International Journal of Safety and Security Engineering
Vol. 16, No. 9, September, 2026, pp. 2177-2188
Journal homepage: http://iieta.org/journals/ijsse

A Risk-Aware and Explainable Machine Learning Framework for Phishing Website Detection Using Structural Website Features

Ahmed Eskander Mezher[1,2]

[1] Business Information Technology Department (BIT), Business Informatics College, University of Information Technology and Communications, Baghdad 10045, Iraq
[2] School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok 06010, Malaysia

Corresponding Author Email: ahmed.mezher@uoitc.edu.iq

Copyright: ©2026 The author. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.160920

Received: 28 July 2026
Revised: 14 September 2026
Accepted: 20 September 2026
Available online: 30 September 2026

Keywords:
phishing website detection, risk-aware machine learning, explainable artificial intelligence, Random Forest, false-negative rate, structural website features

ABSTRACT

Phishing websites are considered a serious cybersecurity threat because they imitate legitimate online services and deceive users into disclosing sensitive information. This study presents a risk-aware and explainable machine learning evaluation for phishing website detection using structural website features. A publicly available benchmark dataset containing 11,055 website instances and 30 structural features was used. Several supervised machine learning classifiers were evaluated, including Random Forest, Support Vector Machine (SVM), Logistic Regression, Decision Tree, K-Nearest Neighbors, Naive Bayes, and XGBoost. The models were assessed using conventional and risk-aware metrics, including accuracy, macro-F1, phishing recall, phishing false-negative rate (FNR), and Area Under the Receiver Operating Characteristic Curve (ROC-AUC). Validation-based model comparison identified Random Forest as the classifier with the highest observed macro-F1, while the complete 30-feature configuration was retained because it outperformed the evaluated reduced subsets on the validation partition. On the untouched final test partition, the default Random Forest threshold achieved an accuracy of 97.01% and a macro-F1 of 96.97%. The validation-selected risk-aware threshold increased phishing recall to 97.65% and reduced the phishing FNR to 2.35%, although the legitimate false-positive rate (FPR) increased to 4.79%. Explainability analysis using Random Forest feature importance and permutation importance confirmed that Secure Sockets Layer (SSL)-related features, Uniform Resource Locator (URL) anchor behavior, web traffic, subdomain structure, and hyperlink-related indicators were among the most influential predictors. Within the evaluated benchmark, the findings show that a leakage-free, risk-aware, and explainable evaluation protocol can provide reproducible phishing-detection results and interpretable security-related insights. However, evaluation using newer temporally separated or external datasets remains necessary to establish current real-world generalizability.

1. INTRODUCTION

The daily activities of individuals, businesses, and organizations have been revolutionized by rapid developments in internet services and digital technologies. The new norm is the adoption of a prominent digital lifestyle that encompasses online banking, cloud computing, e-commerce platforms, and social networking systems. However, certain malicious actors have often succeeded in exploiting security loopholes. Phishing is a recognized type of cybercrime that continues to be a dominant and significant threat with an adverse impact on global organizations [1, 2]. Phishing is a method used by hackers to trick users into sharing sensitive information, including personal identity data, financial information, usernames, and passwords [3]. It's a form of cybercrime that uses bogus e-mail, deceptive messages, or phishing sites to conduct attacks under the guise of authentic businesses, including banks, payment systems and social media platforms, as detailed in continued security reports [4]. Phishing has been identified by the Verizon Data Breach Investigations Report as a key factor in data breaches [5]. Similarly, the European Union Agency for Cybersecurity has identified phishing as one of the most common attack methods of cyber actors to compromise systems and sensitive data [6]. These threats have led to the need for accurate and secure proactive detection methods that can identify fraudulent websites and keep users from being scammed into clicking links on fraudulent sites. Interestingly, phishing identification has relied on the use of blacklist-based techniques and heuristic analysis. Blacklist-based methods involve comparing the Uniform Resource Locators (URLs) with past reports of malicious sites. They are typically employed in browsers and security products due to their low complexity and low computational cost of their computation. One of the most serious drawbacks is that newly established phishing web sites that are not reported or included in phishing blacklist databases are a major challenge in

detecting phishing [7]. Blacklist-based methods are also not effective against real-time attacks, as phishing sites are quickly generated and usually do not last long. Heuristic algorithms, which can detect suspicious URLs and webpage structures, have helped to detect phishing websites. Suspicious indicators can consist of odd domain names, odd URL patterns, and hyperlinks that appear suspicious or anomalies in the activity of a particular webpage. The techniques are based on rules and signals created by hand. They also need to be continually updated because new phishing techniques make them less effective in fast-changing threat landscapes [8, 9]. As a result, machine learning algorithms are recognized by researchers as a solution to phishing website identification issues. Unlike blacklist and heuristic methods, machine learning models can extract discriminative features from labeled datasets containing both phishing and genuine websites. Such systems offer high detection potential and greater flexibility in identifying previously unknown attacks. This is possible through the development of classifiers on extracted website characteristics [10-12].

Consequently, machine learning has emerged as a top-tier phishing detection research topic. In machine learning-based phishing identification, feature extraction is deemed a crucial part of the process. To detect malicious activities, URLs, website architecture, and host-associated data are used as signals. For the detection of structural abnormalities on phishing websites, lexical indicators like URL length, special characters, token composition, and dubious prefixes or suffixes are commonly utilized. Moreover, Domain Name System (DNS)-related qualities, domain registration features, and Secure Sockets Layer (SSL)-associated data are examples of host-based data used to assess website reliability and validity [13, 14]. All of these enable machine learning models to accurately differentiate between phishing and authentic websites. Numerous machine learning techniques have been utilized for the identification of phishing attempts. To this end, Support Vector Machine (SVM), Random Forest, and Logistic Regression are employed for the feature-based classification of phishing and harmful URLs. These algorithms have been previously shown to have high predictive accuracy. Specifically, SVM is reported to be successful in high-dimensional feature space and Random Forest has been shown to be reliable and resistant to overfitting [15]. Moreover, ensemble methods have been proven to be promising as well by combining multiple models to improve classification accuracy and reliability [16].

But finding the best traditional machine learning approach for phishing detection remains a practical challenge. Considering the common issues in existing studies, such as the use of a wide range of datasets, feature representations, preprocessing methods, and evaluation settings, direct comparison of the reported results is difficult. The present study aimed to resolve these issues. Furthermore, conventional machine learning models have been shown to be often easier to understand, computationally efficient, and more convenient to implement in resource-constrained environments. Therefore, a systematic comparison of the performance of widely used conventional classifiers in similar experimental settings is still valuable for use in research and applications. In the current study, several supervised machine learning classifiers, such as Random Forest, SVM, Logistic Regression, Decision Tree, K-Nearest Neighbors, Naive Bayes and XGBoost, are evaluated on a publicly available benchmark dataset of structural features of phishing websites. The classifiers are evaluated using similar experimental conditions and employing commonly used classification performance metrics like accuracy, precision, recall and F1-score. Further, the components of the website for phishing attacks are studied and the Random Forest framework is used, which has achieved remarkable results. Several previous studies have used machine learning-based algorithms to detect phishing websites. Most of these studies focused on the percentage of total accuracy of the classification. However, the accuracy of applications in cybersecurity may not be enough, since it is much costlier to identify a fake website as real. Thus, this work builds upon the traditional model comparison with the addition of risk-aware evaluation, false-negative analysis, and feature-level explainability. This enables the proposed approach to be used to not only determine which model is best, but also the structural website indicators that are most useful in detecting phishing and reducing the number of missed cases.

The aim and direction of this research are guided by the following research questions:

RQ1: Which supervised machine learning classifier provides the best performance for phishing website detection under consistent experimental settings? RQ2: How effective are the tested classifiers in reducing phishing false negatives, where phishing websites are incorrectly classified as legitimate? RQ3: Can compact subsets of important structural website features maintain competitive phishing detection performance compared with the full feature set? RQ4: How does threshold optimization affect the trade-off between phishing detection sensitivity and false alarms on legitimate websites? RQ5**:** Which structural website features contribute most significantly to phishing website detection according to feature-importance and permutation-importance analysis?

