International Journal of Safety and Security Engineering
Vol. 14, No. 6, December, 2024, pp. 1783-1794
Journal homepage: http://iieta.org/journals/ijsse

Phishing Detection Using Random Forest-Based Weighted Bootstrap Sampling and LASSO$^+$ Feature Selection

Wendy Sarasjati, Supriadi Rustad*, Purwanto, Heru Agus Santoso, De Rosal Ignatius Moses Setiadi

Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang 50131, Indonesia

Corresponding Author Email: srustad@dsn.dinus.ac.id

https://doi.org/10.18280/ijsse.140613

Received: 24 September 2024
Revised: 28 November 2024
Accepted: 16 December 2024
Available online: 31 December 2024

Keywords:
phishing detection, LASSO-based feature selection, Random Forest, Weighted Bootstrap Sampling, outlier handling, cybersecurity, predictive performance, computational efficiency

ABSTRACT

Phishing attacks are becoming more complex and harder to differentiate from legitimate websites. This poses serious risks to users and organizations. This study introduces a phishing detection framework that combines LASSO-based feature selection and a Random Forest classifier enhanced by Weighted Bootstrap Sampling (WBS). The framework addresses two key challenges: optimizing feature selection for high-dimensional data and managing datasets with over 70% outliers. LASSO$^+$ extends the traditional LASSO (Least Absolute Shrinkage and Selection Operator) by integrating Pearson Correlation and Grid Search. This combination improves feature selection by identifying the most relevant features, reducing redundancy, and ensuring efficient processing without compromising accuracy. WBS further enhances Random Forest by prioritizing uncertain samples during training, enabling the model to effectively handle outlier-heavy datasets and improve recall. The proposed framework was evaluated on four diverse datasets with distinct challenges. Results demonstrated high recall rates of 99.59% for Dataset A, 98.76% for Dataset B, 100.00% for Dataset C, and 98.99% for Dataset D. The method also achieved competitive execution times. Compared to existing approaches, the framework delivered better predictive accuracy, robustness, and efficiency. This study highlights the advantages of combining LASSO$^+$ and WBS to improve feature selection and manage outliers in phishing detection. The proposed method provides a reliable solution for addressing cybersecurity challenges in practical applications.

1. INTRODUCTION

Phishing attacks have evolved significantly, making it increasingly difficult for users to differentiate between legitimate and malicious websites. According to the Anti-Phishing Working Group (APWG), the number of phishing incidents has increased by 15.87%, from 832,000 in 2022 to 964,000 in 2024 [1-3]. These attacks endanger sensitive data and erode users' trust in online platforms and services [4]. Phishing websites use a technique of URL concealment to trick users and bypass traditional security measures [5-7]. The changing nature of phishing URLs adds another complexity to their detection. Feature selection techniques can address this issue by reducing complexity and enhancing the model's capacity to accurately differentiate between various categories of websites [8]. For instance, Hannousse and Yahiouche [8] demonstrated that implementing feature selection strategies improved accuracy to 96.86%. Therefore, there is a need for more effective detection and protection systems.

Traditional feature selection methods often involve trade-offs between predictive performance and execution time. Achieving high accuracy frequently requires more features, which increases computational cost and training time [9]. Reducing features may improve computational efficiency, but it risks eliminating key characteristics and potentially lowering accuracy [8]. These limitations make traditional approaches inadequate for handling complex datasets where both high accuracy and computational efficiency are essential. To overcome these limitations, we introduce LASSO$^+$, a feature selection method designed to optimize performance and efficiency simultaneously. In contrast to previous studies [10] emphasizing feature count, we focus on balancing performance and efficiency. This method ensures that computational resources are not overburdened while maintaining high accuracy, aligning with Pudjihartono et al., who emphasized balancing feature selection to optimize both performance and computational resources [11].

LASSO$^+$ is an advanced feature selection method combining LASSO (Least Absolute Shrinkage and Selection Operator) with correlation thresholding and Grid Search to enhance feature selection efficiency. Details of its implementation are provided in Section 3.2. Correlation thresholding helps to eliminate highly correlated features, while Grid Search fine-tunes the regularization parameter (λ) to optimize feature selection. Balancing these trade-offs between recall, performance, and computational efficiency is critical, as highlighted by recent studies [11, 12]. This method balances performance and computational efficiency by automatically selecting the most important features during learning, reducing the impact of less important ones, and

simplifying the model without losing effectiveness [11, 13-17].

In phishing detection, recall—also known as the True-Positive Rate (TPR)—is essential because it ensures that all potential threats are detected by minimizing the risk of missing any phishing attempts. The primary objective is to identify every possible threat [15]. Even though emphasizing recall might increase computational demands, it remains a critical priority. Missing a single phishing attempt may result in significant security threats.

Another issue in phishing detection is managing the high proportion of outliers in the datasets. More than 70% of the data points were identified as outliers using the Interquartile Range (IQR) method. This proportion of outliers can impact the performance of traditional machine learning models and often leads to inaccurate predictions [16, 17].

To solve these problems, the LASSO$^+$ model integrates with Random Forest (RF) enhanced by Weighted Bootstrap Sampling (WBS) [18]. This integrated method improves predictive performance and effectively handles imbalanced and outlier-prone phishing datasets. WBS addresses the problem of outliers by ensuring a balanced sample representation during model training, which enhances recall and overall accuracy. Moreover, RF is well-known for its accuracy in multi-class datasets, user-friendliness with different dataset sizes, and stability. Conventional feature selection methods often necessitate a trade-off between maximizing predictive performance and minimizing execution time, which this study aims to address [10, 14, 19, 20].

The primary contributions of this study are threefold. First, we introduce and carefully evaluate the LASSO$^+$ feature selection method for phishing detection. This method strategically combines LASSO with correlation thresholding and Grid Search to optimize feature selection. Second, we investigate the impact of integrating Weighted Bootstrap Sampling with Random Forest. We assess its effectiveness in improving model predictive performance, especially in handling outliers phishing datasets. Third, we provide a detailed analysis of the trade-offs between predictive performance and execution time. This analysis offers insights into how these factors can be effectively balanced to achieve superior performance in phishing detection systems.

2. PRELIMINARY STUDY

2.1 LASSO feature selection

LASSO is increasingly used in phishing detection because it is efficient in handling high-dimensional data. It reduces overfitting and enhances interpretability by shrinking less important feature coefficients to zero [17]. It has been effectively combined with mRMR and machine learning techniques to improve feature selection and accuracy [21]. In malicious URL detection, integrating LASSO with models like RF creates robust systems [19]. This study chose LASSO as a key technique because it can balance accuracy and computational efficiency.

LASSO$^+$ improves traditional LASSO by incorporating correlation thresholding and Grid Search. Correlation thresholding removes redundant and highly correlated features, reducing noise and multicollinearity, as demonstrated in tensor factor models [22, 23]. This process enhances phishing detection by retaining only the most informative and independent features, which is critical for improving both model accuracy and interpretability.

Grid Search fine-tunes the λ parameter in LASSO$^+$, ensuring an optimal balance between feature reduction and performance [24, 25]. This adaptive approach is essential as phishing attacks grow more sophisticated, including techniques like Generative Adversarial Networks (GAN) [26]. Thus, LASSO$^+$ helps address the challenge of building scalable detection systems that adapt to evolving phishing threats.

