IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

Fraudulent Website Detection Using XGBoost and Lexical URL Features with Explainable Risk-Based Alert Presentation

Tarsha P[2], S. Anitha Jebamani[1], Snifa Rose Vilson[2], Neha Raj[2], Vandhana Prabhakar[2]

Assistant Professor, Department of CSE (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India[1]

Student, Department of CSE (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India[2]

Abstract: Phishing websites imitate legitimate services to steal credentials, and blacklist-based defences are slow to cover newly created domains. This paper presents a fraudulent website detection system that classifies a URL as legitimate or potentially phishing from 19 lexical features alone, so that no web page has to be downloaded. An Extreme Gradient Boosting (XGBoost) classifier is trained on the PhiUSIIL Phishing URL dataset (235,795 records); each prediction is converted into a phishing probability, a risk score and level, and risk-attribution cards, and the pipeline is served through a Streamlit application for single and batch analysis. On a held-out test set of 12,000 unseen URLs the model reaches 99.60% accuracy, 99.44% precision, 99.87% recall (legitimate class as positive), a 99.65% F1-score and a ROC-AUC of 0.9984, with 39 phishing URLs missed and 9 legitimate URLs flagged, and it trains in 0.49 s on a CPU. Three features carry almost all of the importance and no independent data set was used, so the result is specific to this benchmark and requires external validation.

Keywords: Phishing detection, fraudulent websites, URL classification, lexical features, XGBoost, gradient boosting, explainable machine learning, risk scoring, PhiUSIIL

I. INTRODUCTION

Websites are central to banking, shopping, communication and education, and the same growth has widened the attack surface for phishing, in which a fraudulent website imitates a genuine one to make users reveal usernames, passwords, banking details and other personal data. Such sites are cheap to create and are often abandoned within hours or days. Defences that depend on previously reported URLs, such as blacklists and manual verification, are precise for known threats but slow to react to newly created ones [3], [4].

Machine-learning (ML) detectors learn the structural patterns that separate phishing addresses from legitimate ones and can therefore score URLs they have never seen. Approaches that analyse page content can be accurate, but they require the page to be fetched, which is slow and exposes the analysis system to the malicious site. Approaches that rely on the URL string alone are fast, safe to run and applicable before a user visits the page [3], [5]. Two problems limit their practical use: classifier outputs are usually presented as a bare label without an explanation, and reported accuracies on a single benchmark say little about behaviour on other data.

This paper addresses these problems with an integrated pipeline that combines lexical feature extraction, an XGBoost classifier [2], probability-based risk assessment and risk-attribution explanations, implemented as a web application for single and batch URL analysis [10]. It is trained and evaluated on the PhiUSIIL Phishing URL dataset [1], [9]. The paper reports the results together with an explicit account of what they do and do not show.

II. RELATED WORK

Blacklist and whitelist look-ups remain the operational baseline for phishing defence but cannot cover new or short-lived sites. Ma *et al.* [3] showed that malicious websites can be identified from the lexical and host-based properties of their URLs rather than from blacklists alone, and Sahoo *et al.* [4] surveyed ML techniques for malicious URL detection, covering feature representations, classification algorithms and open challenges. Sahingoz *et al.* [5] compared several classifiers on URL features for phishing detection, and Mohammad *et al.* [6] predicted phishing websites with a self-structuring neural network, a neural alternative to tree-based methods.

Among tree-based learners, Random Forests [7] average bagged decision trees to reduce over-fitting, and XGBoost [2] is a scalable gradient-boosting system with a regularised objective and optimised tree construction. Prasad and Chandra [1] introduced the PhiUSIIL framework and the accompanying large-scale phishing URL dataset, which is

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

distributed through the UCI Machine Learning Repository [9]. Scikit-learn [8] supplies the datasplitting and evaluation utilities, and Streamlit [10] supplies the interface framework. Table I compares the main families of phishing-detection techniques qualitatively; reported accuracies are not compared because they are not comparable across data sets and protocols.