Unlike simple accuracy-based phishing detection studies, this research emphasizes a risk-aware and explainable evaluation strategy. The study focuses not only on which classifier achieves the highest overall performance, but also on how effectively the models reduce the number of phishing websites misclassified as legitimate. In addition, compact feature subset analysis and threshold optimization are used to examine the trade-off between model simplicity, phishing detection sensitivity, and false-alarm control. This provides a risk-aware benchmark assessment that reflects the asymmetric security cost of missed phishing websites, without assuming that performance on the benchmark transfers directly to contemporary operational environments.

The major contributions of this research are outlined below:

•This study presents a risk-aware and explainable machine learning evaluation for phishing website detection using structural website features.

•This study compares several supervised machine learning classifiers, including Random Forest, SVM, Logistic Regression, Decision Tree, K-Nearest Neighbors, Naive Bayes, and XGBoost, under consistent experimental settings.

•This study evaluates phishing detection performance using both conventional and risk-aware metrics, including accuracy, macro-F1, phishing recall, phishing false-negative rate (FNR), and Area Under the Receiver Operating Characteristic Curve (ROC-AUC).

•This study investigates compact feature subsets to examine whether fewer important structural features can maintain competitive phishing detection performance.

•This study applies threshold optimization to analyze the trade-off between reducing missed phishing websites and

controlling false alarms on legitimate websites.

•This study uses Random Forest feature importance and permutation importance to identify the most influential phishing-related structural indicators.

2. RELATED WORK

The problem of phishing websites has been a key field of interest for cybersecurity research, as it continues to affect vulnerable users who enter bogus websites, leading them to share their personal information. With phishing techniques becoming more sophisticated, research has moved away from signature-based phishing identification and toward sophisticated, automated algorithms capable of recognizing phishing websites based on features of the URL, phishing website structure, data associated with the domain, and learning-based models [17, 18]. There are three types of current research studies: classical blacklist and heuristic methods, feature-based machine learning procedures, and more recent deep learning techniques [19]. This is a step towards a more dynamic detection model that is based on data, as opposed to static detection systems, and is better equipped to adapt to an evolving phishing threat.

2.1 Traditional phishing detection methods

Early phishing detection systems depended mostly on blacklist and heuristic-based techniques. Blacklist-based techniques detect phishing websites by comparing URLs with databases of previously identified malicious domains or webpages that are managed by security services, browser vendors, and other associated security platforms [20, 21]. These computationally efficient techniques are still commonly utilized in real security products and current web browsers. Nevertheless, their success is dependent on past reporting; therefore, they are often unreliable against recently generated phishing websites that have not yet been recognized and added to blacklist databases. To address these issues, instead of depending on existing recognized malicious content, heuristic-based solutions were developed to resolve some of the limitations by searching for suspicious URL patterns and webpage architecture. The general markers of suspicious website activities are strange, uncommon domain names, fake hyperlinks, and anomalies in URL patterns [22]. The efficacy of blacklist techniques is inferior to that of heuristic methods in identifying previously undetected phishing pages. Nevertheless, they are primarily dependent on manually constructed algorithms and constructed signals that require periodic updates due to the fact that phishing strategies are swiftly evolving [23]. This scenario complicates maintenance and may reduce the effectiveness of detection.

2.2 Machine learning approaches for phishing detection

Machine learning is used to detect phishing, so as to resolve the limitations of the blacklist and heuristic approaches. Thus, machine-learning-based systems' classifiers are trained on labelled datasets comprising both phishing and genuine websites. They are able to discriminate phishing websites from legitimate websites by using URLs and domain characteristics [24]. Hence, machine learning is one of the crucial areas of study in the identification of phishing websites. Accordingly, an extensive variety of machine learning methods were examined. For the detection of phishing websites, Ramana et al. have introduced a collection of feature-selection approaches and reported higher accuracy in classification compared to baseline algorithmic methods [25]. Other research investigations have utilized extracted website features, like lexical, structural, and visual-layout characteristics. Additional methods include feature optimization methods that were employed to determine the optimal informative characteristics [26]. Moreover, comparative research indicates that ensemble learning enables significant performance benefits. A study was conducted to compare the results of a Voting Classifier with the results of the single classifiers, namely Decision Tree, Random Forest, Gradient Boosting, XGBoost, AdaBoost and Multilayer Perceptron. It has also simultaneously lowered the false positive and false negative results [27]. Random Forest in particular has been very popular due to its good prediction performance, robustness, and little overfitting [28]. The integration of feature selection methods with machine-learning classifiers is employed by hybrid methodologies to minimize feature redundancy while increasing classification accuracy [29]. The literature shows that despite the advancement of the artificial intelligence field, conventional and ensemble machine learning methods remain highly effective methods for the identification of phishing, especially when they are accompanied by the use of suitable feature engineering, feature selection, and controlled comparison of the models.

2.3 Feature-based phishing detection

Feature engineering is a vital aspect of phishing website identification, particularly in conventional machine-learning techniques. Numerous research studies divide phishing identification characteristics into three categories: lexical, host-driven, and content-driven, as retrieved from URLs and webpage architecture [30]. Lexical characteristics often define a URL's arrangement, such as its length, token structure, application of special characters, questionable prefixes or suffixes, and the use of an IP-address-based routing instead of a domain name in the URL. These characteristics are frequently relevant since phishing URLs often produce structural abnormalities designed to impersonate genuine domains while concealing harmful intentions. Host-driven characteristics offer extra data regarding the validity of websites. Domain registration characteristics, DNS-associated characteristics, SSL-related data, and other domain-level indicators that aid in the detection of questionable or temporary websites are examples of such qualities [31]. These markers have enhanced the identification of phishing attacks due to malicious operations. These indicators are useful for identifying newly registered or suspicious domains. Mohammad and McCluskey [32] created a phishing website dataset that is used widely in phishing detection studies. The assessment of machine-learning models for identifying phishing uses the University of California, Irvine (UCI) dataset that comprises 11,055 cases and 30 URL- and website-associated characteristics. Some studies have shown that structural and domain-related properties are significant indicators for phishing website detection. The continuous dependency on the sets of structured characteristics highlights the practical importance of defining the characteristics that most importantly affect the phishing website categorization.

2.4 Deep learning approaches

During the last few years, phishing detection researchers have been increasingly exploring deep learning algorithms to reduce the dependence on manual feature engineering and improve phishing prediction accuracy. Feature representations were automatically extracted from URLs and web content using deep neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks [33, 34]. These methods are aimed at identifying patterns directly from the raw or somewhat processed input instead of relying entirely on the features that are engineered. CNN methods are often used with character sequences to learn phishing-related patterns without the need for explicit manual feature extraction from URLs. Other deep learning studies have used neural networks to extract URL attributes and then match them to the HTML or web page content to identify phishing-related structures and patterns on web pages [35]. These methods have yielded promising results and shown the potential of representation learning in detecting phishing attacks. Notwithstanding these benefits, deep learning techniques frequently need larger training datasets, extended training durations, and larger computational capacity than conventional machine-learning techniques [36, 37]. Thus, the feature-based machine learning models remain relevant in practical applications where interpretability, computational efficiency, and ease of implementation are important. Although significant research advances in deep learning have been made, traditional machine-learning methods are still an object of research interest.

