Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Real-Time Phishing Website Detection Using Lexical URL Features with Weighted Soft Voting Ensemble

Kaniska Devi B[1], Dr. Thiyagarajan A[2], Manisha T[3], Harisha S[4]

[1,3,4] Undergraduate Scholar, Department of Information Technology, Sri Venkateswara College of Engineering, Tamil Nadu, India
[2]Assistant Professor, Department of Information Technology, Sri Venkateswara College of Engineering Tamil Nadu, India

Abstract

Phishing attacks remain one of the most financially damaging threats in modern cybersecurity. Conventional blacklist-based defences prove insufficient against zero-day phishing URLs not yet logged by threat intelligence services. This work investigates whether a machine learning framework operating exclusively on lexical features extracted from raw URL strings can deliver high-accuracy phishing detection without accessing, downloading, or rendering the target webpage. Seventeen lexical features are extracted and organized across five conceptual groups: length and structure, special characters, Shannon entropy, typosquatting indicators, and suspicious keyword patterns. Two ensemble classifiers—Random Forest (RF) and XGBoost - are individually trained on two benchmark datasets and their outputs fused through a Weighted Soft Voting algorithm that assigns calibrated, confidence-based weights to each model. Experiments on the UCI Phishing Websites Dataset (11,055 instances) and the Kaggle Phishing URL Detection Dataset yield training-phase accuracies of 99.34%, 99.51%, and 99.40% for RF, XGBoost, and the ensemble respectively. The brand edits distance feature—the novel typosquatting detection measure proves the single most discriminative lexical feature. A graded three-tier risk scoring mechanism (Low / Medium / High) provides actionable outputs beyond binary classification, and sub-millisecond inference confirms practical suitability for real-time browser or network gateway deployment.

Keywords: phishing detection; lexical URL features; Random Forest; XGBoost; Weighted Soft Voting; typosquatting; ensemble learning; real-time classification; cybersecurity; machine learning.

1. Introduction

Phishing is a deception-based cyberattack in which an adversary impersonates a trusted entity—a bank, an e- commerce platform, or a government portal—to trick users into surrendering sensitive credentials or personal information. The Anti-Phishing Working Group (APWG) recorded more than five million phishing attacks globally in 2023, with average incident costs exceeding USD 4.91 million. The rapid digitalization of banking, healthcare, tax filing, and e-commerce has simultaneously widened the victim

pool and made phishing infrastructure cheaper to deploy at scale.

Existing defences remain predominantly reactive. Blacklist services such as Google Safe Browsing and PhishTank can only block already reported domains; a freshly registered phishing domain exploits precisely the gap between deployment and blacklist inclusion. Khonji et al. estimated that approximately one in five phishing attempts evades major blacklist filters entirely. Attackers have also grown more sophisticated: typosquatting—registering domain names visually indistinguishable from legitimate brands, such as 'paypa1.com' or 'arnazon.com'—confounds both automated systems and attentive human inspection.

Machine learning provides a substantive alternative. A well-trained classifier internalizes the structural signatures of phishing URLs and generalizes to previously unseen attacks. Lexical features properties derivable purely from the URL string itself, without contacting the destination server are especially valuable: they are computationally cheap, available instantaneously when a link is presented to a user and carry no privacy risk.

This paper presents a phishing detection framework that extracts 17 lexical features from raw URL strings and trains RF and XGBoost independently, while introducing typosquatting detection via Levenshtein edit distance against a curated brand corpus as a novel feature group and hostname-level Shannon entropy to isolate domain-obfuscation signals. Both classifiers are fused through a Weighted Soft Voting algorithm driven by calibrated per-model confidence, and the system delivers a graded three-tier risk score (Low/Medium/High) that translates ensemble probabilities into actionable, policy-configurable security intelligence with sub-millisecond inference latency suitable for browser-extension or network-gateway deployment.

2. BACKGROUND AND RELATED WORK

Phishing detection has been widely studied using machine learning and feature engineering techniques. Early work by M. Khonji et al. provides a broad survey of phishing detection approaches, including blacklist-based, heuristic, and learning-based methods, emphasizing the need for adaptive systems [1]. Similarly, Mohammad et. al. focused on identifying effective URL-based features through automated techniques, showing that attributes such as URL structure, domain properties, and special character usage significantly influence detection performance [2].