TABLE I QUALITATIVE COMPARISON OF PHISHING DETECTION APPROACH FAMILIES

Approach	Strengths	Limitations
Blacklist / whitelist lookup	Very fast; few false positives on listed URLs	Cannot detect new or short-lived sites; needs constant updates
Approach	**Strengths**	**Limitations**
Heuristic / rule-based	Simple and interpretable	Rules are easy to evade and hard to maintain
Content-based ML	Rich signals from HTML, text and images	Page must be downloaded (risky, slow); heavy computation
Deep learning on raw URLs [6]	No manual feature design	Needs large data and GPUs; poor interpretability
Lexical ML with XGBoost (this work)	Fast, light-weight, explainable; no page visit	Limited to what the URL string reveals

The reviewed studies confirm the value of URL-based learning, but an integrated system that joins feature extraction, an efficient classifier, risk scoring and per-alert explanation in an interface for both single and batch analysis is less often described. This work targets that combination.

III. PROBLEM DEFINITION AND CONTRIBUTIONS

A. Problem Definition

Each URL is described by a vector of 19 lexical features and a binary label $y \in \{0, 1\}$; in the dataset's own encoding, 1 denotes a legitimate and 0 a phishing URL. For every submitted URL the system outputs a binary decision, a phishing probability, a risk score with a risk level (Low or High), and risk-attribution cards. Detection uses the URL string only: no page content, host information or reputation data is consulted, and no capability to detect previously unseen phishing campaigns is claimed or evaluated.

B. Contributions

(1) An integrated pipeline, implemented as a working web application, of lexical feature extraction, XGBoost classification, probability-based risk scoring and risk-attribution explanations for single and batch URLs. (2) A onetime evaluation on 12,000 unseen records using accuracy, precision, recall, F1-score, ROC-AUC, a confusion matrix, class-wise results and feature-importance analysis. (3) A qualitative comparison with other common classifiers and a functional test of the application. (4) An explicit account of the limitations of the evidence, including the concentration of feature importance in three features and the absence of external validation.

IV. DATA SET AND PREPROCESSING

A. PhiUSIIL Phishing URL Dataset

The PhiUSIIL Phishing URL (Website) Dataset [1], [9] contains 235,795 website records described by 54 original features. It comprises 134,850 legitimate URLs (about 57%) and 100,945 phishing URLs (about 43%). The data were obtained with the Python ucimlrepo library and stored in CSV format with Pandas. Although the original features include website-level attributes, only the URL string is used as model input in this work.

B. Preprocessing

The dataset was inspected for its dimensions, feature names, data types and target variable. Missing values and duplicate records were identified and handled, and numerical and categorical attributes were checked for compatibility with the XGBoost classifier. The target variable was separated from the input features, and raw text fields such as the complete URL and domain name were not used directly but transformed into numerical features. The feature structure and column order are fixed so that training and inference use identical inputs.

C. Lexical Feature Extraction

Nineteen lexical features are computed from each URL string (Table II). They capture URL and domain size, character composition, path depth, sub-domain structure, query-string structure, the use of HTTPS, the presence of an IP-address host or an "@" symbol, and the occurrence of sensitive keywords. The classifier uses combinations of these features rather than any single indicator.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

TABLE II LEXICAL FEATURES EXTRACTED FROM A URL

No.	Feature	Description
1	url_length	Total number of characters in the URL
2	domain_length	Number of characters in the host / domain part
3	path_length	Number of characters in the path part
4	num_digits	Count of numeric characters
5	num_letters	Count of alphabetic characters
6	num_dots	Count of "." (indicates sub-domain depth)
7	num_hyphens	Count of "-" (common in spoofed domains)
No.	**Feature**	**Description**
8	num_slashes	Count of "/" (path depth)
9	num_at	Count of "@" (can hide the real host)
10	num_question_marks	Count of "?" (query string)
11	num_equals	Count of "=" characters
12	num_percent	Count of "%" (URL-encoded characters)
13	num_query_params	Number of query parameters
14	num_special_chars	Count of special characters
15	num_subdomains	Number of sub-domain labels
16	has_https	Whether the URL uses HTTPS
17	has_ip	Whether the host is an IP address
18	has_at	Whether the URL contains "@"
19	suspicious_keyword_count	Occurrences of sensitive words (e.g., login, verify, secure, bank)