2.5 Positioning of the present work

There are some limitations that are still present in the available literature. A lot of recent work has been done on complex deep learning architectures and hybrid or single machine learning classifiers. While these methods have shown some encouraging results, there is not much attention paid to indicators that are specific for phishing, such as false negative rates, in the majority of studies. For phishing website detection, this is a crucial concern, since a fraudulent phishing site might be marked as a genuine site and expose the user to a serious cyber threat. Moreover, it is difficult to compare previous studies because of variations in datasets, feature representations, data preprocessing and evaluation protocols. Thus, comparative evaluations under controlled conditions with the same data and identical experimental conditions continue to be valuable in both research and practical application. Furthermore, explainability and feature reduction remain relevant as phishing detection systems need to be practical, not only in terms of their performance but also in terms of having understandable indicators and avoiding unnecessary feature complexity. The current study is conducted for the purpose of addressing the aforementioned issues and compares the performance of several supervised machine learning classifiers such as Random Forest, SVM, Logistic Regression, Decision Tree, K-Nearest Neighbors, Naive Bayes and XGBoost in detecting phishing websites with a publicly available benchmark dataset of structural website features. The risk-aware false negative analysis, stratified cross-validation, compact feature subset evaluation, threshold optimization and permutation-based explainability aren't only included because they're simple, but because they align with the specific risk-aware requirements of this study. The goal of the current study, therefore, is to offer a more practical and understandable analysis of the machine learning models for detecting phishing websites.

3. EXPERIMENTAL DESIGN AND IMPLEMENTATION

This section describes the data preparation and experimental protocol used for the risk-aware evaluation of phishing website detection models. It first introduces the dataset, label encoding, preprocessing steps, and data-partitioning procedure. The evaluated machine learning models and implementation environment are then presented, followed by the performance evaluation protocol. Finally, the procedures used for compact feature-subset selection, decision-threshold optimization, and explainability analysis are described.

3.1 Dataset and data preparation

The dataset of Phishing Websites, publicly available in the UCI Machine Learning Repository, was used. It consists of 11,055 website cases, each with 30 prediction variables and a class label of a real or fraudulent website, and is linked to the phishing website research done by Mohammad and McCluskey [32]. Each occurrence is a website representation and is determined using security and structure indicators derived from URL characteristics, domain-related features, and the HTML-based webpage data. They identify characteristics that are generally observed on phishing sites, such as unusual characteristics in the URL, unusual domain behavior, and unusual properties of hyperlinks. The dataset includes attributes such as having_IP_Address, URL_Length, Shortining_Service, having_Sub_Domain, SSLfinal_State, Domain_registeration_length, URL_of_Anchor, Links_in_tags, web_traffic, and Google_Index. The data are classified into two classes: those reputable websites are labeled as 1, while phishing websites are labeled as -1. For a binary classification, the phishing label was changed from -1 to 0, while the authentic websites retained the label 1. After label conversion, there were 6157 genuine and 4898 fake web pages. This dataset was originally saved in Attribute-Relation File Format (ARFF) format. It was imported into Python using the SciPy package and then converted to a pandas DataFrame for further processing. All the features were converted to integers for training the model, as they were given as attributes in the ARFF file as byte strings. After this change, the dataset was now completely numeric and ready to use for the selected machine learning algorithms without any further encoding. The feature matrix (represented by X) was created by removing the class label column (Result), while the target vector (represented by y) contained processed binary class labels. The dataset consisted of 30 predictive characteristics and a single binary target value. To preserve the independence of the final evaluation, the dataset was partitioned in two stages using the train_test_split function from sklearn.model_selection. In the first stage, 60% of the complete dataset was assigned to the training partition, while the remaining 40% formed a temporary partition (test_size = 0.40, shuffle = True, stratify = y, and random_state = 42). In the second stage, the temporary partition was divided equally into validation and final test partitions (test_size = 0.50, shuffle = True, stratify = y_temp, and random_state = 42). The original ARFF row order and zero-based pandas indices were

retained before partitioning. As a reproducibility check, the first five original indices were [3855, 1778, 7296, 2652, 3076] for training, [3015, 8054, 9339, 5851, 220] for validation, and [6589, 3062, 7220, 7302, 9189] for final testing. The resulting training partition contained 6,633 instances, comprising 2,939 phishing and 3,694 legitimate websites. The validation partition contained 2,211 instances, comprising 979 phishing and 1,232 legitimate websites, while the final test partition contained 2,211 instances, comprising 980 phishing and 1,231 legitimate websites. The training partition was used exclusively for model fitting and feature-importance estimation. The validation partition was used to compare classifiers, select the feature-subset size, and optimize the phishing decision thresholds. All model, feature, and threshold choices were frozen before the final test partition was accessed. The final test partition was subsequently used only for independent performance reporting and post-hoc explainability analysis, and no test instance contributed to model fitting or configuration selection.

3.2 Machine learning models and implementation

In this study, the accuracy of the following supervised machine learning classifiers for phishing website detection was evaluated: Random Forest, SVM, Logistic Regression, Decision Tree, K-Nearest Neighbors, Naive Bayes, and XGBoost. These classifiers were chosen to compare the various learning strategies, such as ensemble, kernel-based, linear, tree-based, instance-based, probabilistic, and boosting approaches. The main ensemble classifier used was Random Forest. It builds several decision trees during training and takes the average of their predicted class probabilities to get the final classification. This aggregation process creates more stability in the predictions than a single decision tree does, and also minimizes the chance of overfitting. For this study, the Random Forest classifier was run with 200 trees, which is a common number for the model, and a fixed random state of 42 for ease of repeatability. The SVM was also used as a classification model, which aims to find the optimal decision boundary between phishing websites and legitimate websites. StandardScaler was used prior to the training of the models to ensure that all of the features fed into the model contributed to the model on a comparable scale since SVM is sensitive to feature scale. Logistic Regression was used as a linear baseline classifier for binary classification and estimates class membership probabilities using the logistic function. StandardScaler was also used for Logistic Regression, and the maximum number of iterations was increased to 2000 to ensure convergence of the model. For a more comprehensive comparison evaluation, additional baseline classifiers were added. Simple Tree-based Classifier (Decision Tree), Instance-based Classifier (K-Nearest Neighbors), Probabilistic Classifier (Naive Bayes), and Boosting-based Ensemble Classifier (XGBoost) were used respectively as classifiers. In the case of K-Nearest Neighbors, the features were scaled using StandardScaler. All model configurations were fixed before evaluation. Random Forest used 200 trees, the Gini criterion, unrestricted tree depth, a minimum of two samples for splitting, a minimum of one sample per leaf, square-root feature sampling, and bootstrap sampling. The SVM used the radial basis function kernel, C = 1.0, gamma = "scale," and probability estimation. Logistic Regression used C = 1.0, the lbfgs solver, and a maximum of 2,000 iterations. Decision Tree used the Gini criterion, best-split strategy, unrestricted depth, a minimum of two samples for splitting, and a minimum of one sample per leaf. K-Nearest Neighbors used five neighbors, uniform weighting, automatic neighbor-search selection, and Euclidean distance. Gaussian Naive Bayes used var_smoothing = 10^{-9}. XGBoost used a binary logistic objective, 200 estimators, a learning rate of 0.05, maximum depth of 5, minimum child weight of 1, subsample ratio of 0.90, column-sampling ratio of 0.90, L1 regularization of 0, L2 regularization of 1, histogram-based tree construction, and log-loss evaluation. A random seed of 42 was applied to all stochastic estimators. The ARFF attributes were converted from byte-string representations to integers before partitioning; the dataset contained no missing values and therefore required no imputation. No oversampling, undersampling, or class weighting was applied. Standardization was required only for SVM, Logistic Regression, and K-Nearest Neighbors. For these classifiers, StandardScaler was embedded within a scikit-learn pipeline. Consequently, during each training-only cross-validation iteration, the scaler was fitted exclusively on the nine training folds and then applied to the held-out fold, preventing preprocessing leakage. Random Forest, Decision Tree, Gaussian Naive Bayes, and XGBoost were trained using the unscaled integer features. The reported 10-fold cross-validation results used all 30 features and did not use the validation or final test partitions. All experiments were executed using Python 3.12.13. Data manipulation was performed using pandas 2.2.3 and NumPy 2.5.2, while the ARFF dataset was loaded using SciPy 1.18.1. Model construction, pipelines, cross-validation, metric calculation, and permutation importance were implemented using scikit-learn 1.8.0. XGBoost 3.4.1 was used for the boosting classifier, and Matplotlib 3.10.8 was used to generate the figures. Parallel processing was enabled where supported by setting n_jobs = -1. The complete experimental workflow used a random seed of 42. The resulting metrics, predictions, confusion matrices, feature rankings, and threshold analyses were stored as Comma-Separated Values (CSV) files and figures.

