https://doi.org/10.65542/djei.v2i2.40
Dasinya Journal for Engineering and Informatics. **2026**, *2*, 15.

Article

An Improved Machine Learning-Based Model for Phishing Website and URL Detection

Warveen M. Eido [1*], Omar S. Kareem [2]

[1] Department of Information Technology, Technical College of Informatics, Akre University for Applied Sciences, Duhok, Kurdistan Region, Iraq; warveen.aydo@dpu.edu.krd
[2] Department of Public Health, College of Health and Medical Technology, Duhok Polytechnic University, Duhok, Kurdistan Region, Iraq; omar.kareem@dpu.edu.krd
* Correspondence: warveen.aydo@dpu.edu.krd

Abstract

Cybersecurity experts consider that malicious URLs and phishing websites currently present their most dangerous threats because hackers use these threats to exploit both technical system weaknesses and user trust to steal sensitive data. The current detection methods, which use blacklist-based and rule-based systems, show decreasing effectiveness against new and unknown phishing attacks, which creates a demand for detection systems that can adapt to changing threats. The researchers developed an improved machine learning system that detects multiple types of phishing websites and URLs by using the ISCX-URL-2016 benchmark dataset. The framework uses data preprocessing methods, statistical feature engineering methods, and ANOVA F-test–based feature selection methods to enhance discriminative power while reducing feature redundancy. XGBoost serves as the primary classification model because it can handle the processing of high-dimensional structured URL features and the detection of complex nonlinear relationships. The system uses stratified cross-validation and randomized search as its hyperparameter tuning methods to achieve fairness in learning performance across different traffic types. The improved XGBoost model achieves high precision, recall, and F1-scores across all classes, which include benign, phishing, malware, defacement, and spam, while achieving an overall classification accuracy of 98.42%. The system reliably identifies phishing URLs with an F1 score of 0.96. The analysis of confusion matrix results shows that the system can separate different classes effectively because it produces very few misclassifications. The suggested architecture offers competitive performance with reduced computing complexity when compared to deep learning-based methods.

Keywords: XGBoost; ANOVA F-test; Machine Learning; Cybersecurity; Phishing Web site; URL Detection.

1. Introduction

In today's digital environment, phishing attacks have become one of the most common and destructive cyberthreats, putting people, businesses, and governmental organizations at grave risk. To steal sensitive data, such as login credentials, financial information, and personal identifiers, these assaults mostly use malicious URLs and dishonest websites that pose as trustworthy online businesses [1]. Phishing is a constant and changing cybersecurity threat due to the rapid growth of online services, cloud platforms, and digital transformation programs, which have

Received: January 26, 2026; Revised: March 26, 2026; Accepted: March 26, 2026; Published: April 12, 2026; Available online: April 12, 2026.

greatly increased the volume and sophistication of phishing campaigns [2]. Blacklist mechanisms and manually created rule-based systems were the main foundations of early phishing detection techniques. Although these methods worked well at first against known malicious URLs, they have serious flaws, most notably the inability to identify newly created domains and zero-day phishing attempts[3]. Traditional methods are insufficient for real-time and extensive protection because attackers constantly alter URL structures, domain registrations, and webpage content to get around static detection rules [4].

Machine learning (ML) and deep learning (DL) approaches have drawn a lot of attention as automatic and adaptive phishing detection solutions to address these issues [5]. While DL models can capture intricate and non-linear correlations within URL and webpage information, ML-based systems may learn discriminative patterns from large-scale datasets [6]. In order to increase detection accuracy and resilience against changing phishing tactics, an increasing amount of research has concentrated on supervised learning and deep neural architectures [7]. Deep learning models demonstrate potential, but their operational use in real-time and resource-limited environments gets restricted by their expensive processing needs, complex model designs, and their intense training resource demands [8], [9]. The Random Forest and Gradient Boosting machine learning models which use optimized ensemble-based techniques achieve detection performance that competes with other models while their system requirements become lighter and their results turn out to be easier to understand and their processing speed becomes faster. [10], [11] The practical advantages that ensemble machine learning techniques provide make them highly suitable for implementation in actual cybersecurity defense systems.

Building upon the growing interest in machine learning and deep learning approaches for phishing detection [12]. Recent research has highlighted the importance of intelligent classification models that can effectively analyze URL characteristics and identify malicious patterns in large-scale datasets [13], [14]. Machine learning-based phishing detection systems typically rely on extracting lexical, structural, or host-related features from URLs and training classification algorithms to distinguish phishing websites from legitimate ones [15]. Despite their promising performance, these methods still face several practical challenges in real-world cybersecurity environments. These challenges include the presence of high-dimensional feature spaces, redundant or irrelevant attributes, evolving phishing strategies that cause concept drift, and the need for computationally efficient models that can operate in real-time detection systems [16]. In addition, achieving a balance between detection accuracy, model interpretability, and processing efficiency remains a critical requirement for practical deployment in cybersecurity defense platforms.

To address these challenges, this study proposes an enhanced machine learning based on the XGBoost algorithm for phishing URL detection.[17] The proposed approach focuses on improving detection performance through a structured preprocessing pipeline and statistical feature selection techniques designed to reduce irrelevant features and improve the discriminative capability of the dataset. Specifically, the ANOVA F-test is employed to select the most informative URL features before training the classification model. The improved XGBoost model is then evaluated using the ISCX-URL-2016 benchmark dataset, which includes labeled samples representing benign traffic as well as phishing, malware, defacement, and spam activities. By combining efficient preprocessing, feature selection, and gradient boosting–based machine learning. The paper contains the following key contributions:

1. Proposed model improved phishing website and URL detection based on the XGBoost machine learning algorithm.
2. Develops and evaluates an improved XGBoost-based framework for multi-class phishing URL detection on the ISCX-URL-2016 dataset by combining structured preprocessing, ANOVA-based feature selection, and controlled parameter configuration to improve feature quality, discriminative power, predictive performance, and computational efficiency.
3. Demonstrates that the improved model achieves high detection performance while maintaining computational efficiency suitable for real-time cybersecurity applications.