V. METHODOLOGY

A. XGBoost Classifier

XGBoost [2] builds an additive ensemble of decision trees in which each new tree corrects the errors of the current ensemble. It handles non-linear relationships and interactions among features, applies regularisation to control model complexity, supports histogram-based tree construction for efficient training, and provides probability outputs and feature-importance scores, which suit the structured URL features used here. For n training samples the model minimises the regularised objective (1), where l is a differentiable loss, f_k is the k-th tree and the complexity penalty is given by (2), with T leaves and leaf weights w.

$$L = \Sigma_{i=1}^{n} l(y_i, \hat{y}_i) + \Sigma_{k=1}^{K} \Omega(f_k) \quad (1)$$

$$\Omega(f) = \gamma T + \tfrac{1}{2} \lambda \|w\|^2 \quad (2)$$

Splits are chosen from first- and second-order gradient statistics of the loss, and each new tree's contribution is scaled by the learning rate. With the binary:logistic objective the raw ensemble score $\hat{y}$ is converted into the phishing probability (3), and a URL is classified as phishing when $p \geq \tau$ and as legitimate otherwise, where τ is the classification threshold.

$$p = 1 / (1 + e^{-\hat{y}}) \quad (3)$$

B. Risk Scoring and Classification

The predicted probability is expressed as a risk score, shown as a percentage in the application. If the risk score is greater than or equal to the selected threshold, the URL is classified as Potential Phishing (High Risk); otherwise it is classified as Legitimate (Low Risk). The threshold is selected by the user in the application interface.

C. Risk Attribution

An explainability engine generates risk-attribution cards that highlight the URL characteristics contributing to a classification, for example an IP-address hostname, missing HTTPS, excessive sub-domains or targeted keywords, and the application also displays the extracted feature values for the submitted URL. The cards are intended to show an analyst why a URL was flagged rather than only that it was flagged.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

D. Evaluation Metrics

Performance is measured with accuracy, precision, recall, F1-score and ROC-AUC, computed from the numbers of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) as in (4)–(6); a confusion matrix and a ROC curve complement them. Overall precision, recall and F1-score are reported with the legitimate class (label 1) as the positive class, and class-wise values are reported for both classes. Because a phishing site accepted as legitimate exposes users to harm, the number of missed phishing URLs is examined explicitly.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

$$Precision = TP / (TP + FP), \ Recall = TP / (TP + FN) \quad (5)$$

$$F1 = 2 \cdot Precision \cdot Recall / (Precision + Recall) \quad (6)$$

VI. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Fig. 1(a) shows the architecture and Table IV the per-URL inference procedure. The system has two phases. In the offline training phase the PhiUSIIL data are cleaned, 19 lexical features are extracted, the XGBoost model is trained with histogram-based tree construction, evaluated on 12,000 unseen records and saved with Joblib together with its feature-column order. In the online inference phase a user submits a single URL or a batch of URLs through the Streamlit interface, which has three tabs and a sidebar; the same 19 features are extracted, the saved model returns a phishing probability, a risk-scoring module assigns the class, and the explainability engine generates the riskattribution cards shown on the result dashboard. Separating the phases allows the model to be reused for real-time analysis without retraining. Fig. 1(b) shows the scanning flow: an invalid URL produces an error message and a request for a valid one, and a valid URL proceeds through feature extraction, prediction and risk assessment.