3.3 Performance evaluation protocol

In order to evaluate the performance of the classifiers, several conventional and risk-aware classification metrics were used. The conventional metrics included accuracy, precision, recall, F1-score, macro-F1, weighted-F1, balanced accuracy, and ROC-AUC. These metrics provided an overall assessment of the classifiers' ability to distinguish between phishing and legitimate websites. The candidate classifiers were fitted exclusively to the 60% training partition and compared using the separate 20% validation partition under identical experimental conditions. Macro-F1 was specified as the primary model-selection metric because it assigns equal importance to the phishing and legitimate classes, while the phishing FNR was used as a secondary security-oriented criterion. In addition, stratified 10-fold cross-validation was conducted exclusively within the training partition to assess the stability of each classifier without involving the validation or final test data. For each classifier, the mean and standard deviation were calculated for accuracy, balanced accuracy, macro-F1, weighted-F1, phishing recall, and phishing F1-score across the ten folds. After the model-selection procedure was completed, the selected model was used for feature-subset and threshold analyses on the validation partition. The final

test partition remained inaccessible until all model and operating-point decisions had been fixed. Since phishing website detection is a security-sensitive task, the evaluation did not depend only on overall accuracy. Special attention was given to the phishing class because misclassifying a phishing website as legitimate may expose users to serious cybersecurity risks. Therefore, phishing-class precision, phishing-class recall, phishing-class F1-score, and phishing FNR were also calculated. In this study, the phishing false-negative case refers to a phishing website that is incorrectly classified as legitimate. A lower phishing FNR indicates that fewer phishing websites are missed by the classifier. Therefore, this metric was used to support the risk-aware evaluation of the tested models. In addition to the numerical metrics, confusion matrices and classification reports were generated to examine the distribution of correct and incorrect predictions for both phishing and legitimate websites. Statistical comparisons were performed using paired predictions from the validation partition. A two-sided exact McNemar test was applied to the discordant correctness outcomes for Random Forest versus XGBoost, Random Forest versus Decision Tree, and the complete 30-feature Random Forest versus its top-20-feature counterpart. In addition, paired nonparametric bootstrapping with 10,000 resamples and random seed 42 was used to calculate 95% percentile confidence intervals for the differences in accuracy and macro-F1. Statistical significance was assessed at $\alpha = 0.05$. These analyses were restricted to the validation partition, and the final test partition was not used for significance testing or model selection.

3.4 Feature selection, threshold optimization, and explainability

A compact feature-subset analysis was conducted without using the final test partition. First, impurity-based feature importance was calculated from a Random Forest model fitted exclusively to the training partition. The resulting ranking was then used to construct subsets containing the 5, 10, 15, and 20 highest-ranked features, in addition to the complete set of 30 features. A separate Random Forest model was fitted to the training partition for each feature setting and evaluated on the validation partition. Macro-F1 was used as the primary criterion for selecting the feature-subset size, while the phishing FNR was considered as a secondary security-oriented criterion. The selected feature configuration was frozen before the final test partition was accessed. This procedure ensured that the test data did not influence either the feature ranking or the selection of the number of features. Threshold optimization was performed exclusively on the validation partition after the classifier and feature configuration had been selected. Because phishing instances were encoded as class 0, the Random Forest output corresponding to class 0 was treated as the phishing probability. A website was classified as phishing when its estimated phishing probability was greater than or equal to the specified threshold; otherwise, it was classified as legitimate. Thresholds ranging from 0.10 to 0.90 in increments of 0.01 were evaluated on the validation partition. Two operating points were defined before accessing the final test data: the threshold that maximized validation macro-F1 and a risk-aware threshold that maximized phishing recall while maintaining the legitimate false-positive rate (FPR) at or below 5%. Lowering the phishing threshold generally classifies more websites as phishing, thereby reducing phishing false negatives while increasing false alarms among legitimate websites. After both thresholds had been selected, they were frozen and applied without modification to the untouched final test partition. Two complementary feature-importance methods were used. First, impurity-based importance was extracted from the selected Random Forest fitted exclusively on the training partition; neither the validation nor final test partition contributed to this ranking. Second, permutation importance was calculated as a post-hoc analysis on the untouched final test partition only after the model, feature configuration, and thresholds had been fixed. Macro-F1 was used as the permutation scoring metric, and each feature was independently shuffled 30 times using random seed 42. Both the mean decrease in macro-F1 and its standard deviation across the 30 repeats were reported. Agreement between the training-derived impurity ranking and the post-hoc permutation ranking was assessed across all 30 features using Spearman rank correlation and overlap between top-10 feature sets. The post-hoc permutation results were used only for interpretation and did not influence model or feature selection. This analysis examined whether a compact feature subset could maintain strong phishing detection performance with lower feature complexity.

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section discusses the experimental results obtained from phishing website detection by using a set of machine learning classifiers. The evaluation was carried out on the basis of conventional and risk-aware performance indicators. Besides the overall classification performance, particular focus was placed on the phishing-class recall and phishing FNR as it is a serious security risk to misclassify a phishing website as a legitimate one. This section includes model comparison, cross-validation analysis, compact subset of features evaluation, threshold optimization, and the analysis of explainability.

4.1 Risk-aware model performance comparison

The seven candidate classifiers were fitted using the 60% training partition and compared on the separate 20% validation partition under identical experimental conditions. Their performance was evaluated using accuracy, macro-F1, phishing recall, phishing FNR, and ROC-AUC. Macro-F1 served as the primary model-selection metric, while phishing recall and the phishing FNR provided security-oriented measures of missed phishing websites. Table 1 presents the validation performance of the evaluated classifiers. The final test partition was not used during this comparison or the subsequent configuration-selection procedures.

Random Forest produced the highest observed validation accuracy of 97.24% and macro-F1 of 97.20%, together with a phishing recall of 96.02% and a phishing FNR of 3.98%. XGBoost and Decision Tree achieved the next-highest macro-F1 values of 95.77% and 95.69%, respectively. Although Naive Bayes achieved the highest phishing recall of 99.80% and the lowest phishing FNR of 0.20%, its substantially lower accuracy of 59.43% and macro-F1 of 55.72% indicate that this sensitivity was obtained at the cost of frequent false alarms involving legitimate websites. Based on the prespecified primary criterion of validation macro-F1, Random Forest was selected for the subsequent feature-subset, threshold, and explainability analyses. These values represent validation-set

observations, while the selected configuration was evaluated separately on the untouched final test partition. Because XGBoost achieved the second-highest observed validation macro-F1, it was included in the paired statistical comparison. Random Forest correctly classified 51 instances that XGBoost misclassified, whereas XGBoost correctly classified 20 instances that Random Forest misclassified. The two-sided exact McNemar test was significant ($p = 0.000303$). Random Forest improved accuracy by 1.40 percentage points, with a paired-bootstrap 95% confidence interval from 0.68 to 2.17 percentage points, and improved macro-F1 by 1.43 percentage points, with a 95% confidence interval from 0.69 to 2.20 percentage points. Thus, the observed validation-set advantage of Random Forest over XGBoost was statistically supported. The paired validation comparison provided statistical support for the observed Random Forest advantage over Decision Tree. Random Forest correctly classified 48 instances that Decision Tree misclassified, whereas Decision Tree correctly classified 15 instances that Random Forest misclassified. The two-sided exact McNemar test was significant ($p = 0.000038$). Random Forest improved accuracy by 1.49 percentage points, with a paired-bootstrap 95% confidence interval of 0.81 to 2.22 percentage points, and improved macro-F1 by 1.51 percentage points, with a 95% confidence interval of 0.82 to 2.24 percentage points. Thus, the validation-set advantage of Random Forest over Decision Tree was statistically supported.