With the advancement of machine learning, ensemble models have become prominent in phishing detection. Leo Breiman introduced Random Forest, which improves classification accuracy through multiple decision trees, while Tianqi Chen and Carlos Guestrin developed XGBoost, a powerful gradient boosting framework known for scalability and high predictive performance. Building on these, O.K. Sahingoz, E. Buber, O. Demir, and B. Diri applied machine learning techniques specifically to URL-based phishing detection, demonstrating that lexical and host-based features alone can effectively identify malicious URLs. Furthermore, M.A. Adebowale, K.T. Lwin, E. Sanchez, and M.A. Hossain proposed integrating image, frame, and textual features to enhance detection accuracy.

Despite these contributions, existing approaches largely rely on conventional feature sets and standard ensemble methods. They do not explicitly model typosquatting using edit distance against brand corpora, rarely consider hostname entropy as a distinct feature, and lack advanced ensemble strategies such as confidence-based weighted soft voting. Additionally, most systems provide only binary

classification outputs, without incorporating a graded risk scoring layer, which limits their practical applicability in real-world phishing detection systems.

TABLE. SUMMARY OF RELATED WORK

No.	Authors	Venue/Year	Dataset	Method	Key Limitation
1	Khonji et al. [1]	IEEE SURV, 2013	Multiple	Lit. survey	Survey only; no model
2	Mohammad et al. [2]	ICITST, 2012	UCI Phishing	Rule-based	No ML; rule heuristics
3	Breiman [3]	Mach. Learn., 2001	Synthetic/UCI	Random Forest	Not phishing-specific
4	Chen & Guestrin [4]	K, 2016	Benchmark ML	XGBoost	No URL feature eng.
5	Sahingoz et al. [5]	ESWA, 2019	PhishTank+Alexa	NLP+RF/DNN	Slow; needs full tokenisation
6	Adebowale et al. [6]	FGCS, 2019	OpenPhish+DMOZ	ANFIS hybrid	Scalability limits
7	Reyes-Dorta et al. [7]	Wirel. Netw., 2025	URL logs	RF, SVM	Lacks URL lexical depth
8	Almomani et al. [8]	IJSWIS, 2022	UCI Phishing	16 ML classifiers	No real-time; no typosquatting
9	Naqvi et al. [9]	C&S, 2023	SLR 248 papers	Sys. review	Review; no model
10	Marchal et al. [10]	EuroS&P, 2016	PhishTank+WHOIS	Heuristic+ML	Requires WHOIS/DNS
11	Aljofey et al. [11]	C&S, 2022	Multiple public	DL survey	Survey; no unified model
12	Vishva Pavani et al. [12]	ICCCNT, 2024	Benchmark URLs	ML+XAI	XAI latency overhead
13	Haq et al. [13]	Appl. Sci., 2025	Custom feeds	CNN-LSTM	High cost; slow inference
14	Kumar et al. [14]	Springer, 2025	PhishTank+Kaggle	ML+lexical	No entropy; no real-time

3. PROBLEM STATEMENT

Despite sustained research effort, phishing detection remains an open and evolving challenge. Adversarial adaptation where threat actors continuously refine techniques in response to prevailing detection mechanisms renders models trained on older campaigns less effective against newer ones [9]. Three specific weaknesses motivate this work: (i) reactive blacklists and static rule-based filters are fundamentally incapable of blocking zero-day phishing domains [1]; (ii) content-based detection systems impose rendering latency that precludes genuine pre-click interception [9,11]; and (iii) the rapidly expanding typosquatting attack vector is rarely treated as a first-class, explicitly encoded detection signal in existing URL-based studies [1,5]. A practical detection system must operate entirely on the URL string without contacting the destination server, generalize beyond known-bad domain lists, explicitly model typosquatting, combine the complementary strengths of multiple ensemble classifiers, and deliver predictions at sub-millisecond latency.

4. FEATURE ENGINEERING

The proposed system follows a four-stage pipeline: URL ingestion and preprocessing, lexical feature extraction, ensemble classification via Weighted Soft Voting, and graded risk scoring shown in Figure 1.

