Original Article | *Peer Reviewed* | *Open Access*

International Journal of Experimental Research and Review (IJERR)
© Copyright by International Academic Publishing House (IAPH)
ISSN: 2455-4855 (Online)
www.iaph.in

ISSN 2455-4855

An Approach for Efficient and Accurate Phishing Website Prediction Using Improved ML Classifier Performance for Feature Selection

Anjaneya Awasthi* and Noopur Goel

Department of Computer Applications, VBS Purvanchal University, India

E-mail/Orcid Id:

AA, anjaneyaawasthi@gmail.com, https://orcid.org/0000-0003-4033-936X; *NG,* noopurt11@gmail.com, https://orcid.org/0000-0003-3351-3761

Article History:

Received: 09th Nov., 2023
Accepted: 15th June, 2024
Published: 30th June, 2024

Keywords:

Computer viruses, cybersecurity, Phishing Website Prediction, Machine learning (ML)

How to cite this Article:
Anjaneya Awasthi and Noopur Goel (2024). An Approach for Efficient and Accurate Phishing Website Prediction Using Improved ML Classifier Performance for Feature Selection. *International Journal of Experimental Research and Review, 40(spl.)*, 73-89.
DOI:
https://doi.org/10.52756/ijerr.2024.v40spl.006

Abstract: The article discusses the use of machine learning (ML) to combat phishing websites, which are deceptive sites that mimic trusted entities to steal sensitive information. This is why the continued invention of methods of identifying and counteracting phishing threats is beneficial. Such attacks pose significant risks to the integrity of online security. To enhance the success rate and specificity of predicting phishing websites, this study proposes a new approach that utilizes machine learning algorithms. To enhance the methods mentioned above and achieve better results in classification and better prediction of customer behaviour, the main points exposed to further transformations are increasing classifier accuracy and selecting an optimal feature space. Traditional anti-phishing strategies like blacklisting and heuristic searches often have slow detection times and high false positive rates. The article introduces a novel feature selection method to extract highly correlated features from datasets, thereby enhancing classifier accuracy. Using six feature selection techniques on a phishing dataset, it evaluates eight classifiers, including SVM, Logistic Regression, Random Forest, and others. The study finds that the Random Forest classifier combined with the Chi-2 feature selection method significantly improves model accuracy, achieving up to 96.99%.

Introduction

Computer viruses and biological viruses share a fundamental similarity in their intent to spread and replicate, although they operate in distinct realms. Computer viruses are malicious software programs designed to infect and disrupt computer systems by copying themselves and modifying other programs. On the other hand, biological viruses invade living cells to reproduce and spread, causing harm to their hosts (Franjić, 2020). In the context of cybersecurity, anti-phishing software has been developed to combat phishing attacks (Srinivas et al., 2019), which are similar to viruses in how they propagate. However, these programs often fail to detect all types of phishing attacks, as these frequently involve deceptive web pages rather than executable programs, exploiting vulnerabilities to steal sensitive data. Phishing attacks typically initiate through digital interactions such as emails or social media messages. Like biological viruses that activate through interaction with the host (Korkmaz et al., 2020), phishing attacks use these communications as a medium to spread. The strategy involves deceiving victims into providing personal information like credit card details and passwords (Oest et al., 2018). Over time, phishing techniques evolve to avoid detection (Gupta et al., 2022), often by mimicking legitimate websites and using credible URLs or email addresses to appear trustworthy. This evolution is akin to how biological viruses mutate to evade the immune response of their hosts. The severity of phishing attacks is underscored by a report from SlashNext, which noted a 61% increase in phishing incidents since 2021, identifying 255 million attacks in a six-month period across various digital platforms (Zhong and Sastry, 2017). This highlights the pervasive and escalating threat of phishing, demonstrating the

***Corresponding Author:** anjaneyaawasthi@gmail.com

substantial challenge it poses in both personal and organizational security contexts.

Since 2003, global agencies have been working together to mitigate the impact of phishing URLs,

Table 1. Features selection by different feature selection algorithms.

S. No.	Feature	Pearson	Chi-2	RFE	Logistics	Random Forest	LightGBM	Total
1	web_traffic	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
2	having_sub_domain	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
3	having_IP	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
4	URL_of_Anchor	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
5	SSLfinal_state	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
6	Links_in_tags	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
7	Google_Index	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	6
8	SFH	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	5
9	Prefix_Suffix	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	5
10	Shortining_Service	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	4
11	Request_URL	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	4
12	Redirect	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	4
13	Links-pointing_to_page	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	4
14	DNS_Record	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	4
15	having_At_symbol	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	3
16	age_of_domain	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	3
17	statistical_report	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	3
18	Page_Rank	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	3
19	Domain_registration_length	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	3
20	URL_Length	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	2
21	Abnormal_URL	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	2
22	Port	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	1
23	on_mouseover	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	1
24	Submitting_to_email	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	1
25	Iframe	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	1
26	HTTPS_token	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	1
27	popUpWindow	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0
28	double_slash_redirect	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0
29	Right_Click	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0
30	Favicon	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0