4.2 Cross-validation results

To assess classifier stability without using data reserved for configuration selection or final evaluation, stratified 10-fold cross-validation was conducted exclusively within the 60% training partition. In each iteration, nine folds were used for model fitting and the remaining fold was used for evaluation. Table 2 reports the mean and standard deviation of accuracy, macro-F1, phishing recall, and phishing F1-score across the ten folds.

Random Forest achieved the highest observed mean cross-validation accuracy of 96.64% and macro-F1 of 96.59%, with standard deviations of 0.71% and 0.73%, respectively. XGBoost and Decision Tree produced similar mean macro-F1 values of 95.61% and 95.60%. Naive Bayes achieved the highest mean phishing recall of 99.90%; however, its substantially lower mean macro-F1 of 56.99% indicates poor balance between the phishing and legitimate classes. The training-only cross-validation results were consistent with the validation-set ranking, in which Random Forest also achieved the highest observed macro-F1. These descriptive results supported carrying Random Forest forward to the subsequent validation-based configuration analyses, while the statistical uncertainty of the observed differences was assessed separately.

Table 1. Validation-set risk-aware performance comparison of classification models

Model	Accuracy	Macro-F1	Phishing Recall	Phishing FNR	ROC-AUC
Random Forest	97.24%	97.20%	96.02%	3.98%	99.60%
XGBoost	95.84%	95.77%	94.18%	5.82%	99.42%
Decision Tree	95.75%	95.69%	94.79%	5.21%	97.07%
SVM	95.21%	95.12%	92.75%	7.25%	98.74%
K-Nearest Neighbors	93.94%	93.84%	91.73%	8.27%	98.39%
Logistic Regression	92.67%	92.55%	90.40%	9.60%	97.76%
Naive Bayes	59.43%	55.72%	99.80%	0.20%	96.63%

Note: False-Negative Rate (FNR); Receiver Operating Characteristic Curve (ROC); Area Under the Curve (AUC); Extreme Gradient Boosting (XGBoost).

Table 2. Stratified 10-fold cross-validation results on the training partition

Model	Accuracy	Macro-F1	Phishing Recall	Phishing F1
Random Forest	96.64% ± 0.71%	96.59% ± 0.73%	95.20% ± 1.10%	96.17% ± 0.82%
XGBoost	95.67% ± 0.38%	95.61% ± 0.39%	94.32% ± 0.86%	95.08% ± 0.42%
Decision Tree	95.66% ± 0.78%	95.60% ± 0.79%	94.83% ± 1.41%	95.08% ± 0.90%
SVM	94.80% ± 0.44%	94.71% ± 0.45%	92.65% ± 1.28%	94.04% ± 0.50%
K-Nearest Neighbors	93.13% ± 0.84%	93.02% ± 0.86%	91.53% ± 1.51%	92.18% ± 0.98%
Logistic Regression	92.61% ± 0.80%	92.49% ± 0.81%	90.41% ± 1.97%	91.55% ± 0.94%
Naive Bayes	60.41% ± 1.32%	56.99% ± 1.80%	99.90% ± 0.23%	69.11% ± 0.69%

Note: Extreme Gradient Boosting (XGBoost).

4.3 Compact feature subset analysis

To determine whether feature reduction could preserve detection performance without consulting the final test data, feature importance was first estimated using a Random Forest fitted exclusively to the training partition. The resulting ranking was used to construct subsets containing the top 5, 10, 15, and 20 features, together with the complete 30-feature configuration. A separate Random Forest was trained for each configuration using the training partition and evaluated on the validation partition. Table 3 presents the resulting validation performance.

The validation results show a consistent improvement as additional features were included. The complete 30-feature configuration achieved the highest observed accuracy of 97.24%, macro-F1 of 97.20%, and phishing recall of 96.02%, together with the lowest phishing FNR of 3.98%. The top-20 subset produced similar but slightly lower values, with an accuracy of 97.06%, macro-F1 of 97.02%, phishing recall of 95.71%, and phishing FNR of 4.29%. Therefore, the validation analysis did not provide evidence that reducing the feature set improved predictive performance. The complete 30-feature configuration was consequently selected and frozen for threshold optimization and final test evaluation. The paired comparison did not establish a statistically significant difference between the complete 30-feature model and the top-20-feature model. The complete model correctly classified eight validation instances that the top-20 model misclassified, whereas the top-20 model correctly classified four instances that the complete model misclassified. The two-sided exact

McNemar test was not significant (p = 0.3877). The accuracy difference was 0.18 percentage points, with a paired-bootstrap 95% confidence interval from -0.14 to 0.50 percentage points, while the macro-F1 difference was 0.18 percentage points, with a 95% confidence interval from -0.13 to 0.50 percentage points. The complete 30-feature configuration was retained according to the prespecified validation-selection rule, but its small observed advantage over the top-20 subset should not be interpreted as statistically established.

As shown in Figure 1, validation accuracy, macro-F1, and phishing recall generally increased as additional features were included. Although the gains between the top-20 and complete 30-feature configurations were small, the complete feature set achieved the highest observed value for all three metrics and the lowest phishing FNR reported in Table 3. Therefore, all 30 features were retained for the subsequent threshold analysis and independent final test evaluation.

Table 3. Validation-set feature-subset analysis using Random Forest

Feature Subset	Accuracy	Macro-F1	Phishing Recall	Phishing FNR
Top 5 features	92.72%	92.58%	89.38%	10.62%
Top 10 features	94.48%	94.40%	93.05%	6.95%
Top 15 features	96.65%	96.60%	95.20%	4.80%
Top 20 features	97.06%	97.02%	95.71%	4.29%
All 30 features	97.24%	97.20%	96.02%	3.98%

Note: False-Negative Rate (FNR).

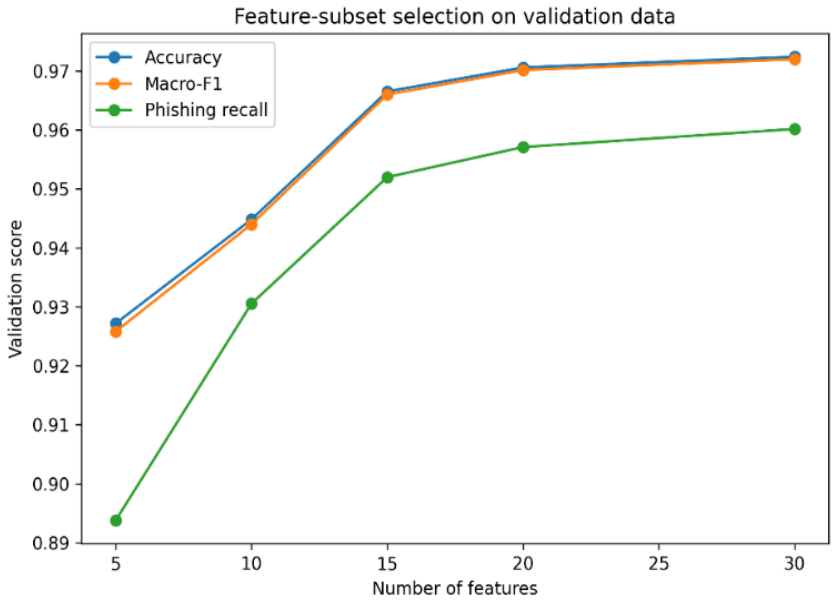


Figure 1. Validation performance across Random Forest feature-subset sizes

4.4 Risk-aware threshold optimization

Following validation-based model and feature selection, threshold analysis was conducted using the complete 30-feature Random Forest configuration. The probability of phishing was calculated for the validation partition and thresholds of 0.10 to 0.90 in increments of 0.01 were tested. Two operating points were chosen for comparison with the 0.50 default threshold: a threshold that maximized the macro-F1 and a risk-aware threshold that maximized the phishing recall while keeping the legitimate FPR at or below 5%. These three operating points have been validated and are shown in Table 4. No thresholds were determined or adjusted using final test data.

