that prevent optimistic bias from training–testing campaign overlap.

II. Threat Landscape and Security Model

A. Attack Characteristics

Contemporary phishing extends well beyond simple credential requests. Attackers construct elaborate deception chains that incorporate redirect sequences, dynamically generated landing pages, and victim profiling for targeted payload selection [8]. Infrastructure increasingly leverages compromised legitimate properties, which inherit established reputation signals.

B. Adversary Capabilities

The threat model covers attackers who employ diverse evasion strategies:

Structural manipulation exploits URL composition through excessive subdomain nesting, misleading path segments that embed brand terminology, percent-encoded characters, and shortening services that obscure destinations.

Visual confusion registers lookalike domains that substitute Unicode characters resembling their ASCII equivalents within internationalized names.

Reputation exploitation routes victims through redirect chains that traverse compromised sites before reaching attacker-controlled resources.

Temporal cloaking delivers benign content to automated scanners while presenting malicious payloads to genuine visitors [29].

Infrastructure cycling rapidly rotates hosting across fresh domains and geographic regions, outpacing blocklist propagation.

C. Adversary Knowledge Assumptions

A gray-box adversary model applies throughout this work. Attackers know the common detection strategies used across the security industry but have no access to the model parameters, trained weights, or decision thresholds of the proposed system. This reflects realistic operational conditions, in which threat actors can probe defensive behavior through test submissions but cannot inspect classifier internals.

D. Defender Constraints

Detection systems operate from different network vantage points, each with distinct constraints. Browser extensions intercept requests client-side with minimal computational resources. Enterprise gateways inspect traffic at network boundaries under strict latency budgets. Email platforms scan embedded links before message delivery.

Defenders also operate under partial visibility. DNS resolution metadata and TLS certificate properties remain accessible for most targets, but full page content is unavailable in numerous scenarios, including TLS inspection restrictions, server-side rate limiting, and cloaking techniques that serve different content depending on request characteristics.

TABLE I
Positioning Against Representative Systems

Approach	Signals	Model	RT	Temp.
URLNet [13]	URL	CNN	×	×
Phishpedia [17]	Visual	CNN	×	×
Ma et al. [9]	Lex+Host	Ensemble	✓	×
CANTINA [6]	Content	Heuristic	×	×
Proposed	Multi	Hybrid	✓	✓

RT = real-time optimized. Temp. = temporal evaluation.

E. Performance Requirements

Operational deployment mandates concrete thresholds: classification within 200 milliseconds for 95% of requests; sustained throughput exceeding 1,000 URLs per second per processing node; and a false positive rate below 2% when configured for 95% threat-detection sensitivity.

III. Prior Art and Positioning

A. Reactive Blocklist Methods

Early countermeasures maintained databases of confirmed malicious indicators. Whittaker et al. described automated pipelines that populate Safe Browsing lists [10], and Seifert et al. explored heuristics derived from observable page properties [5]. Blocklist approaches remain valuable but cannot address threats that have not yet been listed.

B. Discriminative Learning

Machine learning enabled proactive identification. Abu-Nimeh et al. examined classical algorithms applied to phishing corpora [7]. PILFER demonstrated email-based detection through systematic feature engineering [11]. CANTINA contributed content analysis that uses term frequency to detect semantic inconsistencies [6]. Ma et al. extended URL classification beyond blocklists by using lexical and hosting characteristics [9].

C. Representation Learning

Deep architectures enabled end-to-end processing. URLNet pioneered character-level convolution combined with word embeddings [13]. Sahingoz et al. benchmarked several classical classifiers on lexical URL features [14], and Do et al. reviewed deep learning taxonomies for phishing detection [16].

Visual similarity provides complementary capabilities. Phishpedia detects brand impersonation through logo recognition [17]. Such methods are effective against convincing replicas hosted on domains with no negative reputation history.

D. Adaptation Mechanisms

Adversarial dynamics induce concept drift that degrades historical models. Gama et al. surveyed adaptation strategies [23], and ADWIN enables responsive windowing that adjusts to detected distributional changes [24].