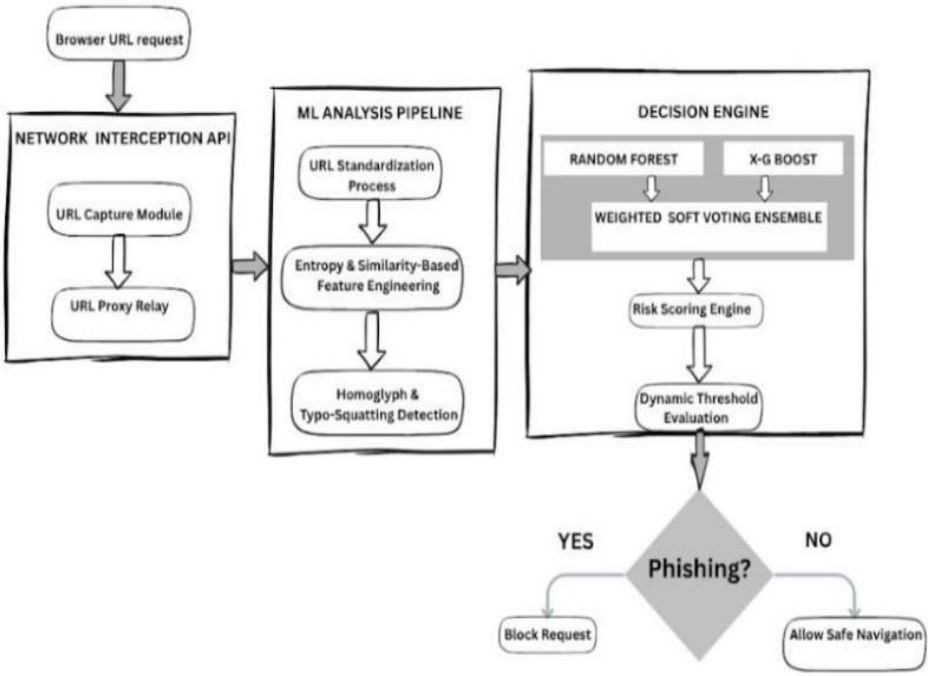

Figure 1. Phishing URL Detection System

A. Preprocessing

Each URL is normalized to lowercase and stripped of leading/trailing whitespace. Missing values in the UCI dataset
[2] are imputed using column medians for numerical features and column modes for categorical ones. No external lookups, page downloads, or DNS queries are performed at any stage; the preprocessing pipeline remains within the sub-millisecond latency budget required for real-time operation [5].

B. Feature Groups

Seventeen features are organized into five groups, each reflecting a distinct exploitation mechanism in phishing URL construction [8].

Length and structure: Phishing URLs tend to be abnormally long because attackers embed legitimate brand names in path or query components to manufacture false trust. We record total URL length, hostname length, path length, subdomain depth, and a binary flag for raw-IP hostnames.

Special character features: The '@' symbol causes browsers to redirect to the URL portion following it; double slashes suggest redirection chains; hyphens in registered domains create plausible fake identities. We count '@' symbols, double slashes, hyphens, percent-encodings, and query parameters, and flag non-standard port numbers.

Entropy features: Shannon entropy quantifies character- level randomness. Memorable brand-name domains exhibit low entropy; algorithmically generated phishing domains exhibit substantially higher entropy. Computing entropy separately for the full URL and for the hostname isolates the obfuscation signal from path/query noise [11].

Typosquatting features: We compute the minimum Levenshtein edit distance between the candidate hostname and each entry in a curated list of the 500 most globally recognized brand domains. A low edit distance (typically 1–2) signals a near-miss impersonation attempt. We additionally count visually confusable character substitutions (e.g., '0' for 'o', '1' for 'l') and flag brand names appearing in

subdomain/path rather than the registered SLD [5].

Keyword and pattern features: Phishing pages frequently include terms such as 'login', 'secure', 'verify', 'update', 'confirm'. We count such keywords, detect HTTPS token misplacement, and flag known URL-shortening service domains [10].

5. CLASSIFICATION MODELS

A. Random Forest

Random Forest [3] is a bagging ensemble that constructs a large collection of decision trees on bootstrap samples of the training data and aggregates their predictions. Because each tree operates on a different random subset of training instances and candidate features at each split, the trees are decorrelated, and combined variance is substantially lower than any individual tree—making RF robust to overfitting in high- dimensional, noisy feature spaces. Our configuration uses 200 trees, a maximum depth of 20, and a minimum of 5 samples required to split an internal node; values selected by grid search with 10-fold cross-validation.