The validation accuracy and macro-F1 were 97.24% and 97.20%, respectively, while the phishing recall, phishing FNR, and legitimate FPR were 96.02%, 3.98%, and 1.79%, respectively, at the default threshold of 0.50. The highest validation macro-F1 (97.33%) and validation accuracy (97.38%) were obtained with a threshold of 0.52 while phishing recall slightly dropped to 95.81%. The risk-aware threshold 0.29, on the other hand, boosted phishing recall to 98.26% and cut the phishing FNR to 1.74%, while raising the legitimate FPR to 4.95%. These results demonstrate the anticipated operating characteristic: a lower phishing threshold decreases the number of phishing sites that are not detected, and raises the number of false alarms involving legitimate sites.

As shown in Figure 2, increasing the phishing probability threshold reduced the number of websites classified as phishing. Consequently, phishing recall generally decreased, and the phishing FNR increased, while the legitimate FPR declined. The threshold of 0.52 was located near the maximum validation macro-F1, whereas the lower risk-aware threshold of 0.29 increased phishing sensitivity while remaining within the predefined 5% legitimate false-positive-rate constraint. Both operating points were selected exclusively from validation data. After model selection, feature-set selection, and threshold optimization were completed on the training and validation partitions, the resulting Random Forest configuration was evaluated on the untouched 20% final test partition. The model used all 30 structural features. The default threshold of 0.50 and the two validation-selected thresholds of 0.52 and 0.29 were applied without further adjustment. Table

5 reports the independent final test performance of these frozen operating points.

On the untouched test partition, the default threshold of 0.50 achieved the highest accuracy of 97.01% and macro-F1 of 96.97%. The validation-selected macro-F1 threshold of 0.52 produced similar but slightly lower final-test values; it was not readjusted after observing the test results. The risk-aware threshold of 0.29 increased phishing recall from 95.20% to 97.65% and reduced the phishing FNR from 4.80% to 2.35%. This improvement in phishing sensitivity was accompanied by an increase in the legitimate FPR from 1.54% to 4.79% and a reduction in accuracy to 96.29%. These independent results confirm the expected trade-off between reducing missed phishing websites and controlling false alarms without using the test partition for configuration selection.

Table 4. Validation-set threshold selection results for Random Forest

Threshold Setting	Phishing-Probability Threshold, τ [P(y = 0 | x)]	Accuracy	Macro-F1	Phishing Recall	Phishing FNR	Legitimate FPR
Default threshold	0.50	97.24%	97.20%	96.02%	3.98%	1.79%
Best validation macro-F1 threshold	0.52	97.38%	97.33%	95.81%	4.19%	1.38%
Risk-aware validation threshold	0.29	96.47%	96.44%	98.26%	1.74%	4.95%

Note: False-Negative Rate (FNR); False-Positive Rate (FPR).

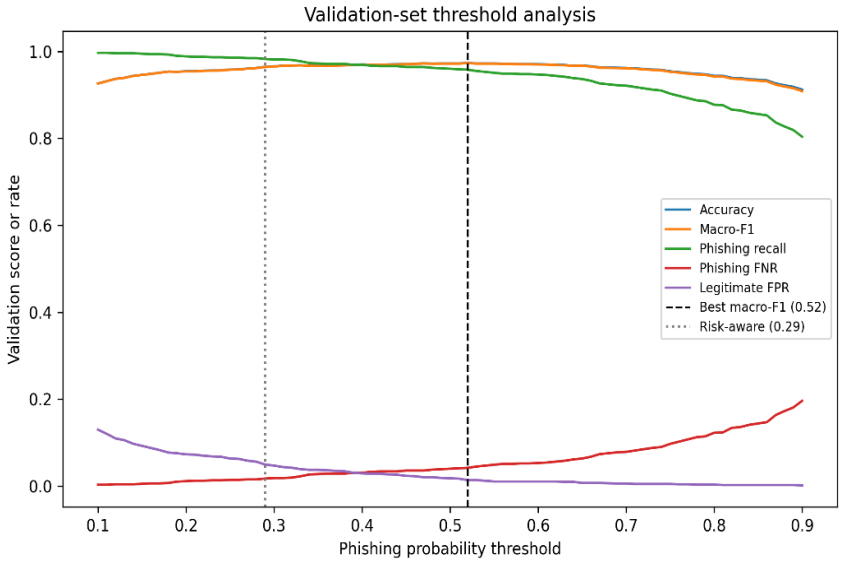


Figure 2. Validation-set trade-off across thresholds applied to the phishing probability P(y = 0 | x), where class 0 represents phishing websites

Table 5. Independent final-test performance of the frozen Random Forest operating points

Operating Point	Threshold	Accuracy	Macro-F1	Phishing Recall	Phishing FNR	Legitimate FPR
Default threshold	0.50	97.01%	96.97%	95.20%	4.80%	1.54%
Validation-selected macro-F1 threshold	0.52	96.97%	96.92%	95.10%	4.90%	1.54%
Validation-selected risk-aware threshold	0.29	96.29%	96.26%	97.65%	2.35%	4.79%

Note: False-Negative Rate (FNR); False-Positive Rate (FPR).

4.5 Explainability analysis

Explainability analysis was conducted to identify the most influential structural website features used by the Random Forest model. First, Random Forest impurity-based feature importance was calculated. Then, permutation importance was used to validate whether the same features remained important when their effect on macro-F1 was measured. Table 6 shows the training impurity and post-hoc permutation importance.

As shown in Table 6, the post-hoc final-test permutation analysis ranked URL_of_Anchor first and SSLfinal_State second, whereas the training-derived impurity analysis ranked SSLfinal_State first and URL_of_Anchor second. Across all 30 features, the two rankings showed strong agreement (Spearman $\rho = 0.951$, $p < 0.001$), and eight features were shared between their respective top-10 lists. This indicates that the two methods produced broadly consistent, although not identical, explanations. The feature web_traffic ranked third under both methods, while Prefix_Suffix, Links_in_tags, having_Sub_Domain, and Links_pointing_to_page were also prominent. Figure 3 presents the top 10 features according to the post-hoc permutation analysis.

As shown in Figure 3, URL_of_Anchor and SSLfinal_State produced the largest mean decreases in macro-F1 and were clearly more influential than the remaining features under the post-hoc permutation analysis. Their first- and second-place ordering should not be interpreted as a statistically established difference. Their prominence under both impurity and permutation analyses indicates stable model reliance within this dataset; however, these importance values are model and dataset-specific associations and should not be interpreted as causal effects.

Table 6. Training impurity importance and post-hoc final-test permutation importance

Permutation Rank	Feature	Training Impurity Importance (Rank)	Final-Test Permutation Importance, Mean ± SD
1	URL_of_Anchor	0.2559 (2)	0.1156 ± 0.0061
2	SSLfinal_State	0.3110 (1)	0.1087 ± 0.0045
3	web_traffic	0.0772 (3)	0.0288 ± 0.0028
4	Prefix_Suffix	0.0396 (6)	0.0247 ± 0.0029
5	Links_in_tags	0.0402 (5)	0.0237 ± 0.0022
6	having_Sub_Domain	0.0674 (4)	0.0183 ± 0.0026
7	Links_pointing_to_page	0.0203 (7)	0.0104 ± 0.0018
8	age_of_domain	0.0161 (11)	0.0088 ± 0.0015
9	SFH	0.0185 (9)	0.0075 ± 0.0016
10	Page_Rank	0.0112 (15)	0.0069 ± 0.0015

Note: Standard Deviation (SD); Server Form Handler (SFH); Secure Sockets Layer (SSL); Uniform Resource Locator (URL).