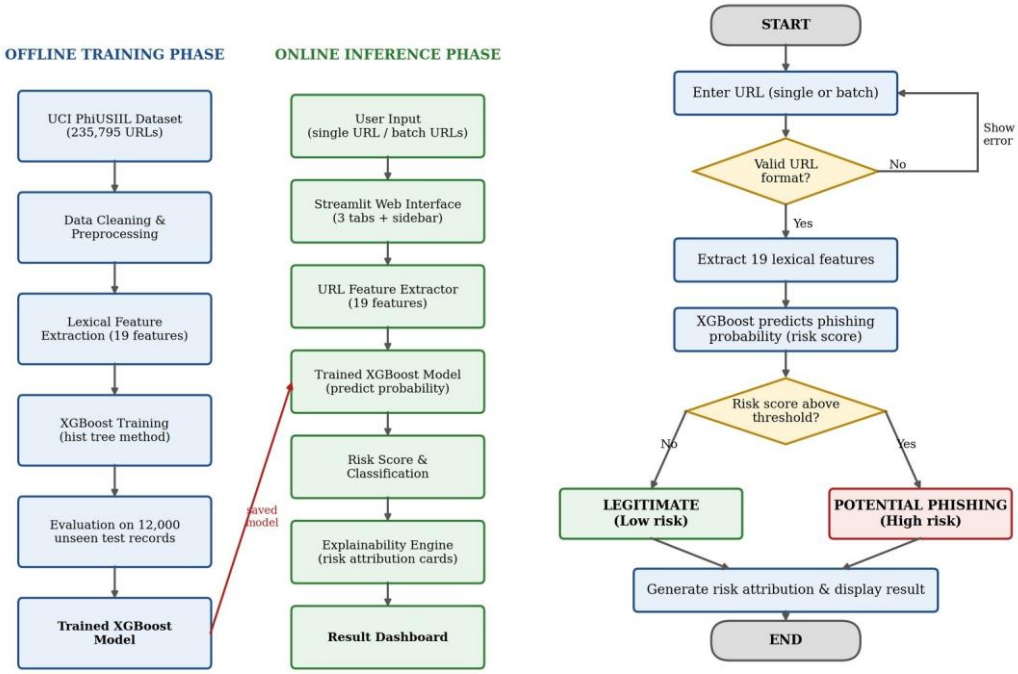


Fig. 1 System architecture (a) and URL scanning flow (b). Left: offline training and online inference phases. Right: validation, feature extraction, prediction, risk assessment and display for each submitted URL.

The system consists of eight modules: (1) data collection and preprocessing; (2) lexical feature extraction, applied identically in training and inference; (3) XGBoost model training; (4) model evaluation and saving; (5) user input and URL validation; (6) online prediction and risk classification; (7) the explainability module; and (8) the Streamlit interface and result dashboard, which presents the classification, phishing probability, risk score and explanations. The system is written in Python with the libraries listed in Table III, and the application is hosted on Streamlit Community Cloud.

TABLE III TOOLS AND TECHNOLOGIES

Tool / library	Purpose
Python	Data processing, feature extraction, training and prediction
XGBoost	Binary classification of legitimate and phishing URLs
scikit-learn	Dataset splitting and evaluation metrics

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

Pandas, NumPy	Loading, cleaning and numerical handling of the dataset
Matplotlib, Seaborn	Evaluation plots and dataset visualisation
Joblib	Saving and loading the trained model
Streamlit	Interactive single- and batch-URL web interface
UCI ML Repository	Source of the PhiUSIIL dataset

TABLE IV PER-URL INFERENCE PROCEDURE (ALGORITHM 1)

Step	Operation
1	Validate the submitted URL format; if invalid, display an error message and request a valid URL
2	x ← the 19 lexical features of the URL, in the column order used for training
3	p ← phishing probability from the saved XGBoost model
4	r ← risk score expressed from p; class ← Potential Phishing (High Risk) if $r \geq \tau$, otherwise Legitimate (Low Risk)
5	E ← risk-attribution cards for the URL characteristics contributing to the result
6	Display the class, probability, risk score and E on the dashboard