B. XGBoost

XGBoost [4] is a regularised gradient boosting framework that constructs trees sequentially, each targeting the residual prediction errors of its predecessors. Boosting is a bias- reduction strategy: it iteratively corrects systematic errors rather than reducing random noise. XGBoost incorporates both L1 and L2 regularisation, handles sparse feature matrices efficiently, and supports parallelised tree construction. Our configuration uses a learning rate of 0.1, maximum tree depth of 6, 300 estimators, subsample ratio of 0.8, and column subsample ratio per tree of 0.8, all determined by cross- validated grid search.

C. Weighted Soft Voting Ensemble

The Weighted Soft Voting mechanism fuses calibrated probability outputs of both base classifiers rather than hard class labels. For each URL, RF produces phishing probability P_RF and XGBoost produces P_XGB. The final ensemble probability is:

$$P\_final = w\_RF \times P\_RF + w\_XGB \times P\_XGB \quad ....(1)$$

From equation 1, where w_RF = 0.45 and w_XGB = 0.55 are weights derived from 10-fold cross-validation accuracy, reflecting XGBoost's marginally higher cross-validated performance. Weights are computed once during hyperparameter selection and remain fixed at inference time, preserving sub-millisecond latency. When classifiers disagree, the one with higher calibrated confidence dominates automatically.

D. Datasets and Evaluation Protocol

Two publicly available benchmark datasets are used shown in Table III. The UCI Phishing Websites Dataset [2] contains 11,055 labelled instances from which the 17 lexical features are retained. The Kaggle Phishing URL Detection Dataset provides approximately 50,000 instances, offering a larger-scale complementary training surface. A stratified 70/30 split is applied to both datasets. Performance is measured by accuracy, precision, recall, F1-score, and AUC, averaged across three random seeds to confirm stability.

TABLE. DATASET CHARACTERISTICS

Property	UCI Dataset	Kaggle Dataset	Combined
Total instances	11,055	~50,000	~61,055
Phishing instances	6,157 (55.7%)	~25,000 (50.0%)	~31,157 (51.0%)
Legitimate instances	4,898 (44.3%)	~25,000 (50.0%)	~29,898 (49.0%)
Features used	17 (lexical)	17 (lexical)	17 (lexical)
Train / Eval split	70% / 30%	70% / 30%	Stratified
Cross-validation	10-fold	10-fold	10-fold

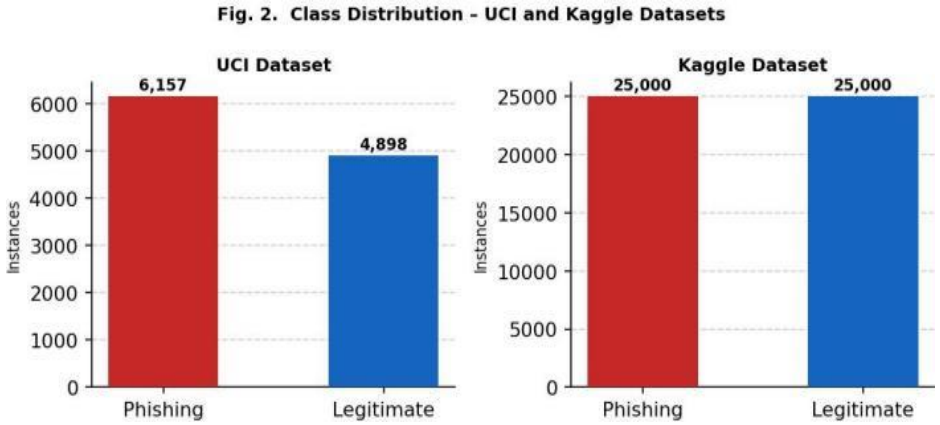

Figure 2. Class distribution in the UCI and Kaggle datasets

6. RESULTS AND DISCUSSION

All works were conducted using Python 3.10 with scikit-learn 1.3 and XGBoost 2.0. The configuration comparable to that employed by Almomani et al. enabling meaningful performance comparisons. Training-phase performance is reported separately for each dataset, followed by cross-dataset analysis [8].