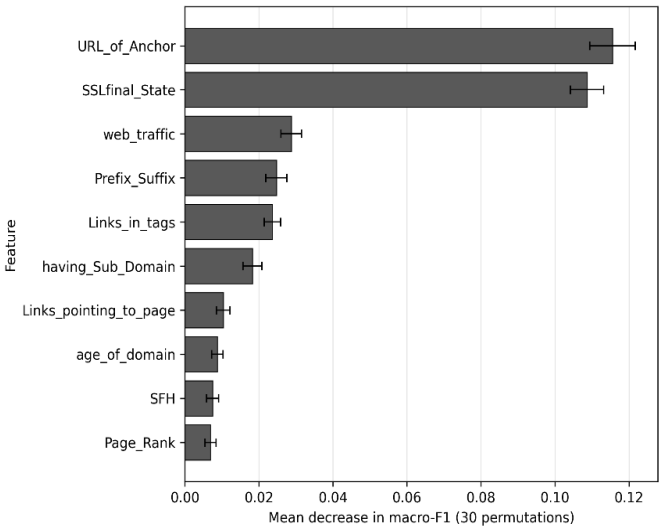


Figure 3. Post-hoc final-test permutation importance for the top 10 features

Note: Uniform Resource Locator (URL); Secure Sockets Layer (SSL); Server Form Handler (SFH).

4.6 Discussion

Random Forest has the highest validation accuracy and macro-F1 score, together with high phishing recall among the classifiers observed. Its advantages over XGBoost and Decision Tree were supported by paired exact McNemar tests (p = 0.000303 and p = 0.000038 for Random Forest versus XGBoost and Random Forest versus Decision Tree, respectively) and paired-bootstrap confidence intervals which did not include zero for the accuracy and macro-F1 differences. The remaining classifiers were not tested by paired significance tests, which is why the comparisons are descriptive. The results also demonstrate that the evaluation of phishing detection based on accuracy alone is not sufficient. Although Naive Bayes had a very high phishing recall, it had a low accuracy and a low macro-F1 score, meaning that it misclassified many of the legitimate websites as phishing. This underlines the need to consider balanced and risk-aware metrics, such as macro-F1, phishing recall, phishing FNR and legitimate FPR. The validation-based feature-subset analysis revealed that adding more features led to an increasingly close match of the reduced subsets to the performance of the full feature set. The full 30-feature configuration, however, performed the best in terms of the observed validation accuracy, macro-F1, and phishing recall, and the lowest phishing FNR. The top-20 subset yielded similar results but failed to enhance any of the reported validation metrics. Furthermore, the paired comparison was not significant (McNemar p = 0.3877), and the intervals of the accuracy and macro-F1 differences contained zero, meaning that there was no statistically significant performance difference between the two feature configurations. Hence, all 30 features are kept for the 'Frozen Final' configuration. The reduced-subset results show a trade-off between dimensionality and prediction but do not prove that there is a performance benefit to feature reduction. A security-oriented operating point selection method, which avoids looking at the final test data, was obtained by threshold analysis based on the validation. The threshold of 0.52 was the one that yielded the highest validation macro-F1, while the default threshold of 0.50 yielded a slightly higher macro-F1 on the independent test partition. No threshold was readjusted after the test results were observed. The risk-aware threshold of 0.29 lowered the final-test phishing false negative rate from 4.80% to 2.35% and raised the phishing recall rate from 95.20% to 97.65% while raising the legitimate FPR from 1.54% to 4.79%. The results show the expected trade-off between phishing sensitivity and false alarms, while the evaluation protocol maintained the independence of the final evaluation. Lastly, the training-derived impurity ranking and the post-hoc final-test permutation ranking were highly correlated for all 30 features (Spearman $\rho = 0.951$, $p < 0.001$) and had 8 features in common in the top 10. Under both methods, the top two features were URL_of_Anchor and SSLfinal_State, but the ranking of these two features was different. The results show that the model built using the Random Forest algorithm used the anchor-link and SSL-related metrics heavily in the analyzed benchmark and do not imply any cause and effect, nor that they were ordered the same way on more recent external data. One of the drawbacks is the size of the data set and its age. The UCI Phishing Websites dataset was published in 2015 and includes 11,055 examples, with each example having 30 structural features [32]. There were no collection times indicated on the records released, so temporal validation could not be carried out. In addition, the present study did not test an independent external data set with the same feature schema. Therefore, the findings reported here are internal generalizations and should not be assumed to reflect performance in current phishing attacks. A more recent dataset, called PhiUSIIL, comprises 235,795 legitimate and phishing URLs using 54 URL- and webpage-based features. It was not used as an external test of the fitted model as it has a different feature schema from the 30 engineered features used in this study, so direct evaluation of the fitted model would require extracting the same 30 features from its underlying websites. Future work should extend the evaluation protocol to newer independent benchmarks, and to continuously

collected (time-stamped) URLs for temporal robustness, as well as to cross-dataset generalizability, concept drift and threshold recalibration.

5. CONCLUSIONS

This study presented a risk-aware and explainable machine learning evaluation for phishing website detection using structural website features. Several supervised machine learning classifiers were evaluated, including Random Forest, SVM, Logistic Regression, Decision Tree, K-Nearest Neighbors, Naive Bayes, and XGBoost. Random Forest achieved the highest observed validation accuracy (97.24%) and macro-F1 score (97.20%), while maintaining high phishing recall (96.02%), together with a phishing FNR (3.98%). It was therefore selected for subsequent evaluation on the untouched final test partition. The results show that it is not enough to measure phishing detection models based on the rate of correct classifications because the most serious mistake is misclassifying a phishing site as a legitimate site. In this regard, the current study focused on phishing recall and phishing FNR as significant phishing risk-aware metrics. The validation-based feature analysis selected the complete 30-feature configuration because it achieved the highest observed validation accuracy, macro-F1, and phishing recall and the lowest phishing FNR. Although the top-20 subset produced similar results, it did not outperform the complete feature set. However, the difference between these configurations was not statistically significant, so retaining all 30 features reflects the prespecified validation-selection rule rather than evidence of statistical superiority. On the untouched final test partition, the default threshold achieved an accuracy of 97.01% and a macro-F1 score of 96.97%. The validation-selected risk-aware threshold of 0.29 increased phishing recall to 97.65% and reduced the phishing FNR to 2.35%, while increasing the legitimate FPR to 4.79%. These results demonstrate that threshold adjustment can reduce missed phishing websites but only through a measurable trade-off with false alarms. The training-derived impurity and post-hoc final-test permutation rankings showed strong agreement (Spearman $\rho = 0.951$), consistently placing URL_of_Anchor and SSLfinal_State as the two leading features, although their order differed. Web traffic, subdomain structure, prefix-suffix behaviour, and hyperlink-related features also appeared prominently. These findings describe the Random Forest model's reliance on structural indicators within the evaluated benchmark and should not be interpreted as causal or externally validated relationships. This study is limited by its reliance on a single benchmark dataset released in 2015 and by the absence of temporal and external validation [32]. Accordingly, the reported performance establishes generalization within the evaluated benchmark but does not establish effectiveness against contemporary phishing campaigns. Future work should prioritize evaluation on newer time-stamped and independently collected datasets, analysis of concept drift, threshold recalibration, and deployment-oriented assessment of predictive performance and computational efficiency.

REFERENCES

[1] Kamble, N., Mishra, N. (2024). Hybrid optimization enabled squeeze net for phishing attack detection. Computers & Security, 144: 103901. https://doi.org/10.1016/j.cose.2024.103901

[2] Sun, Z.B., Kokulu, F.B., Zhang, P., et al. (2024). From victims to defenders: An exploration of the phishing attack reporting ecosystem. In Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses, Padua, Italy, pp. 49-64. https://doi.org/10.1145/3678890.3678926

[3] Singh, T., Kumar, M., Kumar, S. (2024). Walkthrough phishing detection techniques. Computers and Electrical Engineering, 118: 109374. https://doi.org/10.1016/j.compeleceng.2024.109374

[4] Vennela, A., Akarapu, R.B., Rakshith, B.L., Asirvatham, L.G., Sunil, G. (2026). Intelligent cybersecurity systems for phishing attack detection - An overview. Computers and Electrical Engineering, 130: 110829. https://doi.org/10.1016/j.compeleceng.2025.110829

[5] Verizon. (2026). 2026 data breach investigations report. https://www.verizon.com/business/resources/reports/dbir/.