2.2 Random Forest

RF is a widely used ML algorithm known for its robustness in handling large and complex datasets. It was developed as an ensemble method that combines multiple decision trees to improve predictive performance and reduce overfitting. This approach makes RF stable and effective in many applications, particularly useful in classification tasks with high data variability [27, 28].

Several studies have demonstrated RF's effectiveness in phishing detection. For instance, Almseidin et al. [29] found that integrating RF with optimized feature selection extensively improves phishing detection systems' accuracy. Al-Sarem et al. [30] improved this approach by combining RF into an optimized stacking ensemble model, which achieved high accuracy and reduced overfitting. Othman and Hassan [31] conducted an empirical study that further reinforced RF's dominance in phishing detection. Studies conducted by Kandula et al. have identified RF as one of the most effective algorithms for phishing detection, attributed to its robustness and capability in managing high-dimensional data [6]. Their findings implied that RF consistently outperformed other models despite high data variability.

In the current study, RF is integrated with the LASSO$^+$ feature selection framework to optimize both predictive performance and computational efficiency. Recognizing the unique challenge posed by outlier datasets, we introduce the use of WBS within the RF model. WBS focuses on resampling data based on uncertainty in model predictions, enhancing the model's ability to correctly classify difficult samples, including those not explicitly labeled as outliers. This combination of techniques positions the study to contribute significantly to phishing detection. The model can also handle data that is challenging to detect, even if they are not classified as outliers. This is achieved using uncertainty-based resampling techniques, as previously demonstrated in studies [18, 32].

3. METHODOLOGY

The proposed method, as illustrated in Figure 1, is employed to enhance the performance of phishing website detection. This process includes data preparation, advanced feature selection, and robust classification techniques. It begins with data preparation, where duplicate entries and null values of the dataset are removed. During this stage, outliers are identified and permitted to be accounted for in subsequent analysis. Afterward, the data is normalized using a Min-Max Scaler. This scaler adjusts all features to the same scale between 0 and 1. It provides a simple feature range without introducing negative values or producing large ranges. Thus, the model is easier to interpret and reduces computational complexity.

Once the data is prepared, the study moves on to LASSO+ feature selection. The process starts with Pearson Correlation to identify and measure the linear relationship between features. Features with high correlations are subsequently subjected to Correlation Thresholding. In Correlation Thresholding, redundant features are removed to reduce multicollinearity and simplify the model. Grid Search is employed to optimize the parameters for LASSO and refine the feature selection further. This ensures that only the most relevant features are selected efficiently. Then, LASSO is employed to select the final set of features.

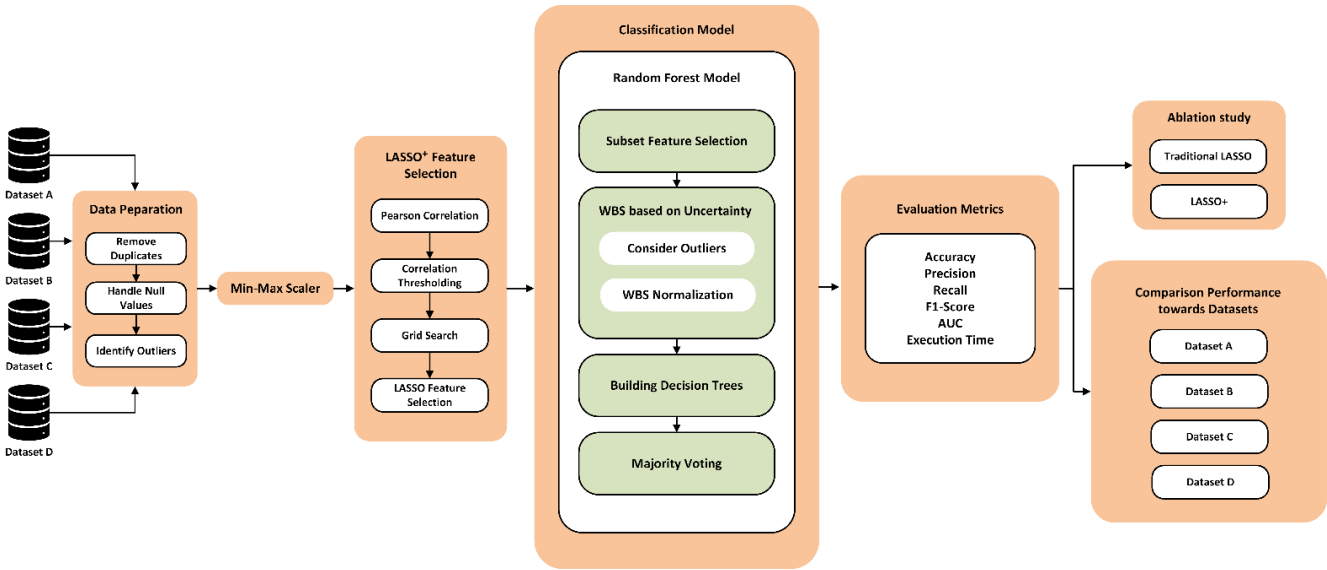

Figure 1. Proposed method for phishing detection using Random Forest based on LASSO+ feature selection

The selected features are then fed into the RF model, which is enhanced by WBS based on Uncertainty. This technique prioritizes samples that are difficult to classify, with one significant factor being outliers identified through the data preparation stage. Within the RF model, subset feature selection is applied at each node of the decision trees. Rather than using all available features, the model randomly selects a subset of features at each node to reduce overfitting and increase model diversity.

WBS normalization adjusts each sample's weights based on uncertainty to ensure their sum equals one. Then, the decision trees were built using these weighted samples and the selected subset of features. Each tree is constructed by splitting nodes based on a subset of features with higher uncertainty values. The final prediction is determined by aggregating the predictions from all decision trees in the forest using majority voting, where the class predicted by most trees is selected as the outcome.

Finally, the model's performance is evaluated using a comprehensive set of metrics, including accuracy, precision, recall, F1-score, Area Under the Curve (AUC), and execution time. These metrics provide a holistic view of the model's effectiveness. An ablation study is also conducted to compare the traditional LASSO method with the proposed LASSO+ method. This highlights the improvements in recall and computational efficiency. The methodology concludes with a Comparison of Performance across four datasets (A, B, C, and D).

3.1 Data collection

This study utilized several public datasets to evaluate the performance of different feature selection methods for phishing website detection. The datasets used are from Vrbančič (Dataset A) [33], Hannousse and Yahiouche (Dataset B) [8], Prasad and Chandra (Dataset C) [34], and Mohammad et al. (Dataset D) [35]. These datasets are widely used in phishing detection research. For example, Dataset A has been utilized in various studies as a benchmark for evaluating feature selection methods in phishing detection [31, 36, 37]. Dataset B is often referenced in studies analyzing phishing characteristics [10, 38-40]. Due to its large size and inclusion of modern attributes such as obfuscation and media elements, Dataset C has been used to test advanced machine-learning algorithms for phishing detection [41]. Finally, Dataset D represents a smaller but balanced dataset [3, 42]. The details of these datasets are summarized in Table 1.