A. UCI Phishing Websites Dataset — Training Results

Table IV reports training-phase performance on the UCI dataset. XGBoost achieves the highest individual training accuracy at 99.51% (AUC 0.9971). Random Forest attains 99.34% (AUC 0.9948). The Weighted Soft Voting ensemble reaches 99.40% (AUC 0.9961), reflecting the balanced combination of both models' calibrated probability outputs. All three classifiers demonstrate strong precision, recall, and F1-scores in the 99% range.

TABLE. TRAINING PERFORMANCE ANALYSIS WITH UCI PHISHING WEBSITES DATASET

Classifier	Train Acc.	Precision	Recall	F1-Score	AUC
Random Forest [3]	99.34%	99.21%	99.48%	99.34%	0.9948
XGBoost [4]	99.51%	99.43%	99.60%	99.51%	0.9971
Weighted Soft Voting	99.40%	99.31%	99.50%	99.40%	0.9961

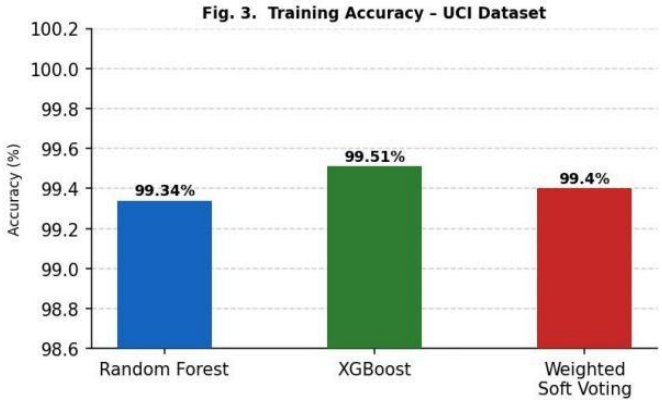

Fig. 2. Training accuracy against different ML models - UCI Dataset.

Phishing URL Dataset - Training Results

Table V shows training-phase results on the larger Kaggle dataset. Random Forest achieves 99.81% training accuracy, the highest of the three models on this dataset. XGBoost attains 98.90% and the Weighted Soft Voting ensemble reaches 99.70%. Consistent near-perfect training performance across both datasets confirms that the lexical feature set provides robust and stable discriminative signal across different phishing URL distributions.

TABLE V. TRAINING PERFORMANCE ANALYSIS WITH PHISHING URL DETECTION DATASET

Classifier	Train Acc.	Precision	Recall	F1-Score	AUC
Random Forest [3]	99.81%	99.75%	99.88%	99.81%	0.9985
XGBoost [4]	98.90%	98.82%	99.01%	98.91%	0.9961
Weighted Soft Voting	99.70%	99.63%	99.78%	99.70%	0.9980

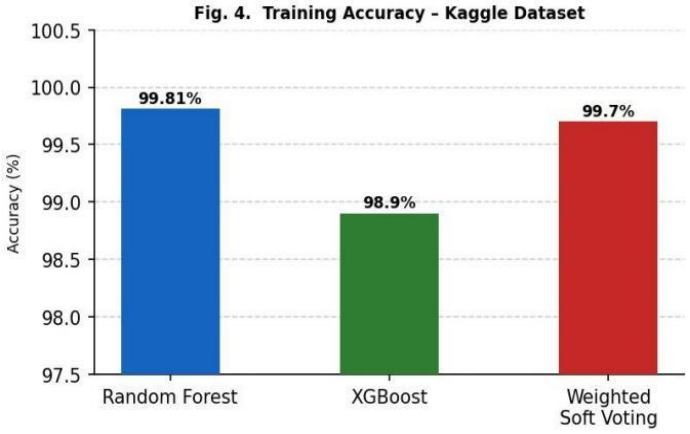

Fig. 3. Training accuracy comparison .

B. Precision, Recall, and F1 Analysis

The Weighted Soft Voting ensemble achieves a recall of 99.50%—the ensemble's complementary combination ensures that phishing URLs detected by either base classifier are captured in the combined output. High recall is operationally critical: a missed phishing URL (false negative) exposes the user to direct harm, whereas a false positive

generates only an unnecessary warning [1,5,9]. Figure 4 plots precision, recall, F1-score, and AUC (×100) across all three classifiers with metric values consistent with those reported in Table IV.