VII. EXPERIMENTAL SETUP

The model was trained on the preprocessed PhiUSIIL data. The data were partitioned with train_test_split() from Scikit-learn using stratified sampling, which preserves the proportion of legitimate and phishing samples in both subsets, with a random state of 42 for reproducibility. The classifier was trained with the baseline hyperparameters in Table V and evaluated once on a held-out set of 12,000 unseen records (5,120 phishing and 6,880 legitimate URLs) using predict() and predict_proba(). The metrics are those defined in Section V-D. Training took 0.49 s on a CPU.

TABLE V TRAINING CONFIGURATION

Parameter	Value
Algorithm	XGBoost classifier (gradient-boosted trees)
Tree construction	hist (histogram-based)
Objective / evaluation metric	binary:logistic / logloss
Estimators / maximum depth	300 / 8
Learning rate	0.05
Row / column subsampling	0.8 / 0.8
Random state	42 (stratified split)
Input features	19 lexical URL features
Dataset	PhiUSIIL, 235,795 records
Held-out test set	12,000 unseen records (5,120 phishing, 6,880 legitimate)
Training time	0.49 s (CPU)

VIII. RESULTS AND DISCUSSION

A. Overall Performance

Table VI gives the overall results on the 12,000 unseen records. The model classified 11,952 of them correctly. Precision, recall and F1-score in this table use the legitimate class as the positive class.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

TABLE VI OVERALL PERFORMANCE ON THE 12,000-RECORD TEST SET

Metric	Value	Basis
Accuracy	99.60%	11,952 correct decisions out of 12,000
Precision	99.44%	6,871 of the 6,910 URLs predicted legitimate are legitimate
Recall	99.87%	6,871 of the 6,880 legitimate URLs are accepted
F1-score	99.65%	Harmonic mean of precision and recall
ROC-AUC	0.9984	Area under the ROC curve

B. Confusion Matrix and Class-wise Results

Fig. 2 and Table VII show the confusion matrix. Of the 5,120 phishing URLs, 5,081 were detected and 39 were missed; of the 6,880 legitimate URLs, 6,871 were accepted and 9 were wrongly flagged, giving 48 errors in 12,000 predictions. The phishing miss rate is 0.76% (39 of 5,120) and the false-alarm rate on legitimate URLs is 0.13% (9 of 6,880). Table VIII gives the class-wise report: for phishing URLs the precision is 99.82% and the recall 99.24%, and for legitimate URLs the precision is 99.44% and the recall 99.87%. Both error types are low, which matters because missed phishing sites expose users to attack while false alarms reduce trust in the tool. The ROC curve in Fig. 2 rises almost vertically to a true-positive rate close to 1 at a very low false-positive rate, consistent with the ROC-AUC of 0.9984.

