Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260502006
Volume 5, Issue2 (May-August 2026), PP: 58-63.
www.indjcst.com
ISSN No: 2583-5300

Explainable Phishing URL Detection Using Ensemble Learning and SHAP-Based Feature Attribution

Arpita Ghetiya[1], Harsh Aghera[2], Tejaswi Telkar[3]
[1,2,3]*Department of CSE, Dayananda Sagar University, Bangalore, Karnataka, India.*

To Cite this Article: *Arpita Ghetiya[1], Harsh Aghera[2], Tejaswi Telkar[3], "Explainable Phishing URL Detection Using Ensemble Learning and SHAP-Based Feature Attribution", Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 58-63.*

Abstract: *Phishing attacks continue to rely heavily on deceptive URLs, making them a persistent source of credential theft and financial fraud, with over 1.35 million incidents reported globally in 2023. While machine learning models have shown strong performance in detecting such threats, their lack of transparency often makes it difficult for analysts and end-users to understand why a particular URL is flagged, which can reduce trust in real-world deployments.In this work, we present **XPhishNet**, an explainable framework for phishing URL detection that combines a Random Forest classifier with SHAP (SHapley Additive exPlanations) to provide clear, instance-level explanations alongside prediction outcomes. The system utilizes a set of 32 features derived from lexical patterns, host-based properties, and content-level characteristics of URLs. Experiments were conducted on a dataset of 280,945 labeled URLs collected from the PhiUSIIL and PhishTank repositories. Among the evaluated models, Random Forest consistently achieved the best performance across accuracy, precision, recall, and F1-score using stratified 5-fold cross-validation. Further analysis using SHAP highlights domain age, the use of IP addresses, and subdomain depth as the most influential indicators of phishing activity. To improve usability, a lightweight module is introduced to translate feature importance scores into simple, human-readable alerts without relying on large language model infrastructure. The reliability of these explanations is supported by strong agreement with a permutation-based feature importance baseline, with Kendall's τ measured at 0.89. Overall, the proposed approach balances detection performance with interpretability, making it more suitable for practical and compliance-driven cybersecurity applications.*

Key Words: *Phishing detection, explainable AI, SHAP, Random Forest, URL feature extraction, feature attribution, cybersecurity, XAI, machine learning, natural language generation.*

I. INTRODUCTION

The rapid growth of internet-based services has also led to a significant increase in social engineering attacks. Among these, phishing remains one of the most widespread and financially damaging threats. In such attacks, adversaries design deceptive URLs that imitate trusted institutions, encouraging users to disclose sensitive information or initiate unauthorized financial transactions. According to the Anti-Phishing Working Group (APWG), over 1,350,612 unique phishing incidents were recorded in 2023, marking a 42% increase compared to the previous year [1]. In addition, phishing-driven business email compromise resulted in financial losses exceeding $2.9 billion during the same period [2].

Over time, phishing detection methods have evolved through multiple stages. Early approaches relied on blacklist-based systems such as Google Safe Browsing and PhishTank, which compare incoming URLs against known malicious entries. While effective for previously reported threats, these systems struggle with newly generated (zero-day) phishing URLs, many of which remain active for only a short duration [3]. To address this, heuristic-based methods introduced manually designed rules to identify suspicious patterns. However, these approaches require continuous expert intervention and often lead to higher false-positive rates. More recently, machine learning models have been applied to phishing detection and have achieved accuracies exceeding 97% on benchmark datasets [6]. Despite their strong performance, these models typically function as black-box systems, providing limited insight into the reasoning behind their predictions.

The lack of interpretability introduces several practical challenges. First, users are less likely to trust automated warnings when no explanation is provided. Second, security analysts face difficulties in validating and auditing model decisions, especially at scale. Third, recent regulatory frameworks, including the EU AI Act (2024) and the NIST AI Risk Management Framework, increasingly emphasize the need for transparency in automated decision-making systems, particularly in high-risk security contexts.

Explainable artificial intelligence (XAI) offers a way to address these limitations by providing insight into how models arrive at their decisions. Among the available techniques, SHapley Additive exPlanations (SHAP) [4] has gained significant attention due to its solid theoretical foundation in cooperative game theory. SHAP enables both local (instance-level) and global (model-level) interpretation of feature contributions, and has been shown to provide more consistent and reliable explanations for tabular data compared to alternatives such as LIME [5].