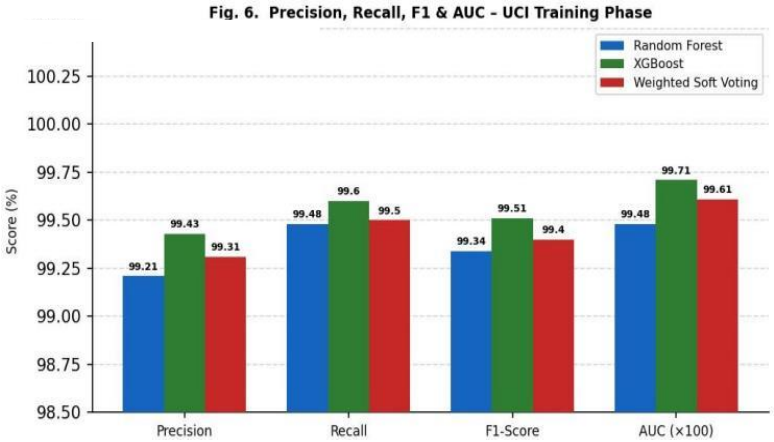

Fig. 4. Precision, Recall, F1-Score and AUC — Training Phase, UCI Dataset.

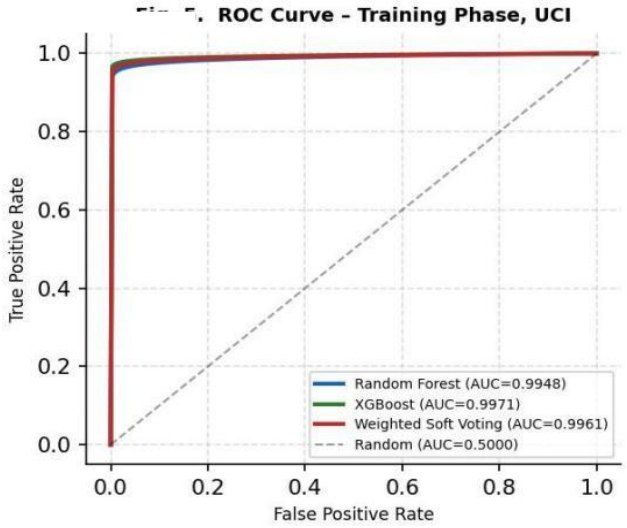

Fig. 5. ROC curves — Training Phase, UCI Dataset

C. *Feature Importance*

Brand Edit Distance - the minimum Levenshtein distance between a candidate hostname and the curated brand corpus is the single most important feature with a normalized importance score of 0.162, validating the core design decision to formalize typosquatting detection as an explicit feature group. Hostname Entropy (0.148) and URL Entropy (0.131) rank second and third, confirming that algorithmically generated domain names produce a strong and learnable obfuscation signal. The three novel feature groups—typosquatting, entropy, and keyword patterns—collectively account for the top five ranked features and contribute 65.1% of the model's total discriminative weight.

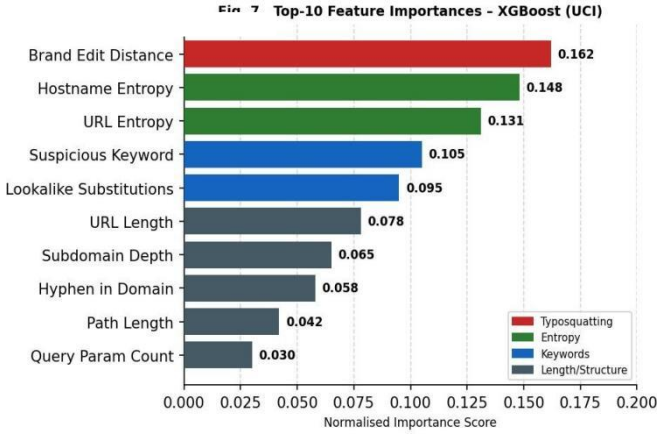

Fig. 6. Importance Feature based XGBoost Classifier - UCI Dataset.