Table 1. Dataset information

Dataset Characteristic	Dataset A	Dataset B	Dataset C	Dataset D
Total Data	88,647	11,430	235,795	11,055
Features	96	87	53	30
Phishing	30,647	5,715	100,945	4,898
Non-Phishing	58,000	5,715	134,850	6,157
Description of Features	URL structure, Special characters count, Response times, Google index status	URL structure, Character counts, HTTP headers, WHOIS data	URL structure, Obfuscation, HTTPS usage, Form submissions, Media elements	URL Structure, Special Characters count, HTTPS Usage, Media elements, WHOIS Data

The datasets vary in feature count, class distribution, and overall size, reflecting diverse characteristics essential for evaluating feature selection methods. Dataset A has the highest feature count and a noticeable imbalance, with legitimate samples dominating. In contrast, Dataset B provides a balanced class distribution. Dataset C is the largest in size, exhibits moderate imbalance, and incorporates modern phishing characteristics. Dataset D, with the smallest feature count and near-balanced classes, offers efficiency for lightweight evaluations. These differences highlight the complementary nature of the datasets, ensuring robust and comprehensive testing under varied conditions.

3.2 Data preparation

This study focused on handling outliers in the datasets. Outliers may impact the performance of ML models. More than 70% of the data were identified as outliers using the IQR method. Figure 2 presents the outlier sensitivity graph for each dataset, illustrating the distribution of inliers and outliers. In the graph, inliers are represented by yellow bars, while outliers are shown in orange bars. It underscores the critical issue of outliers and their substantial impact on model performance, which this study aims to address. Therefore, this study strongly aims to address outliers and their impact on model performance.

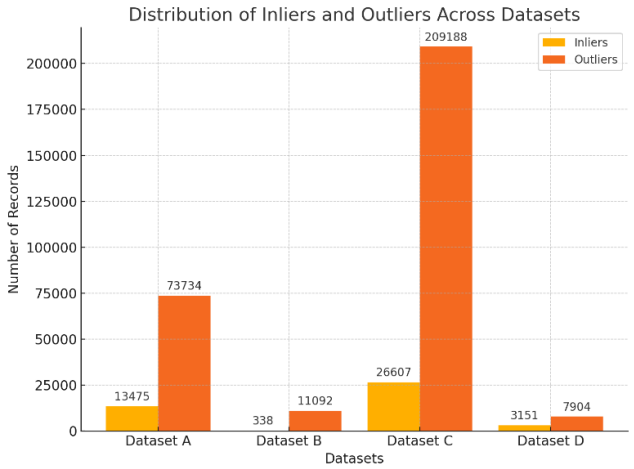

Figure 2. Outlier sensitivity for across the dataset

Data preparation in this study involves several steps. First, duplicate entries are removed to prevent distortion caused by overrepresented data points. Next, missing values are addressed to minimize bias and maintain the model's accuracy. Once the data is cleaned, feature normalization is applied using the Min-Max Scaler to rescale all feature values into a consistent range, typically between 0 and 1 [43]. The formula for Min-Max normalization is as follows: Eq. (1).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where, X is the original feature value, and X_{min} and X_{max} are the minimum and maximum values of the feature, respectively. This formula ensures that all feature values are scaled proportionally within the defined range and prevents features with larger numeric ranges from dominating the learning process. For example, if the URL length has a minimum value of 11 a maximum of 57, and an observed value of 48, the normalized value would be:

$$X_{norm} = \frac{48-11}{57-11} \approx 0.804 \tag{2}$$

Normalization scales all feature values to a non-negative range and standardizes their scale without removing outliers. This ensures equal contribution from small-range features like URL length and large-range features like timestamps to the learning process [43]. This process enhances the model's stability and ensures fairness in feature contribution during learning.

3.3 Feature selection and optimization process

The feature selection and optimization process begin with Correlation Thresholding to identify and remove redundant features. This step reduces multicollinearity by analyzing the linear relationships between features and eliminating those with high correlation. Next, Grid Search optimizes the regularization parameter λ in the LASSO model. LASSO applies this optimal λ to shrink less important feature coefficients to zero, retaining only the most relevant features. This process improves both model performance and computational efficiency.

3.3.1 Pearson correlation

The feature selection process begins with Pearson correlation, which quantifies the linear relationship between pairs of features in the dataset. The Pearson correlation coefficient r_{ij} ranges from -1 to 1, where $r_{ij} = 1$ indicates a perfect positive relationship, $r_{ij} = -1$ indicates a perfect negative relationship, and $r_{ij} = 0$ indicates no relationship. The coefficient is calculated using the Eq. (3) [44].

$$r_{ij} = \frac{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j)}{\sqrt{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)^2}\sqrt{\sum_{k=1}^{n}(X_{jk} - \bar{X}_j)^2}} \tag{3}$$

where, X_{ik} and X_{jk} are the values of features X_i and X_j for the k-th observation, $\bar{X}_i$ and $\bar{X}_j$ are their mean values, and n is the number of observations.

A correlation matrix is computed for all feature pairs to identify and remove redundant features. Threshold values (τ) are tested to find the optimal value that maximizes model performance. Table 2 outlines detailed steps for this process.

3.3.2 Correlation thresholding

Correlation Thresholding is applied after calculating the correlation matrix to identify and remove highly correlated features. Instead of using a fixed threshold, such as 0.9, this study optimizes the τ based on model performance. Threshold values ranging from 0.5 to 1.0 are tested to find the optimal τ. For feature pairs with a correlation ($|r_{ij}| > \tau$), one feature is removed.

The decision on which feature to remove is based on three factors [45]. First, domain knowledge ensures the retention of features relevant to phishing detection. Second, feature importance scores, derived from preliminary analyses, prioritize features with higher contributions to model performance. Third, multicollinearity is reduced by carefully selecting and removing highly correlated features. Detailed steps for this process are outlined in Table 2.

Table 2. Pseudocode for correlation thresholding

Step	Description
Input	Dataset D with N samples and M features, target variable Y
Output	Reduced set of features F after correlation thresholding
Calculate Correlation Matrix	Compute the correlation matrix R for all features in the dataset D
Set Threshold τ	Define an initial range of thresholds τ to evaluate.
Optimize Threshold	For each threshold τ in the defined range, assess the impact on model performance and choose the τ that maximizes the model's accuracy.
Identify Highly Correlated Pairs	Identify pairs of features (X_i, X_j) with correlation coefficient $\lvert r_{ij} \rvert$ greater than the optimized threshold τ.
Remove Redundant Features	Remove one feature from each highly correlated pair based on domain knowledge, feature importance scores, or reduce multicollinearity.
Return Reduced Feature Set	After removing redundant features, return the reduced set of features F.

3.3.3 Grid search for lasso optimization

Grid Search is used to optimize the λ in the LASSO model. The parameter λ determines the strength of the penalty applied to feature coefficients, where larger values shrink more coefficients to zero. This simplifies the model by removing less important features while retaining the most relevant ones [46].

Table 3. Pseudocode for grid search

Step	Description
Input	Dataset D with N samples and M features, target variable Y
Output	Optimal λ for LASSO
Define and evaluate λ	Set up a range of λ values, evaluate them based on model performance.
Perform Cross-Validation	For each λ value, evaluate model performance using cross-validation to ensure robustness.
Select Optimal λ	Choose the λ that results in the best cross-validation score, balancing model simplicity and accuracy.
Return Optimal λ	After evaluation, return the optimal λ value.