The paper is organized as follows: Section 1.1 literature review; Section 2 introduces the datasets and preprocessing pipeline, feature selection, and classification model; Section 3 outlines the experimental setup and evaluation metrics and discusses the results and comparison with prior studies; and Section4 concludes with key findings and future directions.

2. Related work

[18] Barik et al. utilized a dataset of self-made phishing websites that was gathered from various real-world online sources and categorized as either benign or phishing. The study suggested EGSO-CNN, an optimized deep learning model. Using embedded CNN layers, automatic feature extraction was performed without the need for manual feature engineering. The improved model had an F1-score of 99.32% and an accuracy of 99.44%. The results showed that the best deep learning techniques are better for phishing detection.

[19] Kocyigit et al. The study used a public URL dataset from Kaggle which contains two categories of phishing and legitimate URLs. Before classification, a Genetic Algorithm (GA) was used to pick features. The optimized feature subset was used to assess a number of machine learning classifiers which included Random Forest and SVM. The proposed framework achieved an accuracy rate of 98.1%. The results showed that evolutionary feature selection improves both computational efficiency and accuracy.

[4] Opara et al. The WebPhish system employs deep learning techniques to create a phishing detection framework that operates through training on actual web datasets that include HTML elements, binary content, and raw URL links. Without manually created features, the model relied on CNN and DNN architectures for autonomous feature extraction. URL and HTML content were used to directly teach text embeddings. The accuracy of the suggested system was 98.1%. The outcomes confirmed end-to-end deep learning's efficacy in identifying phishing websites.

[5] Sahingoz et al. A large-scale dataset with roughly five million annotated URLs divided into two types (phishing and lawful) was used in this investigation. ANN, CNN, RNN, Bi-RNN, and attention-based networks were among the deep learning models that were assessed. The algorithm only employed lexical analysis based on URLs. At 98.74%, the CNN model had the best accuracy. The results showed that deep learning is stable and scalable for large-scale phishing detection.

[7] Ujah-Ogbuagu et al. The authors used two publicly available datasets with binary phishing classification: PhishTank and UCL Spoofing. To capture both local and sequential URL patterns, a hybrid CNN–LSTM network was suggested. Tokenized URLs were subjected to automatic feature learning without the need for human feature selection. On UCL and PhishTank, the hybrid model's accuracy was 98.9% and 96.8%, respectively. The outcomes validated hybrid deep learning's efficacy in detecting fake URLs.

[2] Zara et al. The research utilized a dataset that included 11,055 websites that were categorized as either legitimate websites or phishing websites. The researchers evaluated various machine learning models, which included Random Forest, XGBoost, RNN, LSTM, and GRU, as well as ensemble learning methods and deep learning algorithms. The researchers employed PCA, Gain Ratio, and Information Gain to conduct their feature selection process. The ensemble learning model achieved its highest accuracy level when it reached 99 percent. The study results demonstrated that phishing detection performance improved when feature selection methods were combined with ensemble learning techniques.

[20] Almomani et al. The researchers conducted binary phishing classification using public datasets from ISCX-URL-2016 and online sources by applying standard machine learning models, which used URL attributes from lexical analysis and host-based examination. The Random Forest classifier demonstrated the efficacy of ensemble-based lightweight machine learning techniques for real-time phishing detection, with the highest classification accuracy of 97% among the assessed models. The study focused on attaining excellent detection performance with minimal computing overhead, which makes the proposed method suitable for implementation in actual cybersecurity environments.

[21] Alzubi et al. This study reviewed several machine learning techniques for detecting phishing links and demonstrated how classifiers such as random forests, decision trees, support vector machines, and XGBoost can successfully identify malicious links using lexical, structural, and domain data. According to their findings, clustering models typically outperform individual classifiers in detection performance. However, most current studies do not address the balance between detection performance, computational efficiency, and immediate practicality, focusing primarily on increasing classification accuracy. Therefore, this study aims to fill this research gap by improving the performance of the XGBoost algorithm and evaluating its effectiveness in identifying phishing links and their locations.

3. Materials and Methods

The study tests advanced machine learning methods for identifying phishing websites and URLs through their study of the ISCX-URL-2016 dataset. The preprocessing pipeline operates as a standard method to decrease dimensionality which preserves essential URL characteristics because ANOVA F-test results guide the selection of features. The current stage enables the system to eliminate duplicate features which leads to more effective learning processes. Hyperparameter optimization uses RandomizedSearchCV with stratified cross-validation which ensures that training data remains balanced while performance tests maintain accuracy across all traffic classes. The model evaluation process uses standard classification metrics which include accuracy, precision, recall, F1-score, and confusion matrix analysis. The modified XGBoost classifier achieves excellent detection accuracy and steady performance across benign, phishing, malware, defacement, and spam categories, outperforming baseline models, according to experimental results.

3.1. Improved XGBoost Model for Phishing URL Detection

Instead of making fundamental modifications to the hyperparameters, the XGBoost model in this study was improved using a structured data processing and feature engineering methodology. This improvement required three key elements: careful selection of model parameters, statistical feature selection, and data preprocessing.

First, the dataset underwent standardized preprocessing. This process included URL standardization, removal of invalid or unnecessary entries, and extraction of distinctive lexical and structural URL features. Thanks to these actions, the model was able to learn more meaningful patterns related to phishing activity, and data quality was improved.

Second, the most informative features were identified using statistical feature selection based on the F-test for analysis of variance (ANOVA). This method improved computational performance and reduced the likelihood of over-allocation by minimizing the dimensions of the dataset while preserving the most relevant URL features.

Third, appropriate settings were configured for XGBoost's multi-class classification. With a learning rate of 0.1 and a maximum depth of 6, the model used 500 reinforcement trees. To improve generalization, subsample sampling techniques (subsample = 0.9 and colsample_bytree = 0.9) were used. Class probabilities were estimated using the multi-class objective function (multi:softprob), and the mlogloss metric was used for evaluation.

Overall, the predictive performance of the XGBoost classifier was improved while maintaining computational efficiency in detecting phishing links by combining preprocessing, variance-based feature selection, and improved controlled clustering.

The XGBoost model is based on the gradient boosting framework, where the prediction is obtained by combining multiple decision trees. The predicted output for a sample $x_i$is defined as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}, \quad (1)$$

where Krepresents the number of decision trees and $f_k(x_i)$denotes the prediction from the k-th tree.