[6] Lella, I., Tsekmezoglou, E., Theocharidou, M., et al. (2023). ENISA threat landscape 2023. European Union Agency for Cybersecurity. https://doi.org/10.2824/782573

[7] Aljofey, A., Bello, S.A., Lu, J., Xu, C. (2025). Comprehensive phishing detection: A multi-channel approach with variants TCN fusion leveraging URL and HTML features. Journal of Network and Computer Applications, 238: 104170. https://doi.org/10.1016/j.jnca.2025.104170

[8] Doshi, T., Patel, V., Shah, N., Swain, D., Acharya, B. (2025). PhishHunter-XLD: An ensemble approach integrating machine learning and deep learning for phishing URL classification. Franklin Open, 12: 100349. https://doi.org/10.1016/j.fraope.2025.100349

[9] Barraclough, P.A., Fehringer, G., Woodward, J. (2021). Intelligent cyber-phishing detection for online. Computers & Security, 104: 102123. https://doi.org/10.1016/j.cose.2020.102123

[10] Zhu, E., Cheng, K., Zhang, Z.Z., Wang, H.B. (2024). PDHF: Effective phishing detection model combining optimal artificial and automatic deep features. Computers & Security, 136: 103561. https://doi.org/10.1016/j.cose.2023.103561

[11] Kavya, S., Sumathi, D. (2025). Staying ahead of phishers: A review of recent advances and emerging methodologies in phishing detection. Artificial Intelligence Review, 58: 50. https://doi.org/10.1007/s10462-024-11055-z

[12] Haraldsdóttir, M., Homayoun, S., Lynge, E., Jensen, C.D. (2024). Unmasking phishers: ML for malicious certificate detection. Computers & Industrial Engineering, 198: 110652. https://doi.org/10.1016/j.cie.2024.110652

[13] Rendall, K., Nisioti, A., Mylonas, A. (2020). Towards a multi-layered phishing detection. Sensors, 20(16): 4540. https://doi.org/10.3390/s20164540

[14] Kehkashan, T., Abdelhaq, M., Al-Shamayleh, A.S., et al. (2025). Explainable phishing website detection for secure and sustainable cyber infrastructure. Scientific Reports, 15: 41751. https://doi.org/10.1038/s41598-025-27984-w

[15] Opara, C., Chen, Y.K., Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw

URL and HTML characteristics. Expert Systems with Applications, 236: 121183. https://doi.org/10.1016/j.eswa.2023.121183

[16] Jain, A., Khan, S., Koli, K., et al. (2026). PhishNet 1.0: Optuna-optimized stacking ensemble with Boruta-based feature selection for phishing URL detection. Scientific Reports, 16: 1480. https://doi.org/10.1038/s41598-025-31447-7

[17] Bahaghighat, M., Ghasemi, M., Ozen, F. (2023). A high-accuracy phishing website detection method based on machine learning. Journal of Information Security and Applications, 77: 103553. https://doi.org/10.1016/j.jisa.2023.103553

[18] Wilk-Jakubowski, J.L., Pawlik, L., Wilk-Jakubowski, G., Sikora, A. (2025). Machine learning and neural networks for phishing detection: A systematic review (2017–2024). Electronics, 14(18): 3744. https://doi.org/10.3390/electronics14183744

[19] Kytidou, E., Tsikriki, T., Drosatos, G., Rantos, K. (2025). Machine learning techniques for phishing detection: A review of methods, challenges, and future directions. Intelligent Decision Technologies, 19(6): 4356-4379. https://doi.org/10.1177/18724981251366763

[20] Wang, B. (2025). Malicious URL detection with explainable machine learning techniques. In Proceedings of 2025 2nd International Conference on Informatics Education and Computer Technology Applications, Kuala Lumpur, Malaysia, pp. 293-299. https://doi.org/10.1145/3732801.3732854

[21] Safi, A., Singh, S. (2023). A systematic literature review on phishing website detection techniques. Journal of King Saud University - Computer and Information Sciences, 35: 590-611. https://doi.org/10.1016/j.jksuci.2023.01.004

[22] Shahbazi, Z., Jalali, R., Molaeevand, M. (2025). AI-based phishing detection and student cybersecurity awareness in the digital age. Big Data and Cognitive Computing, 9(8): 210. https://doi.org/10.3390/bdcc9080210

[23] Jabbar, H., Al-Janabi, S. (2025). AI-driven phishing detection: Enhancing cybersecurity with reinforcement learning. Journal of Cybersecurity and Privacy, 5(2): 26. https://doi.org/10.3390/jcp5020026

[24] Almomani, A., Alauthman, M., Shatnawi, M.T., et al. (2022). Phishing website detection with semantic features based on machine learning classifiers: A comparative study. International Journal on Semantic Web and Information Systems, 18(1): 1-24. https://doi.org/10.4018/IJSWIS.297032

[25] Ramana, A.V., Rao, K.L., Rao, R.S. (2021). Stop-Phish: An intelligent phishing detection method using feature selection ensemble. Social Network Analysis and Mining, 11: 110. https://doi.org/10.1007/s13278-021-00829-w

[26] Dandotiya, M., Goyal, N.K., Khunteta, A., Tiwari, B. (2026). Real time identification of phishing attacks through machine learning enhanced browser extensions. Scientific Reports, 16: 6612. https://doi.org/10.1038/s41598-026-35655-7

[27] Innab, N., Osman, A.A.F., Ataelfadiel, M.A.M., et al. (2024). Phishing attacks detection using ensemble machine learning algorithms. Computers, Materials & Continua, 80(1): 1325-1345. https://doi.org/10.32604/CMC.2024.051778

[28] Alhuzali, A., Alloqmani, A., Aljabri, M., Alharbi, F. (2025). In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets. Applied Sciences, 15(6): 3396. https://doi.org/10.3390/app15063396

[29] Elsoud, E.A., Bahroush, M., Al Henawi, E., et al. (2026). A novel framework for phishing detection based on backward recursive feature selection. Egyptian Informatics Journal, 33: 100922. https://doi.org/10.1016/j.eij.2026.100922

[30] Hamadouche, S., Boudraa, O., Gasmi, M. (2024). Combining lexical, host, and content-based features for phishing websites detection using machine learning models. EAI Endorsed Transactions on Scalable Information Systems, 11(6): 1-15. https://doi.org/10.4108/eetsis.4421

[31] Seweryn, K., Białczak, P., Chytry-Trzeciak, T. (2026). BadDomains: Early detection of phishing domains registration. Sensors, 26(3): 1041. https://doi.org/10.3390/s26031041

[32] Mohammad, R., McCluskey, L. (2012). Phishing websites. UCI Machine Learning Repository. https://doi.org/10.24432/C51W2X

[33] Hussain, M., Cheng, C., Xu, R., Afzal, M. (2023). CNN-fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet. Information Sciences, 631: 328-345. https://doi.org/10.1016/j.ins.2023.02.039

[34] Said, Y., Alsheikhy, A.A., Lahza, H., Shawly, T. (2024). Detecting phishing websites through improving convolutional neural networks with self-attention mechanism. Ain Shams Engineering Journal, 15(4): 102643. https://doi.org/10.1016/j.asej.2024.102643

[35] Zheng, F., Yan, Q., Leung, V.C.M., Yu, F.R., Ming, Z. (2022). HDP-CNN: Highway deep pyramid convolution neural network combining word-level and character-level representations for phishing website detection. Computers & Security, 114: 102584. https://doi.org/10.1016/j.cose.2021.102584

[36] Kyaw, P.H., Gutierrez, J., Ghobakhlou, A. (2024). A systematic review of deep learning techniques for phishing email detection. Electronics, 13(19): 3823. https://doi.org/10.3390/electronics13193823

[37] Prasad, A., Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 136: 103545. https://doi.org/10.1016/j.cose.2023.103545