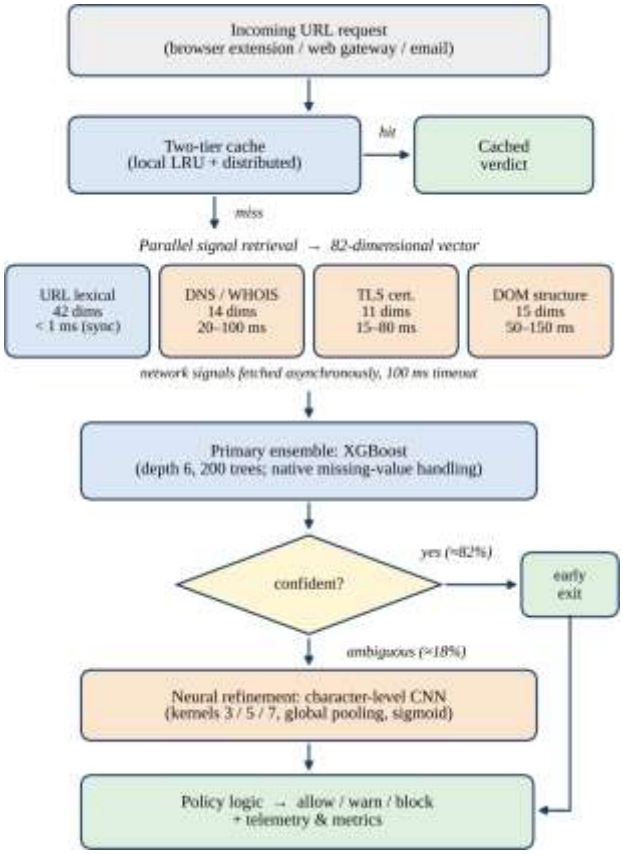


Fig. 1. Processing pipeline with hybrid inference and confidence routing. Lexical features are computed synchronously; network signals are fetched asynchronously under a 100 ms timeout.

E. Comparative Analysis

Table I positions this work relative to representative prior systems. To our knowledge, few prior systems jointly address multi-modal accuracy, strict real-time constraints, and temporal generalization within a single deployable architecture; prior approaches typically address these objectives in isolation, which limits their practical applicability.

IV. System Architecture

A. Component Organization

The detection pipeline comprises cooperating subsystems for signal extraction, model inference, decision orchestration, and telemetry collection. Fig. 1 illustrates the information flow.

B. Processing Sequence

Incoming URLs first consult a two-tier cache that combines local storage with distributed backends. Cache hits return prior verdicts within microseconds, and misses trigger parallel signal retrieval.

Lexical analysis completes synchronously from the submitted string. Network queries execute asynchronously with aggressive timeouts so that external delays cannot block classification. The assembled vector passes first to the boosting ensemble; predictions that exceed the confidence thresholds terminate immediately, and ambiguous cases are routed to neural refinement. Verdicts feed the enforcement and logging subsystems. Algorithm 1 summarizes this procedure.

Algorithm 1 Confidence-routed inference

Require: URL u; confidence thresholds $\tau_1 < \tau_2$ (calibrated on validation data)
Ensure: verdict (legitimate / fake)

```
if cache contains u then
    return cached verdict
end if
x_lex ← LexicalFeatures(u)                          ▷ synchronous
x_net ← AsyncFetch(DNS/WHOIS, TLS, DOM; timeout = 100 ms)
x ← Assemble(x_lex, x_net)                          ▷ missing values flagged
p ← XGBoost(x)
if p ≤ τ1 or p ≥ τ2 then
    verdict ← threshold(p)
else
    verdict ← NeuralRefine(u, p)                    ▷ character-level CNN
end if
store verdict in cache (class-dependent TTL); return verdict
```

TABLE II
Signal Distribution Across Categories

Category	Dimensions	Overhead
URL Lexical	42	<1 ms
DNS/WHOIS	14	20–100 ms
TLS Certificate	11	15–80 ms
DOM Structure	15	50–150 ms
Combined	**82**	**Variable**

V. Signal Engineering

A. Selection Methodology

Signals were selected through iterative ablation that prioritizes discrimination per millisecond of extraction overhead. In addition, temporal stability evaluation reduced sensitivity to short-lived campaign artifacts that might otherwise introduce spurious correlations. Table II summarizes the dimensionality of each category.

B. Lexical Indicators

String analysis captures obfuscation patterns without any network interaction. Measurements span component lengths, character distributions, entropy, tokenization statistics, subdomain depth, and pattern detection for IP literals, encoding artifacts, and embedded brand keywords [14].