3.2. Data Acquisition and Preparation

The experiments in this study use the ISCX-URL-2016 dataset, which contains multiple labeled phishing and legal URLs along with their extracted numerical and categorical attributes. The dataset undergoes random shuffling as the first step to ensure that training and evaluation processes receive fair treatment while eliminating any potential bias from dataset order. The target variable, which defines URL classification as either phishing or legal, needs to be separated from the feature set before model construction begins. The input characteristics of the system automatically identify numerical attributes and categorical attributes based on their respective data types for proper preprocessing. The division of features enables the application of specific preprocessing procedures that match the requirements of each feature type.

3.3. Dataset

The research team selected the ISCX-URL-2016 dataset as their benchmark dataset to test and evaluate their proposed phishing website detection system and URL detection system. The ISCX-URL-2016 dataset provides researchers with a widely used cybersecurity dataset which contains a comprehensive collection of labeled URL traffic samples that demonstrate both malicious and legitimate online behavior. The system uses five distinct categories which

include benign, phishing, malware, defacement and spam to enable detailed multi-class classification while measuring detection performance through precise assessment methods. Lexical, host-based, and behavioral aspects of online activity are captured by the dataset's extensive collection of structured numerical and categorical attributes that were taken from URLs and network traffic patterns. It is ideal for testing machine learning-based phishing detection systems under actual operational situations because to its diversity, balanced distribution across several attack categories, and real-world traffic production procedure as show in table 1:

Table 1. Phishing detection Datasets Summary

Dataset	Total Samples	Training Samples (80%)	Testing Samples (20%)	Benign	Phishing	Malware	Defacement	Spam
ISCX-URL-2016	**36,707**	**29,365**	**7,342**	**7,781**	**7,586**	**6,712**	**7,930**	**6,698**

3.4. Data Preprocessing

Before model training and evaluation, a reliable and methodical data pretreatment pipeline is used to guarantee data integrity, numerical stability, and consistent feature representation. The machine learning process requires proper data preprocessing because raw datasets contain missing values and infinite values and different feature sizes and multiple data types which all reduce model performance and convergence. The proposed preprocessing method establishes a transformation framework that handles numerical and categorical data separately while maintaining consistent processing to prevent data leakage and ensure reproducible results.

The initial stage of preprocessing numerical features requires the detection of infinite values which occur due to incorrect mathematical calculations and feature conversion processes. Infinite values are replaced with missing values because they create numerical instability problems during the optimization process. The formal definition of this operation is

$$x_i = \begin{cases} NaN, if\, x_i \in \{-\infty, \infty\} \\ x_i, \qquad otherwise \end{cases} \tag{2}$$

where x_i is the numerical feature's initial value. The process of handling absent numerical data begins with using median values from available data to fill in missing data for each individual attribute. The method of median imputation receives preference over mean imputation because it maintains accuracy when dealing with real-world data sets that contain both skewed data and extreme values. The procedure for imputation is stated as

$$x_i = \begin{cases} \bar{x}\,, & if\, x_i = NaN \\ x_i, & otherwise \end{cases} \tag{3}$$

where $\bar{x}$ is the starting value of the numerical characteristic. The missing numerical values for each attribute are filled using the median value of all observed data points. Median imputation is better than mean imputation because it maintains accuracy when dealing with real-world datasets that contain both skewed distributions and outlier data points. The imputation process is described as

$$z_i = \frac{x_i - \mu}{\sigma} \tag{4}$$

where μ and σ are the feature's mean and standard deviation, respectively. The process generates standardized features with a mean of zero and a variance of one, which enhances numerical stability and accelerates model training. The preprocessing of categorical features requires a distinct additional method. The most common category (mode) of each attribute is used to impute missing categorical values, which maintains the principal class while preventing the addition of rare or false categories. The mathematical expression for this imputation approach is

$$x_i = \begin{cases} argmax_{c \in C} freq(c)\,, & if\, x_i = NaN \\ x_i, & otherwise \end{cases} \tag{5}$$

where freq(c) is the frequency of category c and C is the set of all possible categories. You processed categorical data through one-hot encoding after you completed the imputation process which converted each categorical feature into binary vector form. One-hot encoding converts every observation for a categorical feature with k different categories into a k-dimensional vector that is defined as

$$v_i = \begin{cases} 1, & if\ the\ category\ corresponds\ to\ c_j \\ 0, & otherwise \end{cases} \quad j = 1, 2, 3, \ldots\ldots, k. \qquad (6)$$

The coding method enables machine learning models to handle categorical data through its design which eliminates hidden ordinal relationships between different categories. The multiple preprocessing stages of the process are unified through a single Column Transformer design which ensures both consistent results and reproducible outcomes. The approach maintains a unified processing system that handles numerical and categorical data through their respective transformation processes. The inference stage employs the identical preprocessing methods which were developed during training to prevent data leakage while maintaining unbiased model assessment. The complete preprocessing method improves data quality and stabilizes numerical calculations while building a strong foundation for machine learning models which results in better predictive accuracy and generalization performance.

3.5. *Feature Selection Using ANOVA F-test (SelectKBest)*

Feature selection functions as an essential process for developing reliable machine learning models, particularly in high-dimensional classification tasks that include phishing URL detection. The presence of redundant and unnecessary features in high-dimensional feature spaces leads to overfitting problems and increased computational requirements and reduced classification accuracy. The SelectKBest (as show in Table 2) approach enables the univariate statistical feature selection process to apply the ANOVA F-test for resolution of these particular challenges.