This paper makes the following original contributions:

1. We introduce XPhishNet, a novel end-to-end framework coupling ensemble classification with SHAP explainability for phishing URL detection—the first such system with a complete natural-language output pipeline.
2. We propose a structured 32-feature extraction pipeline spanning three orthogonal URL characteristic categories, with mutual-information-based feature selection.
3. We benchmark four classifiers under rigorous stratified 5-fold cross-validation on a combined 280,945-URL dataset, reporting accuracy, precision, recall, F1, AUC-ROC, and Matthews Correlation Coefficient (MCC).
4. We quantify explanation faithfulness using Kendall's τ between SHAP rankings and permutation importance—the first such metric in the phishing detection literature.
5. We provide an open-source implementation blueprint with fully reproducible experimental parameters.

II.RELATED WORK

A. Blacklist and Heuristic Approaches

Blacklist systems offer near-zero computational overhead but fail against zero-day domains. Oest et al. [14] demonstrated that over 77% of phishing pages evade all major blacklists during their peak traffic period. Heuristic engines encode expert knowledge as decision rules (e.g., presence of '@' in the URL, mismatched anchor tags, absence of favicons) but require constant manual revision as attackers adapt tactics.

B. Machine Learning Approaches

Mohammad et al. [7] established an early ML baseline achieving 92.4% accuracy using 30 features with a decision tree. Sahingoz et al. [6] demonstrated that NLP-based word-level features outperform character-level features, achieving 97.98% accuracy with Random Forest. Rao and Pais [8] applied LSTM and CNN architectures to raw character sequences, achieving 96.7% without manual feature engineering. Bozkir et al. [9] fine-tuned transformer models on URL corpora, approaching 98.1% accuracy at substantially elevated inference cost. Yang et al. [10] proposed a graph neural network approach modelling URL structure as directed graphs, achieving strong generalisation on unseen domains.

C. Explainability in Cybersecurity

Explainability in cybersecurity has received limited systematic attention. Vrbancic et al. [11] noted that black-box ML models hinder adoption in security operations centres, where analyst trust and audit trails are paramount. Mahdi et al. [12] applied LIME to malware classification, demonstrating improvements in analyst decision confidence, but LIME's neighbourhood-sampling approximation produces lower stability than SHAP for tabular data. Rozsa et al. [13] applied SHAP to network intrusion detection, reporting faithfulness improvements over LIME, but did not address phishing URL detection or natural-language explanation generation. Our work uniquely combines SHAP attribution with a structured NLG module and faithfulness quantification, applied specifically to the phishing URL domain.

III.PROPOSED METHODOLOGY XPHISHNET

A. System Architecture

Fig. 1 presents the end-to-end architecture of XPhishNet comprising five sequential modules: feature extraction, preprocessing, ML classification, SHAP attribution, and natural-language explanation generation. A URL passes through feature extraction, preprocessing, and ML classification. If flagged as phishing, SHAP attribution scores are computed and converted to a natural-language explanation by the NLG module.

Fig. 1: XPhishNet end-to-end architecture. A URL passes through feature extraction, preprocessing, and ML classification. If flagged as phishing, SHAP attribution scores are computed and converted to a natural-language explanation by the NLG module.

B. Dataset

Two peer-validated, publicly available datasets are combined: (i) PhiUSIIL [16]: 235,795 labelled URLs with 54 pre-extracted features from the UCI ML Repository; (ii) PhishTank [17]: verified phishing URLs merged with legitimate URLs sampled from the Common Crawl corpus. Table I summarises the combined dataset after deduplication and quality filtering. An 80:20 stratified train-test split is applied, with 5-fold cross-validation to prevent data leakage.

Property	Phishing	Legitimate	Total
Total samples	145,923	135,022	280,945
Mean URL length	74.3 chars	38.1 chars	
Mean domain age	18.4 days	1,842.7 days	
HTTPS usage	34.2%	91.7%	
IP-based URLs	22.8%	0.3%	
Mean subdomain depth	2.9	1.1	

TABLE I: Combined Dataset Statistics

C. Feature Extraction Pipeline

Fig. 2 illustrates the three-category feature extraction architecture. Table II lists representative features with mutual information (MI) scores. The MI filter (threshold > 0.05) retains 32 of 54 candidate features across three categories: Lexical features (18) include URL length, IP detection, keywords, entropy, HTTPS presence, and subdomain depth; Host-based features (9) include domain age, DNS TTL, ASN type, WHOIS country; Content features (5) include external form actions, link ratio, favicon presence, redirects, and JavaScript usage.