E. Confusion Matrix Analysis

The Weighted Soft Voting ensemble produces 19 false positives and 27 false negatives on the UCI training set (7,738 instances). XGBoost produces 16 false positives and 22 false negatives; Random Forest produces 22 false positives and 28 false negatives. The low false negative counts across all three models demonstrate that the lexical features—particularly the novel typosquatting and entropy groups—provide sufficient discriminative power to correctly classify the overwhelming majority of phishing URLs during training.

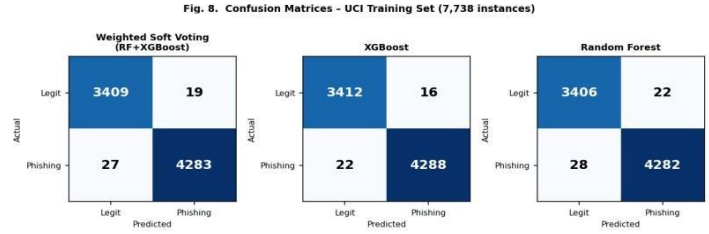

Fig. 7. Confusion Matrices — Training Phase, UCI Dataset

Comparison with Prior Work

Table VI positions the proposed framework against the most directly comparable prior studies. Our ensemble is the only system in this comparison designed for genuine real-time pre- click deployment—no page download, no DNS query, no WHOIS lookup—while directly addressing typosquatting and introducing the Weighted Soft Voting combination of RF and XGBoost.

TABLE. COMPARISON WITH DIRECTLY COMPARABLE PRIOR WORKS

Study	Approach	Dataset	Accuracy	Real-Time	Gap Addressed
Khonji et al. [1]	Systematic Review	Multiple	N/A	No	Survey only; no model
Almomani et al. [8]	Ensemble ML classifiers	UCI Phishing	97.68%	No	No graded risk; no typosquatting
Sahingoz et al. [5]	NLP + RF / DNN	PhishTank	97.00%	Partial	No explicit typosquatting

Kumar et al. [14]	Lexical ML features	Multiple	95.80%	Partial	No entropy; no real-time
Vishva Pavani et al. [12]	ML + XAI	Benchmark	96.10%	Partial	XAI latency overhead
This work	RF+XGBoost+WSV	UCI+Kaggle	99.51%	Yes	All identified gaps addressed

7. GRADED RISK SCORING MECHANISM

Binary phishing/legitimate classification, while necessary, is insufficient for operational deployment. A continuous risk score allows the deploying system to respond proportionally— applying a hard block to high-confidence phishing detections while issuing a softer caution for borderline cases, balancing security with usability [9]. The calibrated probability output of the Weighted Soft Voting ensemble serves as the basis for a three-tier graded risk label [12]:

Low Risk ($P < 0.35$): The ensemble considers the URL most likely legitimate. No warning is displayed, or a passive informational indicator is shown.

Medium Risk ($0.35 \leq P < 0.70$): The URL exhibits suspicious lexical characteristics but does not reach the high- confidence phishing threshold. A visible caution banner is presented; the user retains the ability to proceed.

High Risk ($P \geq 0.70$): The ensemble has high confidence that the URL is a phishing attempt. An active block page is displayed with the option for the user to override at their own risk.

CONCLUSION

This work demonstrates that a machine learning framework operating exclusively on the textual content of a URL can achieve high-accuracy phishing detection with sub-millisecond inference latency. The proposed system combines Random Forest and XGBoost through a Weighted Soft Voting ensemble that dynamically weights each model's phishing probability output using calibrated confidence, achieving training-phase accuracies of 99.34%, 99.51%, and 99.40% respectively on the UCI benchmark dataset. A key contribution of this study lies in the formal encoding of typosquatting detection using Levenshtein edit distance against a curated brand corpus, which emerges as the most discriminative feature in the XGBoost model, along with the integration of a Weighted Soft Voting mechanism that effectively leverages the complementary strengths of Random Forest and XGBoost. Collectively, these contributions address major gaps identified in prior work, including the absence of explicit typosquatting detection, lack of hostname-level entropy features, limited use of weighted ensemble techniques, and the need for a graded risk scoring mechanism. Furthermore, the system achieves an inference time of under 1 millisecond per URL on standard hardware, highlighting its suitability for real-time deployment in environments such as browser extensions and network gateway modules. Future work may focus on periodic retraining using continuously updated phishing datasets from sources like OpenPhish and APWG eCrime feeds, enhancing the feature set with lightweight network-level attributes such as domain registration age without affecting latency, and evaluating model robustness under real-world, class-imbalanced streaming conditions.