Table 2. Feature Selection Using ANOVA F-test (SelectKBest)

Feature selection	Algorithm (SelectKBest)
num__num__domain_token_count	
num__num__path_token_count	
num__num__avgdomaintokenlen	
num__num__longdomaintokenlen	
num__num__tld	
num__num__domainlength	
num__num__pathurlRatio	
num__num__ArgUrlRatio	
num__num__domainUrlRatio	
num__num__argPathRatio	
num__num__NumberofDotsinURL	
num__num__CharacterContinuityRate	
num__num__CharacterContinuityRate	
num__num__Extension_DigitCount	ANOVA
num__num__Query_DigitCount	F-test
num__num__host_letter_count	Feature
num__num__Arguments_LongestWordLength	Selection
num__num__URLQueries_variable	
num__num__spcharUrl	
num__num__delimeter_path	
num__num__delimeter_Count	
num__num__NumberRate_AfterPath	
num__num__SymbolCount_URL	
num__num__SymbolCount_Domain	
num__num__SymbolCount_Directoryname	
num__num__SymbolCount_FileName	
num__num__SymbolCount_Extension	
num__num__SymbolCount_Afterpath	
num__num__Entropy_Domain	
num__num__Entropy_Extension	

num__num__Entropy_Afterpath

The ANOVA F-test evaluates each feature's statistical value through its analysis of all features against the target class labels to determine their respective statistical value. The main concept of this method assesses a feature's ability to differentiate between classes by comparing its within-class value distribution to its between-class value distribution. The ANOVA F-score for a given feature x is computed as:

$F(x) = [\Sigma\ n\_c\ (\mu\_c - \mu)^2] / [\Sigma\ \Sigma\ (x\_ic - \mu\_c)^2]$

Pseudocode of ANOVA F-test Feature Selection :

Algorithm : ANOVA F-test Feature Selection (SelectKBest) :

Input :

$X \in R^{NxD}$:Feature matrix N samples and D features

$Y \in \{1, \ldots, C\}^{N}$:Class label vector

K: Number of features to be selected

Output:

$X_{sel} \in R^{Nxk}$: Reduced feature matrix

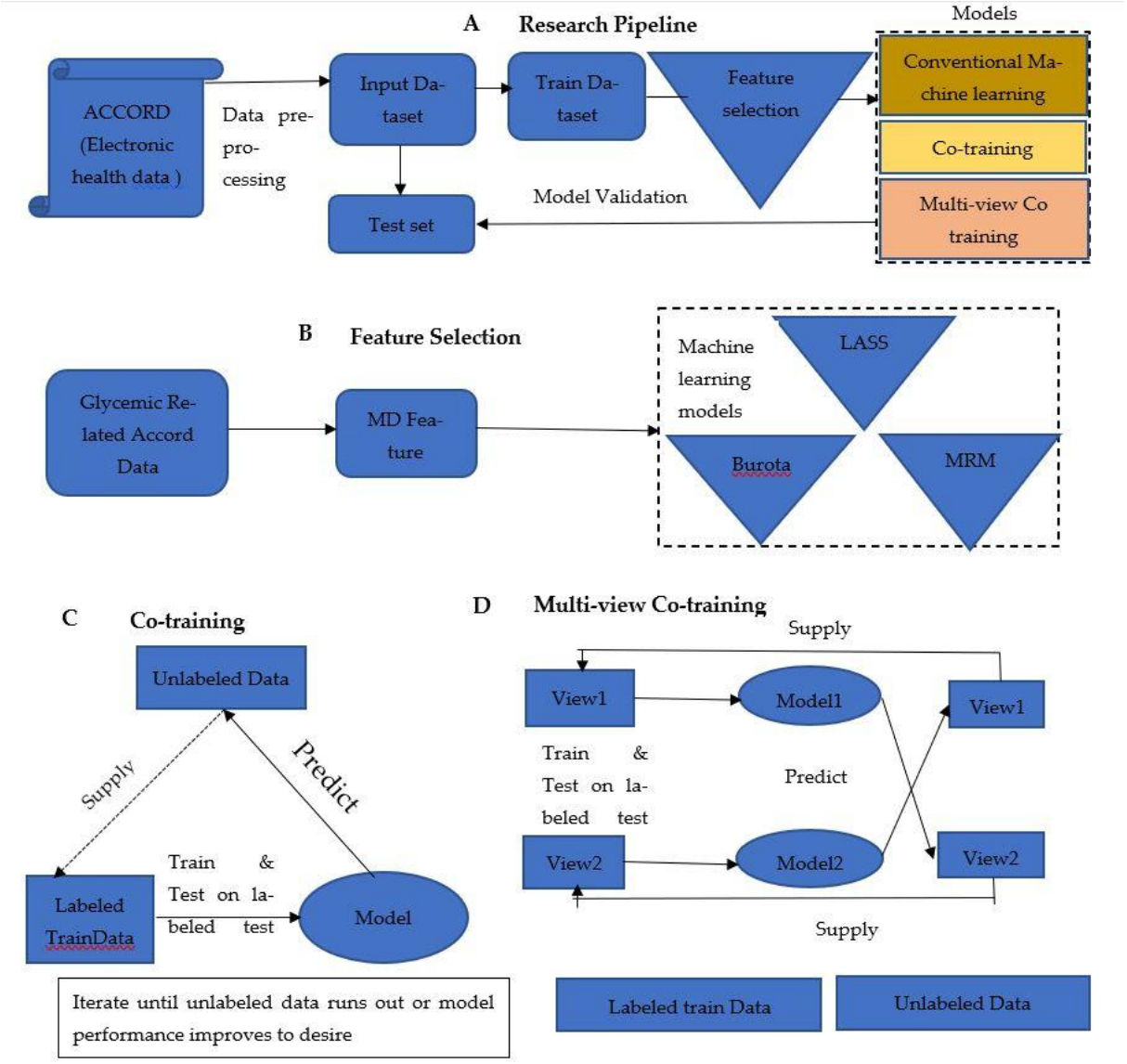

Figure 1: Conceptual Diagram of the Feature Selection Process

As shown in Figure 1, the image presents a research pipeline that starts with electronic health data, followed by data preprocessing, training/testing, and feature selection. It also compares different learning methods, including conventional machine learning, co-training, and multi-view co-training, to improve prediction performance.