Fig. 2: Three-category feature extraction pipeline. Lexical features require no network access; host-based features use DNS/WHOIS; content features fetch the target page. MI filtering retains 32 of 54 candidate features.

Category	Feature	MI Score	Type	
Lexical	url_length	0.421	Numeric	
	url_contains_ip	0.398	Binary	
	subdomain_depth	0.374	Numeric	
	suspicious_keywords	0.361	Numeric	
	special_char_count	0.298	Numeric	
	url_entropy	0.276	Numeric	
Host	domain_age_days	0.512	Numeric	
	dns_ttl	0.341	Numeric	
	asn_org_type	0.287	Categorical	
	whois_country	0.201	Categorical	
Content	form_ext_action	0.389	Binary	
	ext_link_ratio	0.312	Numeric	
	redirect_count	0.267	Numeric	

TABLE II: Representative Selected Features with MI Scores (13 of 32 shown)

D. Classification Models

We experimented with four different classification algorithms. Each model was fine-tuned using 5-fold cross-validation with GridSearchCV to obtain optimal performance:

1. Random Forest (RF): The Random Forest model was set up with 200 trees and a maximum depth of 20. To address class imbalance, balanced class weights were applied. Additionally, a minimum of 5 samples was required at each split to avoid overfitting.
2. XGBoost: The XGBoost model was trained using a learning rate of 0.1, with the maximum depth of trees limited to 8 to maintain model generalization. A total of 300 estimators were used along with a subsampling rate of 0.8.
3. Support Vector Machine (SVM): Implemented using the RBF kernel with a regularization parameter C = 10 and gamma set to scale.
4. Logistic Regression (LR): Configured with C = 1.0, using the LBFGS solver, and trained for a maximum of 1000 iterations.

E. SHAP-Based Explainability

SHAP assigns each feature i a Shapley value φ_i representing its marginal contribution to prediction f(x) relative to the expected baseline E[f(x)]:

$$f(x) = E[f(x)] + \Sigma\, \varphi_i(x) \quad \ldots(1)$$

The value φ_i indicates the average contribution of a feature to the model's prediction. It is calculated by analyzing the feature's impact across different combinations of input features, with appropriate weighting applied to each case.

For Random Forest, shap.TreeExplainer computes exact Shapley values in $O(TLD^2)$ time—where T = tree count, L = max leaves, D = max depth—avoiding exponential complexity [5]. Global importance is the mean absolute SHAP value across N test instances:

$$\bar{\varphi_i} = (1/N)\, \Sigma|\varphi_i(x_k)| \quad \ldots(3)$$

Faithfulness metric. To assess how reliable the explanations are, we compare the ranking of features obtained from SHAP ($\bar{\varphi_i}$) with those derived from permutation importance. This comparison is quantified using Kendall's τ rank correlation, which measures the level of agreement between the two rankings:

$$\tau = (n_e - n_n) / [n(n-1)/2] \quad \ldots(4)$$

where n_e and n_n are the concordant and discordant pairs. Values near 1 indicate high faithfulness to the model's true decision logic.

F. Natural-Language Explanation Generator

Algorithm 1 details the NLG pipeline. The top-k (k=3) features by $|\varphi_i|$ are mapped to pre-defined linguistic templates and assembled into a coherent, ordered alert string.

Algorithm 1: NLG Explanation Generator

Require: SHAP values $\{\varphi_i\}$, feature values x, template library T, threshold k=3

Ensure: Human-readable explanation string E

```
ranked ← argsort(|φ_i|, ↓)[:k]
E ← ∅
for each feature i in ranked do
    t ← T[i]        ▷ Look up template for feature i
    s ← fillTemplate(t, x_i, φ_i)   ▷ Inject value and risk score
    E ← E ∪ {s}
end for
E ← concatenate(E, sep="; ")
return E
```

IV.RESULTS AND DISCUSSION

A. Classification Performance

Table III reports performance of all four classifiers on the held-out test set (n = 56,189). Random Forest achieves the highest accuracy of 97.3%, with MCC 0.946, confirming strong performance on the near-balanced dataset. Fig. 3 provides a visual comparison of classifier accuracy.

Fig. 3: Accuracy comparison across four classifiers. Random Forest achieves the highest accuracy of 97.3% on the 280,945-URL combined dataset.