C. Network Metadata

DNS characteristics reveal infrastructure maturity. Domain age correlates strongly with legitimacy when WHOIS permits access, record indicators note the presence of MX, NS, and

SPF records, and TTL values below typical thresholds suggest evasion tactics. Privacy regulations affect approximately 38% of WHOIS queries [28]. Missing values receive median imputation accompanied by explicit missingness indicators, which preserve information about data-availability patterns.

D. Certificate Properties

TLS metadata provides trust signals despite near-universal HTTPS adoption. Issuer categorization distinguishes automated authorities from commercial providers, and validity metrics, SAN counts, and protocol indicators contribute additional dimensions [12].

E. Content Structure

DOM analysis extracts behavioral indicators when page retrieval succeeds. Form inspection identifies password fields and cross-origin submissions, resource ratios measure external dependencies, and JavaScript scoring detects obfuscation through entropy and function analysis.

VI. Classification Models

A. Primary Ensemble

XGBoost serves as the primary inference engine because of its favorable latency–accuracy characteristics [18]. Gradient boosting constructs additive ensembles of shallow trees, each correcting the residuals of its predecessors. Tree traversal requires only comparison operations, so the inference step itself is inexpensive and needs no specialized hardware; end-to-end latency is dominated by signal retrieval (Section IX-D). Configuration: maximum depth 6, 200 boosting iterations, learning rate 0.1, L2 regularization weight 1.0, and column subsampling 0.8. Native missing-value handling accommodates partial signal availability.

B. Neural Refinement

Character-level convolution provides secondary analysis for uncertain cases. The architecture embeds 95 printable characters into 128 dimensions. Three parallel branches with kernels spanning 3, 5, and 7 positions each produce 128 feature maps, and global pooling aggregates the spatial dimension. Dense layers with dropout precede a sigmoid output. Training uses the Adam optimizer [19] with early stopping on validation performance.

C. Architecture Rationale

Transformer architectures [20], [21] achieve marginally higher accuracy but impose prohibitive deployment costs: BERT-scale models demand 400 MB+ of memory and 45 ms+ of inference latency, which is incompatible with browser-extension constraints and gateway throughput requirements.

The neural component requires approximately 3.4 million parameters, versus roughly 110,000 effective decision nodes within the boosting ensemble. This difference explains the order-of-magnitude latency gap despite comparable predictive performance, and it justifies the hybrid design that routes most traffic through the faster path.

VII. Deployment Engineering

A. Caching Architecture

Two-tier caching substantially reduces redundant computation. A local LRU store serves frequently accessed URLs with sub-millisecond overhead, while distributed backends provide cross-node consistency and persistence. Differentiated TTL policies apply shorter retention to threat verdicts and longer retention to legitimate classifications, balancing freshness against cache efficiency.

B. Asynchronous Retrieval

Network-dependent signals are retrieved concurrently with an aggressive 100 ms timeout. Timeout expiration triggers graceful degradation to lexical-only classification, which affects roughly 8% of production traffic.

C. Confidence Routing

Early-exit thresholds calibrated on validation data route approximately 82% of requests directly from the boosting stage to the output, reserving neural inference for genuinely ambiguous cases in which additional analysis provides a meaningful accuracy improvement.

D. Compression Techniques

Browser deployment packages the ensemble in ONNX format executed via a WebAssembly runtime, totaling 3.2 MB including weights. Quantization of the neural model reduces its footprint by 60% with less than 0.3% accuracy degradation.

VIII. Experimental Protocol

A. Dataset Construction

Phishing samples derive from verified PhishTank entries [26] spanning January through December 2023, totaling 67,842 unique URLs after deduplication. Legitimate samples combine Tranco rankings [15] with stratified Common Crawl sampling [25], totaling 89,156 URLs.

Temporal partitioning allocates January–August to training (103,284 URLs), September–October to validation (21,156 URLs), and November–December to testing (32,558 URLs). The test set has a 1:5 phishing-to-legitimate ratio, which approximates realistic production traffic composition.

B. Infrastructure Specification

Experiments run on Intel Xeon E5-2680 v4 processors (14 cores, 2.4 GHz) with 64 GB RAM and NVIDIA Tesla V100 acceleration. Software versions are Python 3.10, XGBoost 1.7.4, PyTorch 2.1.0, and scikit-learn 1.3.0. A fixed random seed (42) ensures reproducibility across data partitioning, weight initialization, and stochastic training.

C. Comparison Systems

We compare against: a blocklist lookup frozen at the training cutoff; a Random Forest (200 trees, maximum depth 10); a linear SVM with standardization ($C = 1.0$); and a character-level CNN matching the URLNet architecture [13].