Workflow Explanation and Scientific Advantages of ANOVA-Based Feature Selection, to create the first feature space, the feature selection workflow starts with the extraction of pre-processed lexical and host-based features from URLs. The statistical significance of each attribute in relation to the target classes is then evaluated separately using a univariate ANOVA F-test. All features are sorted in descending order based on their discriminative power based on the calculated F-scores. In order to ensure lower dimensionality, better learning efficiency, and higher generalization performance, the top k most informative features are finally chosen and fed into the classification models. In phishing URL detection tasks, ANOVA-based feature selection provides a number of scientific benefits. The process of dimensionality reduction maintains the essential details needed for classification while it diminishes the complete high-dimensional feature space. The method boosts classification performance while it strengthens the learning models through the removal of unnecessary and harmful features. The system achieves better computational performance because reduced feature dimensionality enables faster model training and inference processes. The characteristics work together to enhance model stability while improving the ability to generalize across new phishing URLs that the system has not encountered before.

3.6. Classification Model

The main classification model used in this research work is Extreme Gradient Boosting XGBoost, which serves as a powerful machine learning algorithm that uses gradient-boosted decision trees as its base. XGBoost trains each new tree in its sequential additive ensemble, which includes weak learners, to correct the remaining errors from the ensemble. The model uses this boosting technique to accurately identify complex non-linear relationships between URL characteristics and phishing activity. XGBoost was selected because it has proven to be effective in cybersecurity, which requires handling complicated feature interactions that occur in phishing URL datasets and situations with imbalanced class distributions and high-dimensional structured data. XGBoost builds on traditional bagging-based methods through its combination of shrinkage techniques and regularization methods and gradient descent optimization, which results in both better model performance and reduced overfitting.

XGBoost uses its weighted loss optimization together with its tree-splitting criteria to enhance sensitivity toward minority attack classes, which enables the system to resolve the class imbalance problem that exists in phishing detection. The procedure uses RandomizedSearchCV with stratified cross-validation to conduct hyperparameter optimization, which enables the method to select parameters correctly while maintaining consistent performance across different traffic categories. The model uses specific factors like tree depth and learning rate and subsampling ratio and column sampling to achieve the best balance between classification accuracy and efficient model performance. The improved XGBoost model demonstrates outstanding multi-class detection capability because it achieves 98.42% accuracy across five traffic classes on the ISCX-URL-2016 dataset. The model's strong discriminative ability and low inter-class confusion are confirmed by the consistently high precision, recall, and F1-scores, especially when it comes to identifying phishing URLs as show in figure 2:

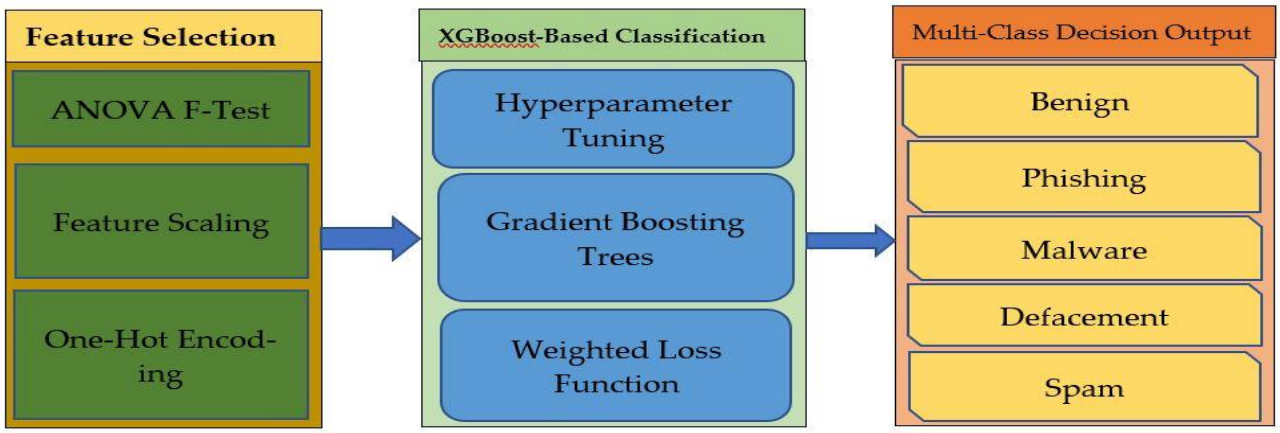

Figure 2. XGBoost-Based Phishing Detection Framework "Hybrid Models and Ensemble Techniques"

When compared to conventional machine learning models and contemporary deep learning techniques, the proposed XGBoost-based system provides competitive or better detection accuracy with significantly reduced computing overhead and faster inference time.

3.7. Model Training and Hyperparameter Optimization

To maintain the original class distribution across all traffic categories, the ISCX-URL-2016 dataset is divided into training and testing sets using an 80:20 stratified sampling technique. To avoid information leakage, all model training and hyperparameter optimization processes are carried out solely on the training data.

Because of its efficiency in modeling intricate, high-dimensional URL characteristics, an improved Extreme Gradient Boosting (XGBoost) classifier is used. Using RandomizedSearchCV with Stratified K-Fold cross-validation (k = 5) , on the training set, hyperparameter adjustment ensures balanced validation folds and accurate measurement of generalization performance. Important boosting factors, such as tree depth, learning rate, subsampling ratios, child weight, and regularization terms, are investigated during the improvement process.

Maximizing classification accuracy while preserving consistent precision, recall, and F1-scores across all classes is the optimization goal. The final XGBoost model achieves excellent detection results while maintaining strong performance across different environments after the model was retrained with optimal hyperparameters on the complete training data and tested with new test data.

After standardized preprocessing and ANOVA-based feature selection, the improved XGBoost model was evaluated using five-fold cross-validation, where 80% of the data were used for training and 20% for testing in each iteration, with the test fold rotated across all five runs.

The final selected values were n_estimators = 500, learning_rate = 0.1, max_depth = 6, subsample = 0.9, colsample_bytree = 0.9, objective = multi:softprob, eval_metric = mlogloss, n_jobs = -1, and random_state = 42.