To find the optimal λ, Grid Search tests a predefined range of values. Each λ is evaluated using cross-validation to measure its impact on model performance. The value that achieves the best balance between accuracy and simplicity is selected as the optimal λ. The detailed steps of this process are outlined in Table 3.

3.3.4 LASSO feature selection

After determining the optimal λ through Grid Search, LASSO applies this parameter to penalize less important features in the model. The LASSO objective function is defined in Eq. (4) [46],

$$\hat{\beta}_{lasso} = argmin_{\beta} \left\{ \frac{1}{2N} \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{p} r_{ij} \beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \quad (4)$$

This objective function minimizes the sum of squared errors and applies an L_1 penalty proportional to the absolute values of the coefficients (β_j). The penalty term, controlled by λ, ensures that coefficients for less important features are shrunk to zero, effectively removing them from the model. The terms are defined as follows:

The first term represents the least squares error. This term calculates the sum of squared differences between observed and predicted outcomes, see Eq. (5).

$$\frac{1}{2N} \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{p} r_{ij} \beta_j\right)^2 \quad (5)$$

where, N is the number of observations, X_{ij} is the value of the j-th feature for the i-th observation, β_j is the coefficient associated with the j-th feature, and β_0 is the intercept term.

The second term is the L_1 regularization term. This term adds a penalty proportional to the sum of the absolute values of the coefficients β_j, see Eq. (6).

$$\lambda \sum_{j=1}^{p} |\beta_j| \quad (6)$$

The regularization parameter λ controls the strength of this penalty. A larger λ increases the penalty, shrinking more coefficients to zero and removing the corresponding features from the model. Conversely, a smaller λ retains more features with non-zero coefficients.

The steps for applying for LASSO in feature selection are outlined in Table 4.

Table 4. Pseudocode for LASSO feature selection

Step	Description
Input	Dataset D with N samples and M features, target variable Y
Output	Selected features from the LASSO model
Initialize LASSO Model	Define the LASSO model with the regularization parameter λ.
Fit Model	Fit the LASSO model to the dataset D by minimizing the objective function.
Extract Coefficients	Extract the coefficients β_j for each feature j after fitting the model.
Feature Selection	Identify features where $\beta_j \neq 0$ as these are the features selected by the model.
Return Selected Features	Return the set of selected features with non-zero coefficients.

Optimizing LASSO can be computationally expensive due to iterative Grid Search. This study simplifies the process to reduce execution time while maintaining model accuracy. The optimization ensures efficient and effective feature selection without sacrificing performance.

3.4 Random Forest

This study uses RF combined with WBS based on uncertainty to improve the performance and efficiency of phishing detection in complex datasets. It starts with LASSO$^+$ feature selection, which identifies the most relevant features for classification. Then, these selected features are used in the RF model. These features are further evaluated and chosen during the building of decision trees.

3.5 Random Forest construction using Weighted Bootstrap Sampling with uncertainty

During the construction of each decision tree T_b in the forest, the dataset is resampled using WBS based on uncertainty sampling approach discussed by Liu and Li [18]. Instead of standard Bootstrap Sampling, WBS assigns a weight W_i to each sample based on the uncertainty of its predicted outcome. Those with higher uncertainty receive greater weights, so that these samples could be selected during the resampling process.

The weight W_i for each sample is calculated using the following formula:

$$W_i = \frac{1}{|P(\hat{Y}_i=1|X_i)-0.5|+\epsilon} \quad (7)$$

where, $P(\hat{Y}_i = 1|X_i)$ is the predicted probability that the sample X_i belongs to the positive class, such as predicting that a website is phishing. The expression $|P(\hat{Y}_i = 1|X_i) - 0.5|$ calculates the absolute difference between the predicted probability and 0.5. This difference shows how unsure the model is about the prediction. If the value is close to 0.5, it means the model is very uncertain. The small constant ϵ is added to avoid dividing by zero.

Once the weights are calculated for all samples, these weights are normalized so that the sum of all weights equals one. This normalization step ensures that the weights can be interpreted as probabilities when selecting samples. The probability of selecting a sample X_i to be included in the new Bootstrap sample D_b is proportional to its weight W_i. This probability is given by:

$$P(X_i \in D_b) = W_i \quad (8)$$

Using these weights, the new Bootstrap sample D_b is created by randomly selecting N observations from the dataset. The probability of each observation being selected depends on its weight W_i. Hence, samples with higher uncertainty are more likely to be included in the training set. This approach helps the model to focus on more challenging cases, ultimately improving its overall performance.

In RF model, each decision tree is constructed by selecting a random subset of features at each node. Instead of using all available features in the dataset, the model selects a smaller random subset of features to consider when splitting at each node. The number of features selected at each node is denoted by m, while M represents the total number of features available in the dataset. The model then examines the selected features m to find the one that best divides the data at that node. To determine how many features m should be selected at each node, the commonly used formula:

$$m = \sqrt{M} \quad (9)$$

The final prediction is determined by combining all these predictions after each decision tree in the forest makes its prediction for a sample X. The following explanation shows how the sample X is being predicted. The prediction made by the b-th tree is denoted as $T_b(X)$. The final predicted class for the sample is represented by $\hat{Y}$.

In a classification model, these predictions are combined using majority voting. This means the class often predicted by the trees becomes the final prediction. The equation is:

$$\hat{Y} = mode\{T_b(X)\}_{b=1}^{B} \quad (10)$$

In Eq. (10), B stands for the total number of trees in the forest. The mode function selects the class that appears most frequently among the predictions from all B trees. The RF formula is widely recognized and applied in [47].

3.6 Model training and evaluation

After resampling the data using WBS, the RF model is trained on the newly constructed dataset to enhance its ability to detect phishing websites in imbalanced datasets. The model's performance is evaluated using accuracy, precision, recall, f1 score, AUC, and execution time.

4. RESULT AND DISCUSSION

This section presents the experimental results and discusses the findings of this study. The analysis demonstrates the impact of the LASSO$^+$ feature selection method compared to the traditional LASSO approach in phishing detection. Performance metrics for the proposed method are evaluated across multiple public datasets and compared to state-of-the-art techniques. The section is organized into several subsections, starting with the parameter tuning of LASSO$^+$ to determine the optimal values for key hyperparameters. This is followed by an ablation study to assess the individual contributions of each component of the proposed model. Finally, a comparative analysis is conducted to evaluate the performance of the proposed method concerning existing approaches in the literature.

4.1 Parameter tunning of LASSO$^+$

The optimal lambda values are determined through Grid Search. For Dataset A, the optimal lambda value of 0.0010 indicates a higher regularization level, suggesting the dataset may contain more noise or irrelevant features. In contrast, the lower lambda values of 0.0001 for Datasets B, C, and D imply less regularization is needed, indicating that these datasets have more informative features and require minimal shrinkage. This adjustment ensures that the most pertinent features are retained, enhancing model accuracy and performance.