TABLE III
CLASSIFICATION PERFORMANCE SUMMARY

Approach	Prec.	Rec.	F1	AUC	FPR@95
Blocklist	.987	.412	.582	–	–
SVM	.891	.903	.897	.954	.067
Random Forest	.924	.918	.921	.971	.041
URL-CNN	.932	.941	.936	.979	.032
XGBoost	.948	.957	.952	.984	.024
Hybrid	**.961**	**.973**	**.967**	**.991**	**.018**

Bootstrap 95% CI: F1 ±0.008. All improvements $p < 0.01$.

TABLE IV
PREDICTION OUTCOME DISTRIBUTION

	Predicted Threat	Predicted Safe
Actual Threat	5,284	147
Actual Safe	489	26,638

D. Evaluation Criteria

Primary metrics are precision, recall, F1-score, and AUC-ROC, where

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \tag{1}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{FPR} = \frac{FP}{FP + TN}. \tag{2}$$

Operational relevance motivates reporting FPR@95%TPR, the false positive rate when detection sensitivity is fixed at 95%. Statistical significance was assessed using paired bootstrap resampling with 10,000 iterations at significance level $\alpha = 0.05$.

IX. RESULTS

A. Detection Performance

Table III summarizes classification accuracy across the evaluated approaches and Fig. 2 visualizes the comparison. The blocklist achieves only 41.2% recall, confirming its inadequacy against threats that have not yet been listed. The complete hybrid architecture reaches a 96.7% F1-score with a 1.8% false positive rate when configured for 95% detection sensitivity. A direct comparison against commercial gateway products was not performed and is left for future work.

B. Error Characterization

Table IV presents the confusion matrix of the hybrid system on the test set.

False-positive clusters include: URL shorteners that generate suspicious lexical patterns (23%); CDN resources with encoded parameters (19%); recently launched legitimate services that lack reputation history (31%); internationalized domains that trigger similarity heuristics (12%); and aggressive marketing landing pages (15%).

False-negative patterns involve: compromise of established domains that inherit positive reputation (41%); cloud-platform abuse that hosts malicious content (28%); sophisticated URL construction that mimics legitimate patterns (18%); and cloaked content that serves benign payloads to analysis infrastructure (13%).

Illustrative failure. A URL hosted within legitimate cloud storage evaded detection because platform reputation signals and valid certificates outweighed concerning content indicators, which motivates future integration of visual similarity.

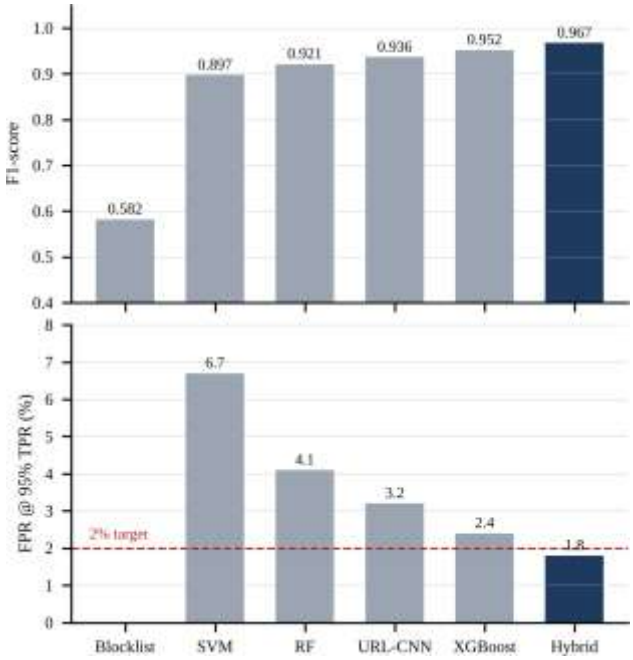


Fig. 2. F1-score (top) and false positive rate at 95% true positive rate (bottom) for each approach, from Table III. The dashed line marks the 2% operational target.

TABLE V
CATEGORY CONTRIBUTION ANALYSIS

Configuration	F1	ΔF1	FPR@95
Complete (82 dim.)	.952	–	.024
Exclude DNS	.931	−.021	.034
Exclude TLS	.944	−.008	.028
Exclude DOM	.923	−.029	.039
Lexical only	.908	−.044	.052

C. Ablation Analysis

Table V reports a leave-one-category-out ablation of the XGBoost stage, and Fig. 3 plots the result. Removing the DOM signals causes the largest drop (−2.9 points of F1), followed by DNS (−2.1) and TLS (−0.8), while lexical signals alone fall 4.4 points short of the complete 82-dimensional set. All category removals produce statistically significant degradation ($p < 0.05$), which supports the multi-modal design.