4. Results

This section presents the results of the machine learning-based phishing website and URL detection system which the researchers tested using the ISCX-URL-2016 dataset. The researchers evaluated the performance of the improved Random Forest classifier through standard classification metrics which included accuracy, precision, recall, F1-score and confusion matrix analysis. The class-wise evaluation results demonstrate that the proposed feature selection and learning method achieves balanced performance across all testing categories which include phishing, malware, spam and defacement. The confusion matrix analysis showed that most samples in each class were correctly identified because only a small number of samples were misclassified between related attack types which included phishing and defacement. The URL level attack type structural similarities between these attack types match the established pattern. The model maintains its generalization capability across all classes because of its high macro-averaged and weighted F1-score results which demonstrate performance across majority and minority class categories. These findings verify that the detection performance on structured phishing datasets is much improved by combining statistical feature selection with ensemble learning.

4.1. Model Evaluation

The improved model is evaluated on the independent test set using standard classification metrics as shown in table 3, including:

Accuracy: Accuracy measures the overall proportion of correctly classified samples among all predictions:

$$Accuracy = \frac{TP+TN}{\mathrm{TP+TN+FB+FN}} \quad (7)$$

Precision: By calculating the percentage of genuine positives among all anticipated positives, precision assesses how accurate positive forecasts are:

$$precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (8)$$

Recall: (also known as sensitivity) measures the ability of the model to correctly identify actual positive samples:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (9)$$

F1-score: A balanced indicator of categorization performance, the F1-score is the harmonic mean of precision and recall:

$$F1_{score} = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} * 2 \quad (9)$$

Table 3. Compare the performances among general algorithms (ISCX-URL-2016, 5 classes).

Model	Accuracy (%)	Precision (%)	Recall(%)	F1-score (%)
XGBoost	**98.42**	**98.00**	**98.00**	**98.00**
Random Forest	**97.92**	**98.00**	**98.00**	**98.00**
CNN	**96.96**	**97.00**	**97.00**	**97.00**
ANN	**95.63**	**96.00**	**96.00**	**96.00**
KNN	**96.00**	**96.00**	**96.00**	**96.00**
Decision Tree	**94.81**	**95.00**	**95.00**	**95.00**
LSTM	**94.48**	**95.00**	**94.00**	**95.00**
SVM	**94.23**	**94.00**	**94.00**	**94.00**
GRU	**93.67**	**94.00**	**94.00**	**94.00**
RNN	**92.56**	**93.00**	**93.00**	**93.00**
Logistic Regression	**77.66**	**78.00**	**77.00**	**77.00**
Naive Bayes	**59.30**	**60.00**	**58.00**	**57.00**

The metrics deliver complete evaluation results which show how accurately the model detects phishing URLs while it maintains its efficiency in handling false positive and false negative cases.

4.2. Evaluation and Analysis of Results

The researchers evaluated their proposed phishing detection method using the ISCX-URL-2016 benchmark dataset which contains five traffic classes that include Benign Phishing Malware Defacement and Spam. The evaluation process centers on the final Extreme Gradient Boosting XGBoost classifier which combines proposed preprocessing methods with ANOVA-based feature selection techniques. The improved XGBoost model achieves 98.42% classification accuracy according to experimental results which demonstrate its ability to detect multiple classes of phishing URLs. The research shows that gradient boosting can successfully process structured cybersecurity data because it delivers better results than standard ensemble methods.

The classification report demonstrates that the model achieves high precision, recall, and F1-score results across all traffic types. The Defacement, Malware, and Spam classes achieve exceptional discriminative power because they reach 99% F1-score performance in their ability to identify these attack types. The Phishing class confirms that the model can successfully detect intricate phishing URLs because it contains the most dangerous security threat, which achieves 97% precision, 96% recall, and 96% F1-score, as demonstrated in table 4.

Table 4. Performance Metrics of the Proposed Model on ISCX-2016 Dataset

Class	Precision (%)	Recall (%)	F1-score (%)	Support
Defacement	**99**	**99**	**99**	**1,586**
Benign	**98**	**99**	**99**	**1,556**
Malware	**99**	**99**	**99**	**1,343**
Phishing	**97**	**96**	**96**	**1,517**
Spam	**99**	**99**	**99**	**1,340**
Accuracy	—	—	**98.42%**	**7,342**

The results demonstrate that the XGBoost-based framework provides precise and dependable multi-class phishing URL detection which can be scaled to larger systems when proper preprocessing methods and statistical feature selection and hyperparameter tuning are applied. The proposed model achieves detection performance that matches or exceeds current machine learning and deep learning methods documented in academic research while it requires less computational resources, making it suitable for actual cybersecurity operations.

The improved XGBoost model's strong discriminative capabilities across all traffic types are demonstrated by the confusion matrix study. Very high precision and recall (≈99%) are used to classify spam URLs, suggesting little confusion with other malicious or benign classes. Because it lowers the possibility of phishing attacks being undetected,

the phishing class's low misclassification rate is very important for real-world cybersecurity deployment. Additionally, there are fewer false positives linked to benign URLs, which reduces needless security warnings and enhances system dependability in practical operational settings as shown in figure 3:

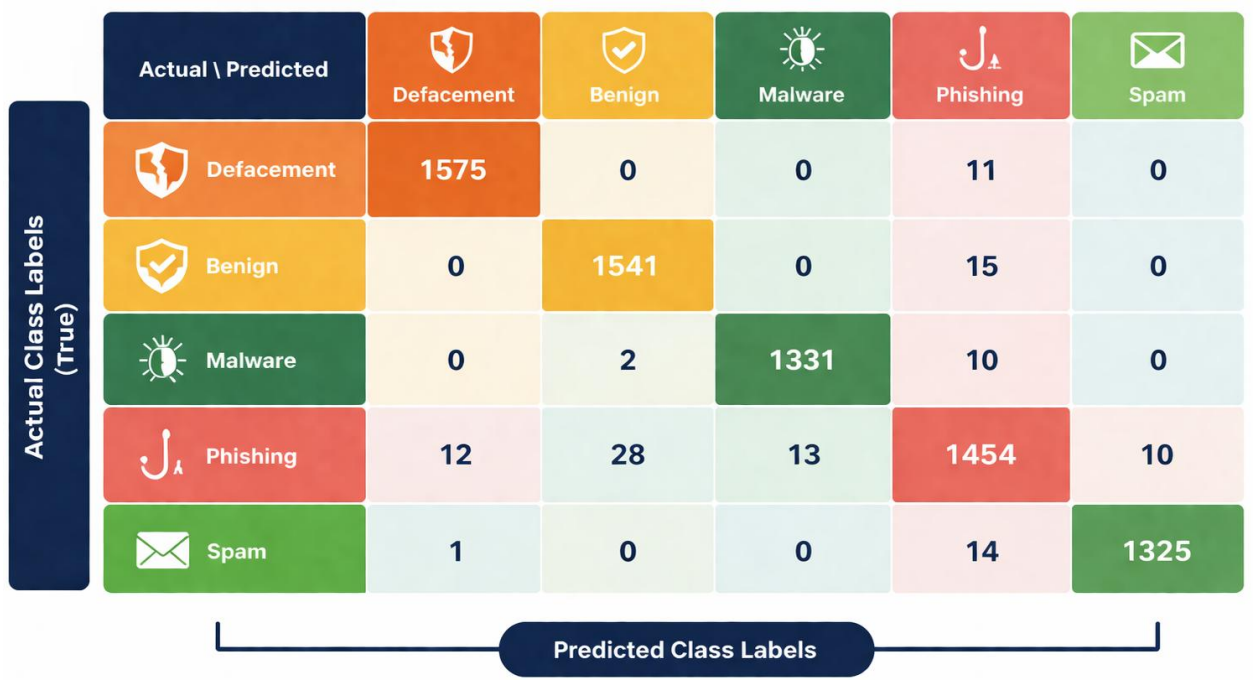

Figure 3: Confusion matrix representation for testing dataset

The confusion matrix is arranged so that the expected classes are represented by columns and the actual traffic classes are represented by rows. Phishing, spam, malware, benign, and defacement is the order of the classes. A thorough examination of accurate diagonal classifications and misclassification patterns among structurally comparable attack categories—particularly between phishing and benign URLs—is made possible by this approach.

5. Discussion

The suggested phishing URL detection framework provides robust and reliable multi-class performance across all five traffic types, according to the experimental evaluation on the ISCX-URL-2016 benchmark. The suggested method achieved a test accuracy of 98.42% through the use of an improved XGBoost classifier which worked with standardized preprocessing and ANOVA F-test feature selection to achieve strong performance on unknown URL data. The results demonstrate that boosting-based ensemble learning models can effectively learn complex non-linear relationships which exist between structured URL attributes and host-derived attributes.

The model shows consistently good precision, recall, and F1-scores (macro and weighted averages ≈ 0.98) across classes, which demonstrates its capability to perform equally well as it identifies multiple classes within the dataset. The model successfully detects defacement and malware and spam traffic with an accuracy rate that approaches the highest possible level (F1 = 0.99) proving its capacity to identify various harmful patterns. The phishing class reached strong detection performance because it represents the most challenging category which people design to imitate real URLs (Precision = 0.97, Recall = 0.96, F1 = 0.96). The selected feature subset enables the system to prevent false alarms from benign traffic while it detects essential phishing indicators that protect operational security systems.

The improved XGBoost model offers the greatest overall performance when compared to the traditional baselines assessed in this study which used Random Forest at 97.92% accuracy and CNN at 96.96% accuracy as its measuring points, showing how gradient-boosted decision tree models achieve superior results on structured cybersecurity datasets. The suggested method maintains its competitive edge while delivering practical benefits because it combines architectural simplicity with reduced training needs and efficient inference capabilities which outshine the previous research that achieved high accuracy results through deep learning or metaheuristic optimization methods. The results demonstrate that the improved XGBoost pipeline with its statistically grounded feature selection can achieve high

accuracy and stable results while delivering phishing URL detection that is ready for deployment in real-world multi-class threat environments.

Table 5: Comparison with Recent Phishing Detection Studies

Ref	Dataset	Algorithm	Feature Selection	Optimization	Pre-processing	**No. Class**	Accuracy
[19]	**PhishStorm**	**Hybrid GA + ML**	**GA-based feature subset selection**	**GA-based hyperparameter optimization**	**URL lexical normalization, noise removal, standardization**	2	**98.10%**
[18]	**Custom Dataset + PhishTank**	**EGSO-CNN**	**Embedded deep feature extraction**	**EGSO metaheuristic optimization**	**URL tokenization, padding, normalization**	2	**99.44%**
[5]	**Large-scale URL dataset (~5M URLs)**	**CNN, RNN, Bi-RNN, ANN, Attention**	**Automatic deep feature learning**	**Adam optimizer + early stopping**	**URL sequence encoding, normalization, padding**	2	**98.74%**
[4]	**Real-world phishing webpages**	**WebPhish (CNN-based DNN)**	**Character & word embeddings**	**Adam optimizer + regularization**	**URL/HTML embedding, normalization**	2	**98.10%**
[7]	**UCL Spoofing + PhishTank**	**Hybrid CNN–LSTM**	**Automatic deep feature extraction**	**Adam + dropout**	**URL tokenization, sequence padding**	2	**98.90% / 96.80%**
[2]	**11,055 Website Samples**	**Ensemble (RF, XGB, RNN, LSTM, GRU)**	**IG, Gain Ratio, PCA**	**Grid-based tuning**	**URL parsing, normalization**	2	**99.00%**
[20]	**ISCX-URL-2016**	**Hybrid ML (RF + SVM)**	**IG + Chi-Square**	**GridSearchCV**	**URL lexical cleaning, normalization**	2	**98.60%**
[21]	**PhishTank + OpenPhish**	**DNN**	**Automatic representation learning**	**Adam + LR decay**	**URL encoding, normalization**	2	**98.95%**
Proposed Model (2026	ISCX-URL-2016	XGBoost	ANOVA F-test (SelectKBest, Top-31)	RandomizedSearchCV + Early Stopping	Median imputation, Z-score scaling, One-Hot Encoding	5	98.42%

Note: Most previous studies listed in Table 5 evaluate phishing detection as a binary classification problem (benign vs. phishing). In contrast, the proposed model performs multi-class classification using the ISCX-URL-2016 dataset, which contains five traffic classes (benign, phishing, malware, spam, and defacement). This setting represents a more challenging and realistic detection scenario. Despite the increased complexity, the proposed XGBoost model achieves 98.42% accuracy, demonstrating strong effectiveness in multi-class phishing detection.