4.2 Ablation study

The ablation study evaluates the performance of traditional LASSO and LASSO$^+$ across four datasets. Table 5 shows that LASSO$^+$ consistently achieves better accuracy, precision, and F1-score results with comparable AUC. In Dataset A, LASSO+ improves accuracy from 89.12% to 90.58% and F1-score from 86.06% to 87.68%, though execution time increases from 0.48s to 31.72s. In Dataset B, accuracy increases from 92.65% to 96.93%, and precision improves from 91.96% to 97.48%, with recall adjusted from 93.26% to 96.27% and execution time reaching 21.47s. Dataset C maintains perfect recall, with LASSO$^+$ improving accuracy from 99.85% to 99.88% and execution time rising from 0.44s to 63.95s. Dataset D demonstrates the improvements in accuracy (84.89% to 92.17%) and recall (84.89% to 94.98%), with execution time increasing from 0.04s to 2.29s.

Table 5. Comparison performance LASSO and LASSO$^+$

Dataset	Feature Selection Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
A	LASSO	89.12	77.12	97.33	86.06	96.96	0.48
	LASSO$^+$	90.58	80.53	96.21	87.68	97.62	31.72
B	LASSO	92.65	91.96	93.26	92.61	97.78	0.08
	LASSO$^+$	96.93	97.48	96.27	92.61	97.78	21.47
C	LASSO	99.85	99.74	100.00	99.87	99.99	0.44
	LASSO$^+$	99.88	99.80	100.00	99.90	99.99	63.95
D	LASSO	84.89	87.95	84.89	84.89	97.37	0.04
	LASSO$^+$	92.17	91.55	94.98	93.23	97.39	2.29

Table 6. Comparison of phishing detection performance in Dataset A

Publications	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
Wei and Sekiya	96.94	95.55	95.55	-	99.50	15.43
Othman and Hassan	98.69	98.58	98.80	98.69	-	-
Kalabarige et al.	98.43	97.93	98.96	98.44	-	-
Proposed Method	99.20	99.53	99.59	99.56	99.92	6.80

Table 7. Comparison of phishing detection performance in Dataset B

Publications	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
Moedjahedy et al.	97.96	-	-	-	-	-
Kumar et al.	99.70	95.70	98.10	-	-	-
Adane et al.	97.90	97.63	98.14	97.88	-	10.00
Trad and Chehab	97.30	97.78	96.80	97.29	99.56	-
Proposed Method	98.85	98.97	98.76	98.81	99.90	2.11

Table 8. Comparison of phishing detection performance in Dataset C

Publications	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
Vajrobol et al.	99.97	99.97	99.97	99.97	-	-
Proposed Method	100.00	100.00	100.00	100.00	100.00	18.22

Table 9. Comparison of phishing detection performance in Dataset D

Publications		Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
Adane et al.		97.37	96.65	98.18	97.40	-	12
Shabudin et al.	FSOR	97.08	-	-	-	-	10
	FSFM	95.19	-	-	-	-	6
Taha et al.		97.00	98.00	97.00	97.00	-	-
Toğaçar		97.26	96.35	97.28	96.91	-	-
Ubing et al.		95.40	93.50	95.90	-	-	-
Proposed Method		98.69	98.69	98.99	98.84	99.90	1.09

Table 10. Comparison of baseline methods and proposed method across all datasets

Dataset	Baseline Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
A	Random Forest	97.02	95.21	96.22	95.71	99.49	15.87
	Random Forest + WBS	95.74	92.88	94.94	93.90	99.15	23.14
	LASSO$^+$ + Random Forest	97.15	95.34	96.45	95.89	99.48	16.43
	Proposed Method	99.20	99.53	99.59	99.56	99.92	6.80
B	Random Forest	96.85	97.39	96.19	96.79	99.43	1.09
	Random Forest + WBS	97.37	97.59	97.07	97.33	99.52	2.15
	LASSO$^+$ + Random Forest	96.06	96.17	95.83	96.00	99.15	0.84
	Proposed Method	98.85	98.97	98.76	98.81	99.90	2.11
C	Random Forest	100.00	100.00	100.00	100.00	100.00	33.69
	Random Forest + WBS	100.00	100.00	100.00	100.00	100.00	67.84
	LASSO$^+$ + Random Forest	99.99	99.98	100.00	99.99	100.00	12.94
	Proposed Method	100.00	100.00	100.00	100.00	100.00	18.22
D	Random Forest	96.78	96.17	98.24	97.20	99.41	1.18
	Random Forest + WBS	96.06	96.06	96.06	96.06	99.25	2.57
	LASSO$^+$ + Random Forest	96.38	96.38	96.38	96.37	99.41	1.2732
	Proposed Method	98.69	98.69	98.99	98.84	99.90	1.09

Table 11. Statistical comparison of accuracy across datasets

Dataset	Baseline Methods	Mean Difference	t-Statistic	p-Value
A	Random Forest	2.18	92.00	1.50×10^{-64}
	Random Forest + WBS	6.65	286.28	4.78×10^{-89}
	LASSO^{+} + Random Forest	3.14	107.33	4.89×10^{-67}
B	Random Forest	2.00	83.24	6.14×10^{-62}
	Random Forest + WBS	1.38	55.26	1.40×10^{-51}
	LASSO^{+} + Random Forest	2.93	121.68	1.63×10^{-71}
C	Random Forest	0.00	-0.33	7.42×10^{-1}
	Random Forest + WBS	0.00	2.56	1.33×10^{-2}
	LASSO^{+} + Random Forest	0.00	0.80	4.30×10^{-1}
D	Random Forest	1.91	63.40	8.51×10^{-55}
	Random Forest + WBS	2.63	110.82	6.47×10^{-69}
	LASSO^{+} + Random Forest	2.61	110.42	1.96×10^{-66}

LASSO^{+} demonstrates consistent advantages in predictive performance, particularly in accuracy and recall, while maintaining competitive overall metrics. Despite higher computational costs in some cases, its reliability and effectiveness make it a strong candidate for feature selection in phishing detection.

4.3 Comparison analysis

This section compares the proposed model with previous studies across multiple datasets, as shown in Tables 6 to 9. The results demonstrate that the proposed model consistently achieves higher recall while maintaining strong performance in accuracy, recall, F1-score, and AUC. Additionally, the model significantly reduces execution time compared to other approaches.

For Dataset A, the proposed model achieves the highest recall, outperforming studies by Wei and Sekiya [36], Othman and Hassan [31], and Kalabarige et al. [37], while also demonstrating the shortest execution time. In Dataset B, the model demonstrates superior recall compared to methods by Moedjahedy et al. [10], Pandey et al. [40], Adane et al. [38], and Trad and Chehab [39]. While Pandey et al. [40] achieved slightly higher accuracy, the proposed model demonstrates superior precision and recall.

In Dataset C, the proposed model matches the precision of Vajrobol et al. [41] while providing competitive execution time. For Dataset D, the model surpasses Adane et al. [38], Shabudin et al. [42], Taha et al. [48], Toğaçar [21], and Ubing et al. [49] in all aspects while achieving the fastest execution time. These findings highlight the proposed model's ability to focus on accurate detection, ensuring minimal false positives, which is crucial for phishing prevention systems.