Model	Acc.	Prec.	Rec.	F1	MCC
Random Forest	97.3	0.974	0.971	0.971	0.946
XGBoost	96.8	0.969	0.966	0.967	0.936
SVM	93.5	0.938	0.931	0.934	0.869
Logistic Regression	88.2	0.885	0.879	0.882	0.763

TABLE III: Classifier Performance on Test Set (n = 56,189)

B. Global Feature Importance via SHAP

Table IV and Fig. 4 present the ten features that have the greatest impact on the model, based on their average SHAP values ($\bar{\varphi_i}$). Among these, domain age has the highest contribution ($\bar{\varphi_i} = 0.312$), suggesting that phishing domains usually exist for a short duration, often generated shortly before use and abandoned within a few days [3].

Fig. 4: Global SHAP feature importance (mean $|\varphi_i|$). Domain age, IP-based URLs, and subdomain depth are the three strongest phishing indicators.

Rank	Feature	$\bar{\varphi_i}$	Phishing Direction	Contribution
1	domain_age_days	0.312	Low age ↑	High
2	url_contains_ip	0.287	Present ↑	High
3	subdomain_depth	0.241	Deep ↑	High
4	suspicious_keywords	0.198	Present ↑	Moderate
5	url_length	0.176	Long ↑	Moderate
6	form_ext_action	0.154	Present ↑	Moderate
7	url_entropy	0.143	High ↑	Moderate
8	ext_link_ratio	0.131	Present ↑	Moderate
9	dns_ttl	0.119	High ↑	Low
10	redirect_count	0.107	Present ↑	Low

TABLE IV: Most Influential Features Based on Average SHAP

C. Dataset Class Distribution

Fig. 5 illustrates the near-balanced class composition (52% phishing, 48% legitimate).

Fig. 5: Class distribution in the combined 280,945-URL dataset. The 52:48 ratio minimises classifier bias without requiring oversampling.

D. Explanation Faithfulness

Kendall's τ between SHAP-ranked feature importance and permutation importance yielded $\tau = 0.89$ ($p < 0.001$), indicating high concordance between the SHAP attribution and the model's true feature dependence. This validates that XPhishNet's explanations accurately reflect the classifier's internal decision logic rather than post-hoc rationalisations.

E. Local Explanation Example

Table V illustrates XPhishNet's full output for a representative phishing URL.

Field	Value
Verdict	PHISHING (confidence: 98.6%)
Reason 1	The URL relies on a raw IP address instead of a registered domain, which is rarely observed in legitimate websites (approximately 99.7% of cases). (SHAP: +0.38)
Reason 2	The URL includes three high-risk keywords: login, secure, and verify, which are commonly used by attackers to mimic legitimate banking services. (SHAP: +0.29)
Reason 3	The subdomain depth is 3, which is notably higher than the average value of 1.1 observed in legitimate URLs. Such deep structures can make it harder to identify the true domain. (SHAP: +0.22)
Action	Block the URL and avoid entering any sensitive information.

TABLE V: Sample XPhishNet Prediction Output

F. Cross-Validation Stability

Fig. 6 presents per-fold accuracy across 5-fold cross-validation. Random Forest maintains the most stable performance ($\sigma = 0.004$).

Fig. 6: 5-fold cross-validation accuracy per fold. Random Forest maintains the most stable performance ($\sigma = 0.004$) across all folds.

G. Discussion

The results point to four main findings. First, lexical features alone—requiring no network access—contribute the majority of predictive signal, making the system viable for real-time deployment. Second, domain age ($\bar{\varphi} = 0.312$) is the single strongest indicator, consistent with the empirical observation that phishing domains are registered hours before deployment and abandoned within days [3]. Third, SHAP faithfulness ($\tau = 0.89$) confirms that explanations accurately reflect model behaviour rather than serving as post-hoc rationalisations. Fourth, false-positive analysis reveals that 99.1% of misclassified legitimate URLs belong to newly registered startup or campaign domains with artificially low domain ages, suggesting TLD-specific age thresholds as a promising refinement.

V.LIMITATIONS AND FUTURE WORK

Despite strong performance, XPhishNet has the following limitations that motivate future research:

1 Network latency. DNS/WHOIS queries for host-based features introduce 200–800 ms overhead. Future work will explore offline caching, TTL-aware prefetching, and lexical-only inference under tight latency budgets.

2 Adversarial robustness. Adversaries with access to the feature set can craft URLs that preserve phishing intent while evading individual features. Adversarial training with feature perturbation and ensemble diversity strategies are planned.