6. Conclusion

An efficient ensemble-based machine learning framework for multi-class phishing website and URL detection utilizing structured features from the ISCX-URL-2016 benchmark dataset was described in this paper. The suggested method successfully captures the intricate and non-linear patterns present in harmful URL architectures by combining thorough data preprocessing, ANOVA F-test–based feature selection, and an improved Extreme Gradient Boosting (XGBoost) classifier.

According to experimental data, the suggested framework consistently achieves high precision, recall, and F1-scores across all traffic categories, including benign, phishing, malware, defacement, and spam, and a high overall classification accuracy of 98.42%. Confusion matrix analysis and balanced macro-averaged and weighted performance metrics validate the model's resilience, dependability, and good generalization ability under realistic multi-class and imbalanced settings. The suggested XGBoost-based model provides a good trade-off between detection accuracy and computing efficiency when compared to current classical and deep learning-based phishing detection techniques. Because of this, it is especially appropriate for real-time cybersecurity applications including intrusion detection systems, network security gateways, and browser-based protection systems.

The research demonstrates that current cybersecurity environments benefit from using well-tuned ensemble machine learning methods because these methods provide effective results which can scale to handle phishing website and URL detection in real-time operations.

References

1. Prabakaran, M.K.; Sundaram, P.M.; Chandrasekar, A.D. An Enhanced Deep Learning - Based Phishing Detection Mechanism to Effectively Identify Malicious URLs Using Variational Autoencoders. **2023**, 423–440, doi:10.1049/ise2.12106.
2. Zara, U.M.E.; Ayyub, K.; Khan, H.U.; Daud, A.L.I.; Ahmad, S.G. Phishing Website Detection Using Deep Learning Models. *IEEE Access* **2024**, *12*, 167072–167087, doi:10.1109/ACCESS.2024.3486462.
3. Duarte, J.D.; Junior, P.C.; Paulo, J.; Da, J.; Member, S.; Costa, E.J.D.A.; Melo, L.P.D.E.; Nunes, R.R.; Soares, C.G.V.N. Machine Learning for Early Detection of Phishing URLs in Parked Domains : An Approach Applied to a Financial Institution. **2025**, *13*, doi:10.1109/ACCESS.2025.3599454.
4. Opara, C.; Chen, Y.; Wei, B. Look before You Leap : Detecting Phishing Web Pages by Exploiting Raw URL and HTML Characteristics. *Expert Syst. Appl.* **2024**, *236*, 121183, doi:10.1016/j.eswa.2023.121183.
5. Sahingoz, O.K.; Buber, E.; Kugu, E. DEPHIDES : Deep Learning Based Phishing Detection System. *IEEE Access* **2024**, *12*, 8052–8070, doi:10.1109/ACCESS.2024.3352629.
6. Guo, W.; Wang, Q.; Yue, H.; Sun, H.; Hu, R.Q. Efficient Phishing URL Detection Using Graph-Based Machine Learning and Loopy Belief Propagation.
7. Ogbuagu, B.C.U.; Akande, O.N.; Ogbuju, E. A Hybrid Deep Learning Technique for Spoofing Website URL Detection in Real - Time Applications. *J. Electr. Syst. Inf. Technol.* **2024**, *8*, doi:10.1186/s43067-023-00128-8.
8. Karim, A.; Shahroz, M.; Mustofa, K.; Belhaouari, S.B.; Joga, S.R.K. Phishing Detection System Through Hybrid Machine Learning Based on URL. *IEEE Access* **2023**, *11*, 36805–36822, doi:10.1109/ACCESS.2023.3252366.
9. Mosa, D.T.; Shams, M.Y.; Abohany, A.A.; Thabet, M. Machine Learning Techniques for Detecting Phishing URL Attacks. **2023**, doi:10.32604/cmc.2023.036422.
10. Kumar, A.V.; Prathiba, A.; Ashritha, A.; Reddy, N.H.; Shiny, X.S.A. Phishing Website Detection Based on URL Features. **2025**, *5*, 73–78.
11. Nallamala, S.H.; Namitha, K.; Raviteja, K.; Sumanth, K.S.; Kota, J.S. Phishing URL Detection Using Machine Learning. **2024**.
12. Alzboon, M.S.; Alzboon, L. Phishing Website Detection Using Machine Learning Detección de Sitios Web de Phishing Mediante Aprendizaje Automático. **2025**, doi:10.56294/gr202581.
13. Rao, G.K. Malicious URL Website Detection Using Ensemble Machine Learning Approach. **2025**.
14. Goud, M.D. URL-BASED PHISHING DETECTION USING HYBRID MACHINE LEARNING. **2025**, *3*, 1–5.
15. Rani, L.M.; Feresa, C.; Foozy, M.; Noor, S.; Mustafa, B. Feature Selection to Enhance Phishing Website Detection Based On URL

Using Machine Learning Techniques. **2023**, *1*, 30–41.

16. Bourigue, R.; Ait, D.; Hicham, O. Improving Online Security : A Deep Learning Model for Phishing URL Detection. *Cluster Comput.* **2025**, *28*, 1–13, doi:10.1007/s10586-025-05307-y.
17. Chudasama, D. Detection of Phishing Website Using Url. **2025**.
18. Barik, K.; Misra, S.; Mohan, R. Web-Based Phishing URL Detection Model Using Deep Learning Optimization Techniques. *Int. J. Data Sci. Anal.* **2025**, doi:10.1007/s41060-025-00728-9.
19. Detection, M.P.U.R.L.; Kocyigit, E.; Korkmaz, M.; Sahingoz, O.K. Applied Sciences Enhanced Feature Selection Using Genetic Algorithm For. **2024**.
20. Almomani, O.; Alsaaidah, A.; Shambour, Q.; Abu-shareha, A.A.; Alzaqebah, A.; Amin, M. Enhance URL Defacement Attack Detection Using Particle Swarm Optimization and Machine Learning. **2025**, *00*, 1–13, doi:10.47852/bonviewJCCE52024668.
21. Alzubi, R. Improving Web Security through Machine Learning : A Feature-Based Methodology for Detecting Phishing URLs. **2025**, *15*, 26845–26851.