4.4 Baseline comparison

The proposed method demonstrates superior performance compared to baseline approaches across key metrics, as summarized in Table 10. It achieves higher precision and recall, essential for phishing detection while maintaining faster execution times, particularly for smaller datasets. The proposed method effectively handles scalability on larger datasets, delivering consistent and robust performance. Among the baseline methods, RF with WBS shows competitive results but suffers from longer execution times due to increased computational complexity. In contrast, the proposed method strikes an optimal balance between performance and efficiency.

The statistical analysis in Table 11 further emphasizes the superiority of the proposed method. The method achieves significant accuracy improvements for Datasets A and B, with t-statistics exceeding 80 and p-values nearing zero. These results reflect their robust feature selection and classification capabilities. In Dataset C, the simplicity and homogeneity of the dataset result in minimal differences between methods, as indicated by near-zero mean differences and high p-values. However, the slight negative t-statistic further underscores the stability of all approaches under uniform conditions. Dataset D, representing a moderately complex scenario, highlights the proposed method's adaptability, with t-statistics above 60 and notable p-values demonstrating its scalability.

Across all datasets, the proposed method consistently outperforms baseline approaches in precision, recall, and execution time. These results underline its effectiveness in balancing accuracy and efficiency.

4.5 Performance summary across datasets

Table 12 demonstrates that the characteristics of the datasets and the number of outliers impact the proposed method's performance. Dataset A has the highest number of outliers and substantial variability. It achieves consistent performance and highlights the method's effectiveness in managing complex datasets. Dataset C contains a moderate number of outliers. It performs perfectly and illustrates the model's robustness in scenarios with balanced data.

Dataset D is characterized by a moderate to high number of outliers and a smaller size. It achieves good performance but is slightly outperformed by Dataset C. This indicates that dataset size also contributes to performance variations. Dataset B has the lowest number of outliers and a compact structure. It records the fastest execution time while maintaining high performance. These findings indicate that although the number of outliers influences performance, the proposed method is highly adaptable. It consistently delivers optimal results across diverse dataset conditions.

Table 12. Overall performance metrics of the proposed method across all datasets

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Execution Time (s)
A	99.20	99.53	99.59	99.56	99.92	6.80
B	98.85	98.97	98.76	98.81	99.90	2.11
C	100.00	100.00	100.00	100.00	100.00	18.22
D	98.69	98.69	98.99	98.84	99.90	1.09

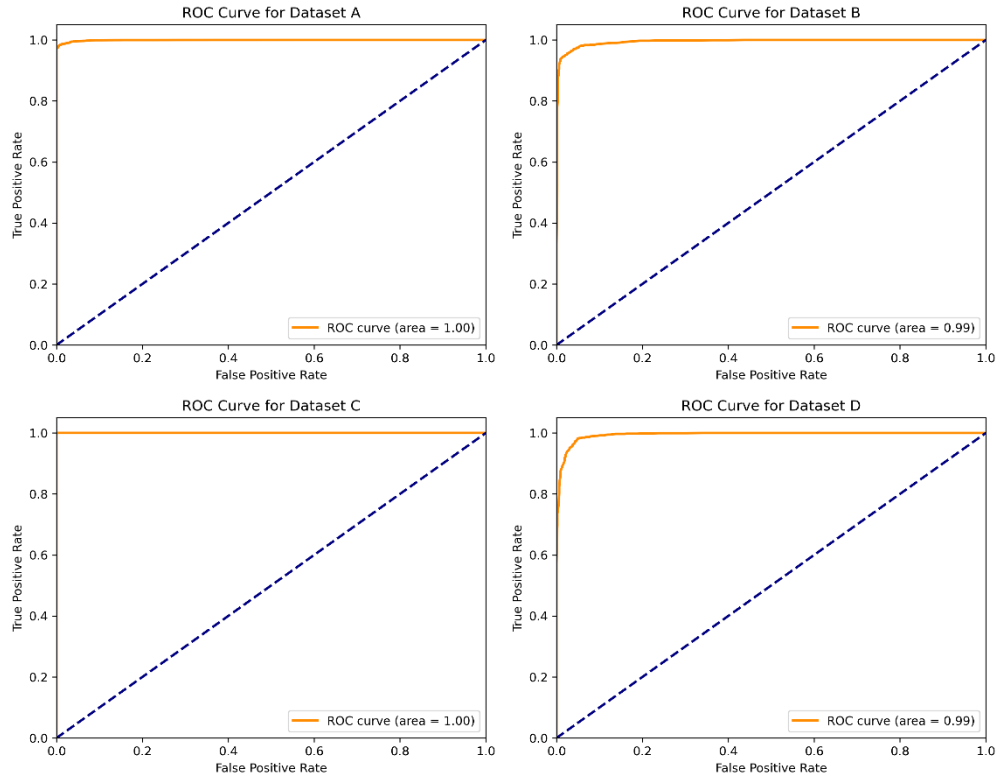

Figure 3. ROC curves for all datasets

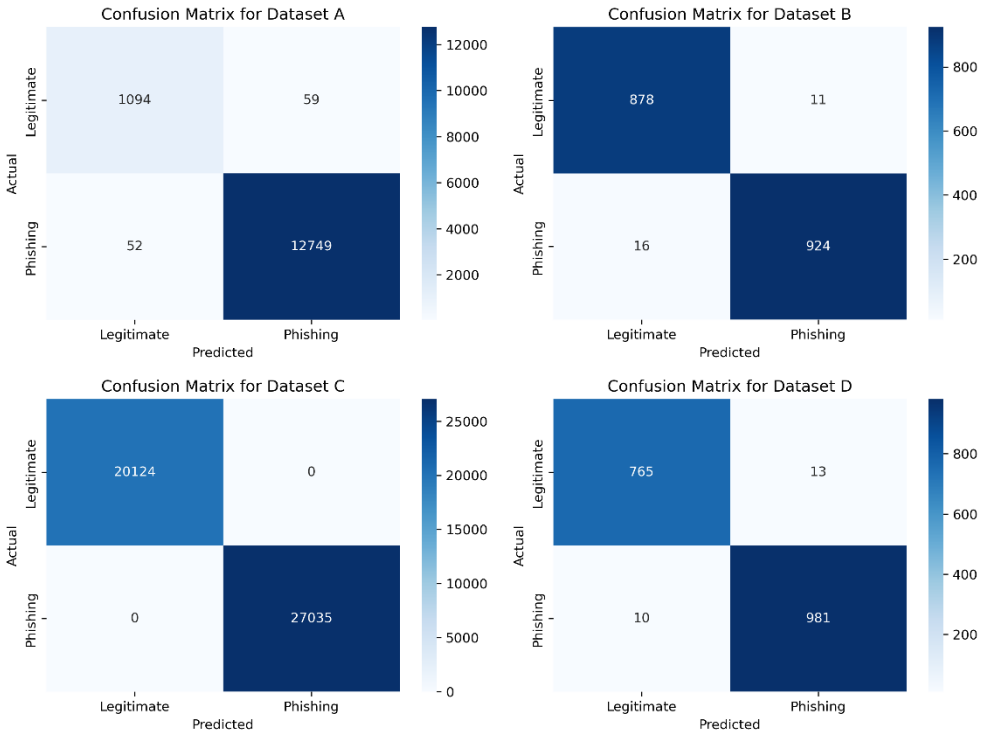

Figure 4. Confusion matrices for all datasets

The ROC curve illustrates the relationship between True Positive Rate (TPR) and False Positive Rate (FPR). TPR reflects the model's ability to correctly identify phishing instances, while FPR indicates the rate of legitimate cases misclassified as phishing. High TPR and low FPR demonstrate the model's effectiveness in distinguishing phishing from legitimate cases.