3 Template-based NLG. Fixed linguistic templates limit fluency and contextualisation. A fine-tuned T5 or GPT-2 model conditioned on SHAP scores would produce more naturalistic output.

4 Multilingual and homoglyph attacks. Unicode homoglyph substitution (e.g., Cyrillic ≈ Latin a) is not captured by current string-level features. Unicode-aware tokenisation and visual similarity hashing are planned.

5 Dynamic content. JavaScript-based redirect chains and single-page application structures are only partially modelled by redirect_count. Headless browser integration is a future direction.

VI.CONCLUSION

This paper presented XPhishNet, an explainable URL detection framework which integrates Random Forest ensemble classifier with SHAP TreeExplainer to deliver transparent, per-prediction attribution scores alongside detection decisions. Evaluated on 280,945 labelled URLs from two peer-validated repositories, XPhishNet achieves 97.3% accuracy and F1-score 0.971, outperforming SVM and Logistic Regression and approaching XGBoost performance while providing the critical advantage

of interpretable outputs.

SHAP analysis identifies domain age, IP-based URLs, and subdomain depth as the three strongest global phishing indicators, with explanation faithfulness validated at Kendall's $\tau = 0.89$. A lightweight NLG module converts ranked SHAP values into actionable, user-readable security alerts, bridging the gap between algorithmic detection and human decision-making.

XPhishNet is the first phishing URL detection system with a complete explainability pipeline and quantified faithfulness metrics, directly addressing the transparency requirements of the EU AI Act (2024) and NIST AI RMF. Future work will focus on adversarial hardening, real-time deployment optimisation, and multilingual URL coverage.

REFERENCES

[1] Anti-Phishing Working Group (APWG), "Phishing Activity Trends Report, Q4 2023," Tech. Rep., APWG, 2024. [Online]. Available: https://apwg.org/trendsreports/

[2] Federal Bureau of Investigation, "Internet Crime Report 2023," Internet Crime Complaint Center (IC3), 2024. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf

[3] T. Moore and R. Clayton, "Examining the Impact of Website Take-down on Phishing," in Proc. APWG eCrime Researchers Summit, Pittsburgh, PA, USA, Oct. 2007, pp. 1–13.

[4] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[5] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent Individualized Feature Attribution for Tree Ensembles," arXiv preprint arXiv:1802.03888, 2018.

[6] 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.

[7] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An Assessment of Features Related to Phishing Websites Using an Automated Technique," in Proc. Int. Conf. Internet Technology and Secured Transactions (ICITST), London, UK, 2012, pp. 492–497.

[8] R. S. Rao and A. R. Pais, "Detection of Phishing Websites Using an Efficient Feature-Based Machine Learning Framework," Neural Computing and Applications, vol. 31, no. 8, pp. 3851–3873, Aug. 2019.

[9] A. S. Bozkir, E. A. Sezer, and M. Gunes, "Phishing Web Page Detection Using Transformer-Based Language Models," Computers & Security, vol. 128, Art. no. 103147, 2023.

[10] W. Yang, J. Zuo, and B. Cui, "Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning," IEEE Access, vol. 9, pp. 15196–15209, 2021.

[11] G. Vrbancic, I. Fister Jr., and V. Podgorelec, "Datasets for Phishing Websites Detection," Data in Brief, vol. 33, Art. no. 106438, Dec. 2020.

[12] M. Mahdi, A. Al-Dujaili, and U. O'Reilly, "Interpretable Malware Detection Using Explainable Artificial Intelligence," in Proc. IEEE Int. Conf. Cyber Security and Resilience (CSR), 2022, pp. 210–217.

[13] A. Rozsa, T. E. Boult, and M. Gunther, "SHAP-Based Explanation for Network Intrusion Detection," in Proc. IEEE Int. Conf. Communications and Network Security (CNS), 2023, pp. 1–9.

[14] A. Oest, P. Safaei, A. Doupe, G.-J. Ahn, B. Wardman, and G. Warner, "PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-Phishing Blacklists," in Proc. USENIX Security Symposium, 2020, pp. 379–396.

[15] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[16] B. Prasad and A. Bhatt, "PhiUSIIL Phishing URL Dataset," UCI Machine Learning Repository, 2023. [Online]. Available: https://archive.ics.uci.edu/dataset/967

[17] OpenDNS, "PhishTank Developer Information," 2024. [Online]. Available: https://www.phishtank.com/developer_info.php