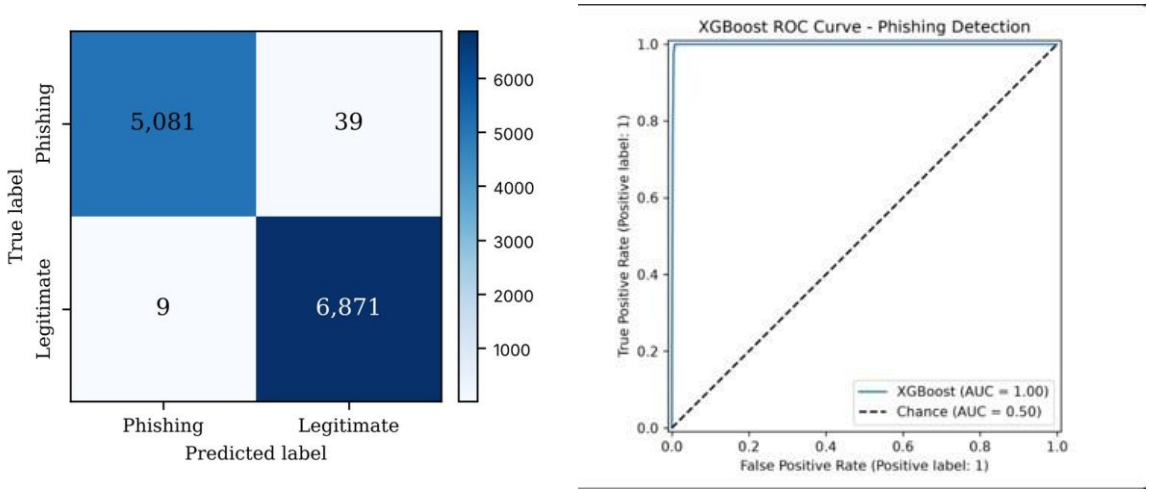


Fig. 2 Confusion matrix (left) and ROC curve (right, AUC = 0.9984) of the XGBoost model on the 12,000 test records.

TABLE VII CONFUSION MATRIX ON THE 12,000 TEST RECORDS (ROWS: TRUE CLASS)

	Predicted phishing	Predicted legitimate	Total
True: phishing	5,081	39	5,120
True: legitimate	9	6,871	6,880
Total	5,090	6,910	12,000

TABLE VIII CLASSIFICATION REPORT OF THE XGBOOST MODEL

Class	Precision	Recall	F1-score	Support
Phishing	99.82%	99.24%	99.53%	5,120
Legitimate	99.44%	99.87%	99.65%	6,880

C. Feature Importance

Fig. 3 ranks the 19 features by XGBoost importance. Three features dominate: num_slashes (approximately 0.37), has_https (approximately 0.34) and path_length (approximately 0.27), values read from the chart. Together they account for almost all of the importance, and each of the remaining 16 features contributes less than 0.01. The model therefore

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

relies mainly on path structure and the use of HTTPS. Such a concentration suggests that part of the high accuracy may reflect URL-construction patterns specific to this data set, which motivates external validation.

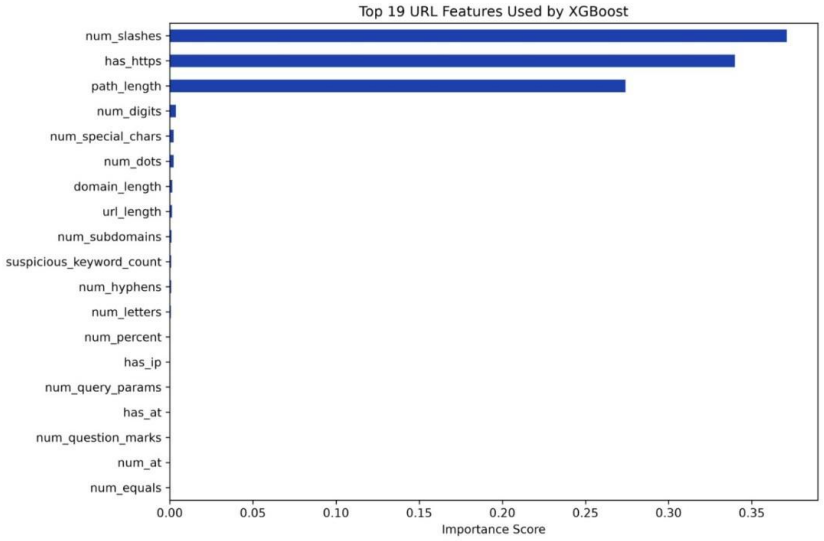


Fig. 3 Feature importance of the 19 URL features used by XGBoost.

D. Comparison with Other Algorithms

Table IX compares XGBoost qualitatively with other common classifiers, based on their general characteristics reported in the literature [7]. It is not an experimental comparison: the five algorithms were not trained and tested under identical conditions in this study, and a controlled quantitative comparison is left for future work.