For Dataset A, as shown in Figure 3, the ROC curve indicates a high TPR with a low FPR. This is supported by the confusion matrix, which reveals strong performance despite the highest number of outliers, as illustrated in Figure 4. For Dataset B, the ROC curve and confusion matrix collectively demonstrate the model's ability to maintain balanced performance in smaller datasets with fewer outliers.

Dataset C, the ROC curve, and the confusion matrix highlight perfect performance, achieving a TPR of 1 across all

FPR values. Finally, Dataset D shows a similar trend to Dataset B, achieving high TPR and low FPR despite moderate outliers, as reflected in the confusion matrix.

5. CONCLUSIONS

This study addresses critical challenges in phishing detection, particularly the trade-off between model performance and execution time when handling datasets with a high proportion of outliers. By integrating the LASSO$^+$ feature selection method with Random Forest and WBS based on uncertainty, the proposed approach balances high recall and reduced execution time. The LASSO$^+$ method improves feature selection by incorporating correlation thresholding and optimizing parameters through Grid Search. WBS enhances the model's ability to handle datasets with a high presence of outliers and ensures consistent performance across different evaluation metrics.

The findings reveal minimal variability in some datasets and result in uniformly high performance. While this consistency highlights the proposed method's robustness, it may limit further exploration of statistical variability in specific scenarios. Despite this, the proposed method demonstrates robust results across all tested scenarios and highlights its practical applicability for phishing detection.

The primary limitation of this study lies in its focus on phishing detection using URL features characterized by outliers. While the proposed method effectively handles such datasets, its applicability to other types of cyberattacks remains uncertain. Cybersecurity threats such as malware distribution, typosquatting, and malicious redirects may exhibit different patterns of outliers or lack significant anomalies. This poses challenges for the current approach. Furthermore, the model has not been tested on datasets with normal distributions or minimal outlier presence, which limits its generalizability across diverse contexts.

Future research should explore the application of the proposed method to other cyberattacks with unique outlier patterns and assess its adaptability to datasets with normal distributions. These directions aim to enhance the method's robustness and broaden its applicability beyond phishing detection.

REFERENCES

[1] APWG. (2023). Phishing Activity Trends Report.

[2] Setiadi, D.R.I.M., Widiono, S., Safriandono, A.N., Budi, S. (2024). Phishing website detection using bidirectional gated recurrent unit model and feature selection. Journal of Future Artificial Intelligence Technology, 2(1): 75-83. https://doi.org/10.62411/faith.2024-15

[3] Waseso, B.M.P., Setiyanto, N.A. (2023). Web phishing classification using combined machine learning methods. Journal of Computer Theory and Applications, 1(1): 11-18. https://doi.org/10.33633/jcta.v1i1.8898

[4] Dhahir, Z.S. (2024). A hybrid approach for efficient DDoS detection in network traffic using CBLOF-based feature engineering and XGBoost. Journal of Future Artificial Intelligence Technology, 1(2): 174-190. https://doi.org/10.62411/faith.2024-33

[5] Tharani, J.S., Arachchilage, N.A.G. (2020). Understanding phishers' strategies of mimicking uniform resource locators to leverage phishing attacks: A machine learning approach. Security and Privacy. 3(5): e120. https://doi.org/10.1002/spy2.120

[6] Kandula, L.R.R., Lakshmi, T.J., Alla, K., Chivukula, R. (2022). An intelligent prediction of phishing URLs using ML algorithms. International Journal of Safety and Security Engineering, 12(3): 381-386. https://doi.org/10.18280/ijsse.120312

[7] Kothamasu, G.A., Venkata, S.K.A., Pemmasani, Y., Mathi, S. (2023). An investigation on vulnerability analysis of phishing attacks and countermeasures. International Journal of Safety and Security Engineering, 13(2): 333-340. https://doi.org/10.18280/ijsse.130215

[8] Hannousse, A., Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. Engineering Applications of Artificial Intelligence, 104: 104347. https://doi.org/10.1016/j.engappai.2021.104347

[9] Guptta, S.D., Shahriar, K.T., Alqahtani, H., Alsalman, D., Sarker, I.H. (2024). Modeling hybrid feature-based phishing websites detection using machine learning techniques. Annals of Data Science, 11(1): 217-242. https://doi.org/10.1007/s40745-022-00379-8

[10] Moedjahedy, J., Setyanto, A., Alarfaj, F.K., Alreshoodi, M. (2022). CCrFS: Combine correlation features selection for detecting phishing websites using machine learning. Future Internet, 14(8): 229. https://doi.org/10.3390/fi14080229

[11] Pudjihartono, N., Fadason, T., Kempa-Liehr, A.W., O'Sullivan, J.M. (2022). A review of feature selection methods for machine learning-based disease risk prediction. Frontiers in Bioinformatics, 2: 1-17. https://doi.org/10.3389/fbinf.2022.927312

[12] Emmert-Streib, F., Dehmer, M. (2019). High-dimensional LASSO-Based computational regression models: regularization, shrinkage, and selection. Machine Learning and Knowledge Extraction, 1(1): 359-383. https://doi.org/10.3390/make1010021

[13] Htwe, C.S., Myint, Z.T.T., Thant, Y.M. (2024). IoT security using machine learning methods with features correlation. Journal of Computer Theories and Applications, 2(2): 151-163. https://doi.org/10.62411/jcta.11179

[14] Butnaru, A., Mylonas, A., Pitropakis, N. (2021). Towards lightweight URL-based phishing detection. Future Internet, 13(6): 154. https://doi.org/10.3390/fi13060154

[15] Divakaran, D.M., Oest, A. (2022). Phishing detection leveraging machine learning and deep learning: A review. IEEE Security & Privacy, 20(5): 86-95. https://doi.org/10.1109/MSEC.2022.3175225

[16] Uzun Ozsahin, D., Mustapha, M.T., Mubarak, A.S., Ameen, Z.S., Uzun, B. (2022). Impact of outliers and dimensionality reduction on the performance of predictive models for medical disease diagnosis. International Conference on Artificial Intelligence in Everything (AIE), pp. 79-86. https://doi.org/10.1109/AIE57029.2022.00023

[17] Demir, S., Sahin, E.K. (2023). Application of state-of-the-art machine learning algorithms for slope stability prediction by handling outliers of the dataset. Earth Science Informatics, 16(3): 2497-2509. https://doi.org/10.1007/s12145-023-01059-8