D. Operational Metrics

Table VI reports runtime performance under sustained load (1,000 concurrent connections, 60-second measurement windows), and Fig. 4 compares median and tail latency against the 200 ms budget.

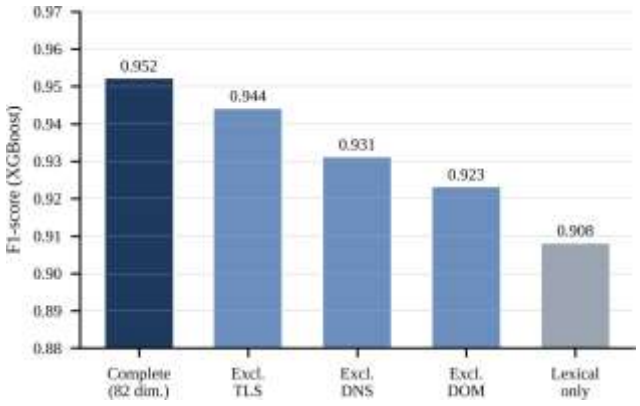


Fig. 3. F1-score of the XGBoost stage when individual feature categories are removed (Table V).

TABLE VI
LATENCY AND THROUGHPUT MEASUREMENTS

Configuration	p50 (ms)	p95 (ms)	Rate/s
XGBoost (CPU)	12	43	3,847
CNN (CPU)	89	156	412
Hybrid Opt. (CPU)	23	78	2,412
Hybrid Opt. (GPU)	14	52	3,156

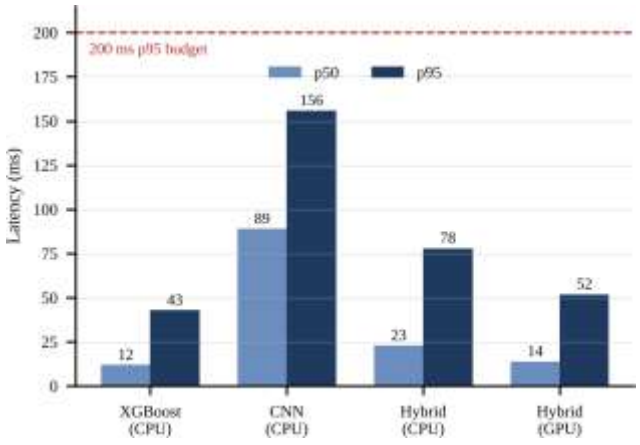


Fig. 4. Median (p50) and 95th-percentile (p95) latency for each configuration (Table VI).

The optimized hybrid configuration satisfies the operational targets: 78 ms tail latency (well under the 200 ms threshold) and 2,412 classifications per second (above the 1,000 minimum) on CPU hardware without acceleration. CPU utilization stabilizes around 65% under sustained load, leaving headroom for traffic bursts.

At enterprise scale, the observed throughput corresponds to inline inspection for approximately 200,000 concurrent users per gateway node, assuming an average of about one classification request per user every 80 seconds.

E. Deployment Observations

- Cache hit rates approached 35% during sustained operation, substantially reducing average response latency.
- DNS timeout events correlated with eventually confirmed threats, suggesting that the occurrence of a timeout itself carries diagnostic signal.
- User warning-override patterns provided valuable ground truth for subsequent model-refinement cycles.

F. Validity Considerations

Several factors may limit generalizability. PhishTank coverage may under-represent certain regional attack categories. Temporal partitioning mitigates, but cannot eliminate, distribution shift between training and deployment periods. Hardware-specific optimizations may require adjustment across platform configurations. Results may also vary across geographic regions with differing domain-registration practices and hosting ecosystems. Organizations deploying the system should validate performance against locally representative threat samples.

X. ETHICAL DIMENSIONS

Security research inherently involves dual-use tensions: the detection techniques described here could, in principle, inform the development of evasion strategies. We judge the defensive benefit to outweigh this concern given the community's existing awareness of the underlying principles. Specific threshold values, trained parameters, and decision boundaries remain unpublished to impede direct exploitation.