TABLE IX XGBOOST VERSUS OTHER CANDIDATE ALGORITHMS (QUALITATIVE)

Algorithm	Behaviour on URL features and remark
Logistic regression	Linear boundary, needs scaled inputs; fast baseline but cannot model feature interactions well
Decision tree	Captures non-linear rules and is readable; single trees over-fit and are unstable
Random forest	Strong ensemble of independent trees; larger and slower to train than histogram boosting
SVM	Powerful with scaling and kernel tuning; training cost grows quickly with very large data sets
XGBoost (used)	Sequential boosting with regularisation; good accuracy-to-speed trade-off, trained in 0.49 s

E. Application Testing

The Streamlit application was tested with single and batch URL inputs (Fig. 4). It validates the input, extracts the features, returns the classification, probability, risk score and explanation cards, and shows an error message for invalid input. Table X lists five test URLs; all five were classified as expected, with risk scores of 0.49% for the legitimate URLs and above 99% for the phishing-style URLs. This is a functional demonstration on five examples and not a statistical evaluation.

TABLE X APPLICATION TEST RESULTS

URL	Expected	Prediction	Risk score
https://www.google.com	Legitimate	Legitimate	0.49%
https://www.github.com	Legitimate	Legitimate	0.49%
http://192.168.1.1/update-account/bank-verify-secure	Phishing	Potential phishing	99.97%

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2026

DOI: 10.17148/IJIREEICE.2026.141104

http://www.shprakserf.gq	Phishing	Potential phishing	99.81%
http://secure-paypal-login.account-verification-alert.com/ signin	Phishing	Potential phishing	99.97%

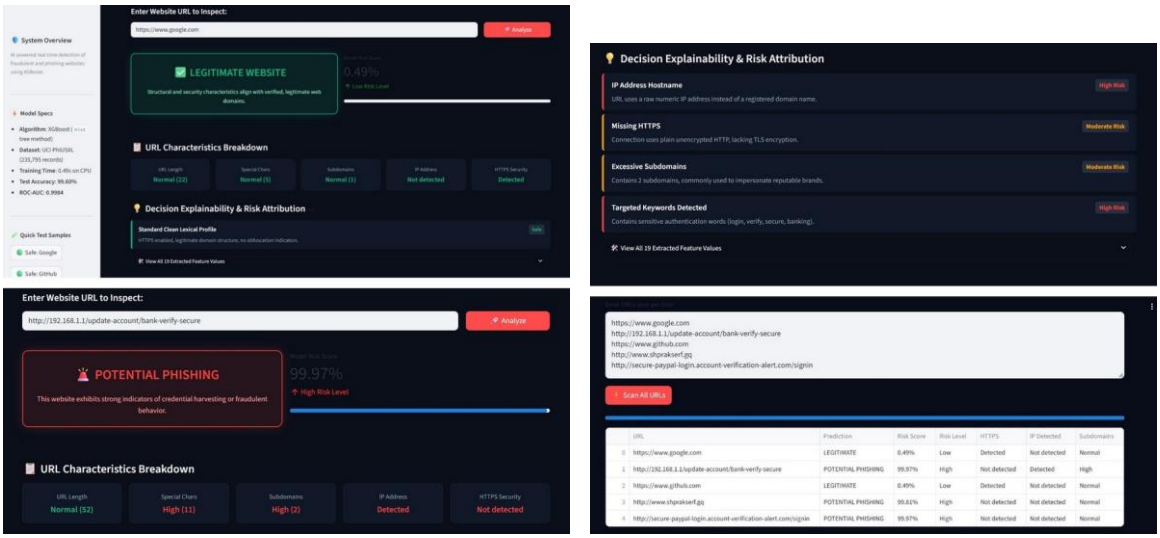


Fig. 4 Streamlit interface. Left: legitimate result for a safe URL (top) and potential phishing result for a suspicious URL (bottom). Right: risk-attribution cards (top) and batch URL analysis results (bottom).