[18] Liu, S., Li, X. (2023). Understanding Uncertainty Sampling. arXiv:2307.02719. https://doi.org/10.48550/arXiv.2307.02719
[19] Sarasjati, W., Rustad, S., Santoso, H.A., Syukur, A., Rafrastara, F.A. (2022, September). Comparative study of classification algorithms for website phishing detection on multiple datasets. In 2022 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, pp. 448-452. https://doi.org/10.1109/iSemantic55962.2022.9920475
[20] Le, H.L., Le, T.T., Vu, T.T.H., Tran, D.H., Chau, D.V., Ngo, T.T.T. (2023). A survey on the impact of hyperparameters on random forest performance using multiple accelerometer datasets. International Journal of Computers and their Applications, 30(4): 351-361.
[21] Toğaçar, M. (2021). Detection of phishing attacks on websites with lasso regression, minimum redundancy maximum relevance method, machine learning methods, and deep learning model. Turkish Journal of Science and Technology, 16(2): 231-243.
[22] Kim, D.S. (2020). A correlation thresholding algorithm for learning factor analysis models. Ph.D. dissertation. University of California.
[23] Lam, C. (2021). Rank determination for time series tensor factor model using correlation thresholding. LSE, London, UK, Working Paper.
[24] Klosa, J., Simon, N., Westermark, P.O., Liebscher, V., Wittenburg, D. (2020). Seagull: Lasso, group lasso and sparse-group lasso regularization for linear regression models via proximal gradient descent. BMC Bioinformatics, 21(1): 407. https://doi.org/10.1186/s12859-020-03725-w
[25] Bertrand, Q., Klopfenstein, Q., Blondel, M., Vaiter, S., Gramfort, A., Salmon, J. (2020). Implicit differentiation of lasso-type models for hyperparameter optimization. 37th International Conference on Machine Learning, PartF16814: 787-798.
[26] Al-Qurashi, R., AlEroud, A., Saifan, A.A., Alsmadi, M., Alsmadi, I. (2021). Generating optimal attack paths in generative adversarial phishing. IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1-6. https://doi.org/10.1109/ISI53945.2021.9624751
[27] Akazue, M.I., Debekeme, I.A., Edje, A.E., Asuai, C., Osame, U.J. (2023). Unmasking fraudsters: Ensemble features selection to enhance random forest fraud detection. Journal of Computer Theory and Applications, 1(2): 201-211. https://doi.org/10.33633/jcta.v1i2.9462
[28] Okpor, M.D., et al. (2024). Pilot study on enhanced detection of cues over malicious sites using data balancing on the random forest ensemble. Journal of Future Artificial Intelligence Technology, 1(2): 109-123. https://doi.org/10.62411/faith.2024-14
[29] Almseidin, M., Abu Zuraiq, A., Al-kasassbeh, M., Alnidami, N. (2019). Phishing detection based on machine learning and feature selection methods. International Journal of Interactive Mobile Technologies, 13(12): 171. https://doi.org/10.3991/ijim.v13i12.11411
[30] Al-Sarem, M., Saeed, F., Al-Mekhlafi, Z.G., Mohammed, B.A., Al-Hadhrami, T., Alshammari, M.T., Alshammari, T.S. (2021). An optimized stacking ensemble model for phishing websites detection. Electronics, 10(11): 1285. https://doi.org/10.3390/electronics10111285
[31] Othman, M., Hassan, H. (2022). An empirical study towards an automatic phishing attack detection using ensemble stacking model. Future Computing and Informatics Journal, 7(1): 1-12. https://doi.org/10.54623/fue.fcij.7.1.1
[32] Yang, H., Lim, H., Moon, H., Li, Q., Nam, S., Kim, J., Choi, H.T. (2022). Simple optimal sampling algorithm to strengthen digital soil mapping using the spatial distribution of machine learning predictive uncertainty: A case study for field capacity prediction. Land, 11(11): 2098. https://doi.org/10.3390/land11112098
[33] Vrbančič, G. (2020). Datasets for phishing websites detection. Data in Brief, 33. https://doi.org/10.1016/j.dib.2020.106438
[34] 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
[35] Mohammad, R.M., Thabtah, F., Mccluskey, L. (2013). Phishing Websites Features. IEEE, pp. 1-7.
[36] Wei, Y., Sekiya, Y. (2022). Sufficiency of ensemble machine learning methods for phishing websites detection. IEEE Access, 10: 124103-124113. https://doi.org/10.1109/ACCESS.2022.3224781
[37] Kalabarige, L.R., Rao, R.S., Abraham, A., Gabralla, L.A. (2022). Multilayer stacked ensemble learning model to detect phishing websites. IEEE Access, 10: 79543-79552. https://doi.org/10.1109/ACCESS.2022.3194672
[38] Adane, K., Beyene, B., Abebe, M. (2023). Single and hybrid-ensemble learning-based phishing website detection: examining impacts of varied nature datasets and informative feature selection technique. Digital Threats: Research and Practice, 4(3): 1-27. https://doi.org/10.1145/3611392
[39] Trad, F., Chehab, A. (2024). Prompt engineering or fine-tuning? a case study on phishing detection with large language models. Machine Learning and Knowledge Extraction, 6(1): 367-384. https://doi.org/10.3390/make6010018
[40] Pandey, M.K., Singh, M.K., Pal, S., Tiwari, B.B. (2022). Prediction of phishing websites using stacked ensemble method and hybrid features selection method. SN Computer Science, 3(6): 488. https://doi.org/10.1007/s42979-022-01387-4
[41] Vajrobol, V., Gupta, B.B., Gaurav, A. (2024). Mutual information based logistic regression for phishing URL detection. Cyber Security and Applications, 2: 100044. https://doi.org/10.1016/j.csa.2024.100044
[42] Shabudin, S., Sani, N.S., Ariffin, K.A.Z., Aliff, M. (2020). Feature selection for phishing website classification. International Journal of Advanced Computer Science and Applications, 11(4): 587-595. https://doi.org/10.14569/IJACSA.2020.0110477
[43] Coste, C.I. (2023). Malicious web links detection - a comparative analysis of machine learning algorithms. Studia Universitatis Babeș-Bolyai Informatica, 68(1): 21-36. https://doi.org/10.24193/subbi.2023.1.02
[44] Jebarathinam, C., Home, D., Sinha, U. (2020). Pearson correlation coefficient as a measure for certifying and quantifying high-dimensional entanglement. Physical

Review A, 101(2): 022112. https://doi.org/10.1103/PhysRevA.101.022112
[45] Grady, S.K., Dojcsak, L., Harville, E.W., Wallace, M.E., Vilda, D., Donneyong, M.M., Langston, M.A. (2023). Seminar: Scalable preprocessing tools for exposomic data analysis. Environmental Health Perspectives, 131(12): 1-7. https://doi.org/10.1289/EHP12901
[46] Hastie, T., Tibshirani, R., Friedman, J. (2017). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer.
[47] Breiman, L. (2001). Random forests. Machine Learning, 45: 5-32. https://doi.org/10.1023/A:1010933404324
[48] Taha, M.A., Jabar, H.D.A., Mohammed, W.K. (2024). A machine learning algorithms for detecting phishing websites: A comparative study. Iraqi Journal for Computer Science and Mathematics, 5(3): 275-286. https://doi.org/10.52866/ijcsm.2024.05.03.015
[49] Ubing, A.A., Jasmi, S.K.B., Abdullah, A., Jhanjhi, N.Z., Supramaniam, M. (2019). Phishing website detection: An improved accuracy through feature selection and ensemble learning. International Journal of Advanced Computer Science and Applications, 10(1): 252-257. https://doi.org/10.14569/IJACSA.2019.0100133