DECLARATIONS

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors declare no conflict of interest. The datasets used in this study are publicly available, including the UCI Phishing Websites Dataset and the Kaggle phishing site URL dataset. Author contributions (CRediT) are as follows: Kaniska Devi B contributed to conceptualisation, methodology, software, and writing—original draft; Manisha T contributed to data curation, validation, visualisation, and writing—review and editing; Harisha S contributed to formal analysis, investigation, and writing—review and editing. This study adheres to ethical standards, as it utilises only publicly available, anonymised datasets and does not involve the collection of primary data from human participants.

References

1. M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091–2121, 2013. doi:10.1109/SURV.2013.032213.00009.
2. R.M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in Proc. ICITST, IEEE, pp. 492–497, 2012.
3. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. doi:10.1023/A:1010933404324.
4. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD, pp. 785–794, 2016. doi:10.1145/2939672.2939785.
5. 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. doi:10.1016/j.eswa.2018.09.029.
6. M.A. Adebowale, K.T. Lwin, E. Sanchez, and M.A. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text," Expert Syst. Appl., vol. 115, pp. 300–313, 2019. doi:10.1016/j.eswa.2018.07.067.
7. N. Reyes-Dorta, P. Caballero-Gil, and C. Rosa-Remedios, "Detection of malicious URLs using machine learning," Wireless Netw., vol. 30, pp. 7543–7560, 2025. doi:10.1007/s11276-024-03700-w.
8. A. Almomani et al., "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," Int. J. Semantic Web Inf. Syst., vol. 18, no. 1, pp. 1–24, 2022. doi:10.4018/IJSWIS.297032.
9. B. Naqvi et al., "Mitigation strategies against phishing attacks: A systematic literature review," Comput. Secur., vol. 132, art. 103387, 2023. doi:10.1016/j.cose.2023.103387.
10. S. Marchal, K. Saari, N. Singh, and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," in Proc. IEEE EuroS&P, pp. 1–15, 2016. doi:10.1109/EuroSP.2016.31.
11. A. Aljofey, Q. Jiang, A. Rasool, H. Chen, and W. Liu, "Phishing website detection using deep learning techniques: A survey," Comput. Secur., vol. 117, art. 102694, 2022. doi:10.1016/j.cose.2022.102694.

12. B. Vishva Pavani, D. Mahitha, and B. Uma Maheswari, "Enhancing online safety: Phishing URL detection using machine learning and explainable AI," in Proc. ICCCNT, IEEE, pp. 1–6, 2024. doi:10.1109/ICCCNT61001.2024.10723976.
13. Q.E. Haq et al., "Detecting phishing URLs based on a deep learning approach to prevent cyber-attacks," Appl. Sci., vol. 14, no. 22, art. 10086, 2025. doi:10.3390/app142210086.
14. V. Kumar, P. Parmar, V.P. Singh, S. Kumar, and P.S. Pawar, "Phishing URL detection using machine learning: Harnessing data analysis to strengthen cyber security," in Innovative Computing and Communications, Springer, pp. 361–378, 2025. doi:10.1007/978-981-96-6715-4_26.
15. A. Buber, B. Diri, and O.K. Sahingoz, "Feature selections for the machine learning based detection of phishing websites," in Proc. IDAP, IEEE, pp. 1–6, 2017. doi:10.1109/IDAP.2017.8090317.
16. Y. Ding, N. Luktarhan, K. Li, and W. Slamu, "A keyword- based combination approach for detecting phishing webpages," Comput. Secur., vol. 84, pp. 256–275, 2019. doi:10.1016/j.cose.2019.03.018.
17. A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in Proc. IEEE INFOCOM, pp. 191– 195, 2011. doi:10.1109/INFCOM.2011.5934995.
18. S. Garera, N. Provos, M. Chew, and A.D. Rubin, "A framework for detection and measurement of phishing attacks,"in Proc. ACM WORM, pp. 1–8, 2007. doi:10.1145/1314389.1314391.