This investigation did not involve human subjects and required no institutional review board approval. No personally identifiable information was processed during experimentation, and all evaluation used publicly accessible datasets, namely PhishTank, Tranco, and Common Crawl. Production deployment must implement appropriate privacy safeguards, including data minimization, secure logging practices, and compliance with applicable jurisdictional regulations [27], [28].

XI. CONSTRAINTS AND FUTURE DIRECTIONS

Temporal degradation affects models as adversary tactics evolve. Monitoring indicates 4.2% F1 erosion over six-month periods without retraining, which motivates the integration of online learning mechanisms [23].

Registration privacy limits domain-age availability for 38% of queries because of GDPR-driven WHOIS redaction [28], reducing the capability to identify newly registered domains. Content-access barriers, including JavaScript-heavy rendering, server-side cloaking [29], and rate limiting, impede DOM signal extraction for a significant fraction of traffic.

Adversarial robustness remains only partially characterized. Preliminary perturbation testing (homoglyph substitution and random insertion) degraded accuracy by approximately 3.8% [22].

Future directions include drift-aware online updating via ADWIN-style change detection [24], federated learning across organizational deployments to preserve data sovereignty,

SHAP-based explainability to support analyst review of verdicts, and visual-similarity integration to address brand impersonation hosted on reputable infrastructure [17].

XII. Conclusion

This research demonstrates that effective fake and phishing website defense need not sacrifice operational viability. Through multi-modal signal integration, hybrid inference with confidence-based routing, and deployment-conscious optimization, the proposed architecture achieves a 96.7% F1-score with a 1.8% false positive rate while sustaining more than 2,400 classifications per second at sub-80 ms tail latency.

Temporal evaluation reveals realistic generalization characteristics that investigations using random data partitioning do not capture, and systematic ablation quantifies category contributions to guide deployment decisions under resource constraints. By bridging the persistent gap between research-grade accuracy and operational feasibility, this work provides a practical blueprint for next-generation real-time phishing defense infrastructure capable of protecting organizations against continuously evolving social-engineering threats.

Reproducibility Statement

Experiments employed a fixed random seed (42), specified software versions, and a documented hardware configuration. The signal-extraction schema and evaluation protocols accompany this publication. Trained model weights remain restricted to prevent adversarial exploitation.

Acknowledgment

The authors acknowledge the PhishTank, Tranco, and Common Crawl maintainers for the data resources that made this investigation possible.

References

[1] Anti-Phishing Working Group, "Phishing Activity Trends Report," 2023–2024.
[2] FBI Internet Crime Complaint Center, "Internet Crime Report," 2024.
[3] Verizon, "Data Breach Investigations Report," 2024.
[4] ENISA, "Threat Landscape Report," 2023.
[5] C. Seifert, I. Welch, and P. Komisarczuk, "Identification of malicious web pages with static heuristics," in *Proc. ATNAC*, 2008.
[6] Y. Zhang, J. Hong, and L. Cranor, "CANTINA: A content-based approach to detecting phishing web sites," in *Proc. WWW*, 2007.
[7] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. eCrime*, 2007.
[8] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupe´, and G.-J. Ahn, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale," in *Proc. USENIX Security*, 2020.
[9] J. Ma, L. Saul, S. Savage, and G. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. KDD*, 2009.
[10] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *Proc. NDSS*, 2010.
[11] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proc. WWW*, 2007.
[12] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," arXiv:1701.07179, 2017.
[13] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," arXiv:1802.03162, 2018.
[14] O. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, 2019.
[15] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proc. NDSS*, 2019.
[16] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. 10, 2022.
[17] R. Lin et al., "Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages," in *Proc. USENIX Security*, 2021.
[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. KDD*, 2016.
[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[20] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, 2017.
[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019.
[22] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR*, 2015.
[23] J. Gama, I. Zˇ liobaite˙, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, 2014.
[24] A. Bifet and R. Gavalda`, "Learning from time-changing data with adaptive windowing," in *Proc. SDM*, 2007.
[25] Common Crawl Foundation, "Common Crawl," https://commoncrawl.org, 2024.
[26] PhishTank, "Join the fight against phishing," https://phishtank.org, 2024.
[27] NIST, "The NIST Cybersecurity Framework (CSF) 2.0," 2024.
[28] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," *Official Journal of the European Union*, 2016.
[29] P. Zhang et al., "CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing," in *Proc. IEEE Symp. Security and Privacy (S&P)*, 2021.