F. Discussion

The results show that 19 inexpensive lexical features are sufficient for a gradient-boosted classifier to separate phishing from legitimate URLs with few errors on this test set, with a training time under one second. The probability, risk score and attribution cards make each prediction easier to interpret than a bare label. The predictions should nevertheless be treated as risk assessments rather than proof of safety, because websites with malicious content may not show obvious lexical indicators.

IX. LIMITATIONS AND THREATS TO VALIDITY

1) External Validity: The model was trained and tested on a single data set. No cross-data-set evaluation or test on independent, continuously updated phishing and legitimate URL sources was performed, so performance on URL patterns that differ from those in PhiUSIIL may vary. Periodic evaluation on newer samples would be needed to maintain effectiveness.

2) Internal Validity: The reported results come from one held-out test set and one trained model; no confidence intervals or repeated runs with different random seeds were computed. The hyperparameters are baseline values and were not systematically tuned. Feature importance is concentrated in three features (num_slashes, has_https, path_length), which suggests that the very high scores may depend on patterns specific to this data set. The comparison with other algorithms is qualitative.

3) Construct Validity: The system relies on 19 lexical features and does not analyse page content, forms, images, scripts or hosting information, so phishing sites with carefully designed URLs may be missed. The two risk levels and the threshold are design choices that were not validated with users, and the predicted probability was not assessed for calibration. False positives and false negatives still occur: 48 of the 12,000 test URLs were misclassified.

4) Operational Limitations: The application was tested functionally on five URLs and has not been evaluated under production traffic. Online prediction requires the same 19 features and column order as training, so any inconsistency between the training and inference pipelines would reduce reliability.

X. CONCLUSION AND FUTURE WORK

This paper presented a fraudulent website detection system that uses 19 lexical URL features and an XGBoost classifier trained on the PhiUSIIL dataset. On 12,000 unseen records it reached 99.60% accuracy, 99.44% precision, 99.87% recall, a 99.65% F1-score and a ROC-AUC of 0.9984, with 39 phishing URLs missed and 9 legitimate URLs flagged. The model is integrated into a Streamlit application that provides single and batch analysis, risk scoring and risk-attribution explanations, offering a content-free first line of defence against phishing. The evidence is limited to one data set and one test split, and the importance of three features dominates the model.

DOI: 10.17148/IJIREEICE.2026.141104

Future work will validate the model on independent and continuously updated URL collections; combine URL features with web-page content analysis and with domain-registration, DNS, SSL and hosting information; connect the system to threat-intelligence sources such as PhishTank; compare XGBoost experimentally with Random Forest, LightGBM, neural and hybrid models under identical conditions; retrain periodically as phishing patterns change; improve explainability with SHAP; offer configurable multi-level risk thresholds (low, medium, high); and extend the application with scan history, downloadable batch reports, authentication and wider cloud deployment.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, for its support.

REFERENCES

[1]. A. Prasad and S. Chandra, “PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning”, Comput. Secur., vol. 136, p. 103545, 2024.

[2]. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system”, in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[3]. J. Ma, L. K. Saul, S. Savage and G. M. Voelker, “Beyond blacklists: Learning to detect malicious web sites from suspicious URLs”, in Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2009, pp. 1245–1254.

[4]. D. Sahoo, C. Liu and S. C. H. Hoi, “Malicious URL detection using machine learning: A survey”, arXiv:1701.07179, 2017.

[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.

[6]. R. M. Mohammad, F. Thabtah and L. McCluskey, “Predicting phishing websites based on self-structuring neural network”, Neural Comput. Appl., vol. 25, pp. 443–458, 2014.

[7]. L. Breiman, “Random forests”, Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[8]. F. Pedregosa et al., “Scikit-learn: Machine learning in Python”, J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[9]. UCI Machine Learning Repository, “PhiUSIIL Phishing URL Dataset”. [Online]. Available: https://archive.ics.uci.edu

[10]. Streamlit Documentation. [Online]. Available: https://docs.streamlit.io