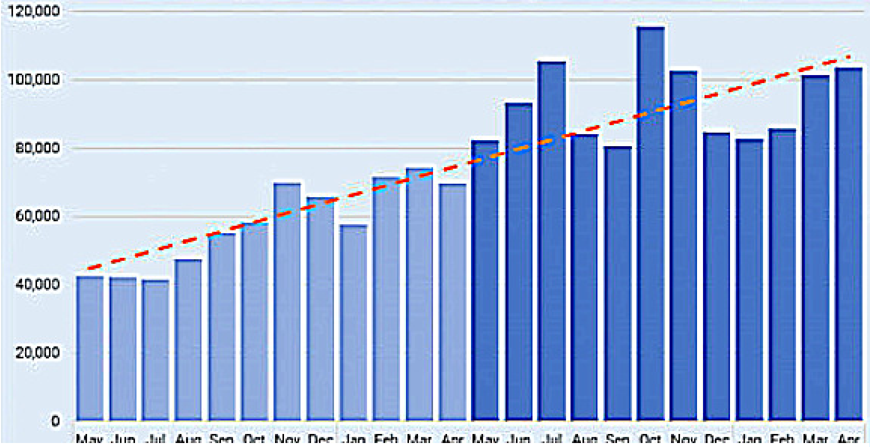


Figure 1. Phishing URL distribution between 2020 and 2022.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.006

emphasizing the need for literacy and education in combating such cyber threats (Awasthi and Goel, 2021a). Professionals and academic research play critical roles in these preventive measures. The aim of this study is to enhance the accuracy of phishing website detection (Awasthi and Goel, 2021b). This involves a two-phase process where the first phase uses eight machine learning classifiers to analyze the dataset. The second phase employs six feature selection algorithms along with machine learning classifiers to refine the data representation. The objective here is to identify key features that improve the accuracy of classifiers while using a reduced set of features. The study provides detailed tables listing the selected features of each algorithm, along with their descriptions and abbreviations, to illustrate the methodology and results systematically.

Table 2. Names of Features and their Abbreviations.

Feature Name	Abbreviation
web_traffic	f_1
having_sub_domain	f_2
having_IP	f_3
URL_of_Anchor	f_4
SSLfinal_state	f_5
Links_in_tags	f_6
Google_Index	f_7
SFH	f_8
Prefix_Suffix	f_9
Shortining_Service	f_10
Request_URL	f_11
Redirect	f_12
Links-pointing_to_page	f_13
DNS_Record	f_14
having_At_symbol	f_15
age_of_domain	f_16
statistical_report	f_17
Page_Rank	f_18
Domain_registration_length	f_19
URL_Length	f_20
Abnormal_URL	f_21
port	f_22
on_mouseover	f_23
Submitting_to_email	f_24
Iframe	f_25
HTTPS_token	f_26
popUpWindow	f_27
double_slash_redirect	f_28
Right_Click	f_29
Favicon	f_30

Table 3. Number of Features Selected by Feature Selection Algorithms.

S. No.	Pearson	Chi-2	RFE	Logistics	Random Forest	Light GBM
1	f_1	f_1	f_1	f_1	f_1	f_1
2	f_2	f_2	f_2	f_2	f_2	f_2
3	f_3	f_3	f_3	f_3	f_3	f_3
4	f_4	f_4	f_4	f_4	f_4	f_4
5	f_5	f_5	f_5	f_5	f_5	f_5
6	f_6	f_6	f_6	f_6	f_6	f_6
7	f_7	f_7	f_7	f_7	f_7	f_7
8	f_8	f_8	f_8	f_8	f_8	f_12
9	f_9	f_9	f_9	f_9	f_9	f_13
10	f_10	f_10	f_10	f_10	f_11	f_14
11	f_11	f_11	f_11	f_12	f_13	f_18
12	f_14	f_12	f_12	f_13	f_16	-
13	f_15	f_14	f_13	-	f_19	-
14	f_16	f_15	f_14	-	-	-
15	f_17	f_16	f_15	-	-	-
16	f_18	f_17	f_17	-	-	-
17	f_19	f_18	f_22	-	-	-
18	f_20	f_19	f_24	-	-	-
19	f_21	f_20	f_25	-	-	-
20	f_23	f_21	f_26	-	-	-
No. of features Selected	20	20	20	12	13	11

The article continues by reviewing previous methods for detecting phishing websites in Section 2. Section 3 provides a detailed overview of the experimental setup and its rationale. Information about the dataset and its attributes is presented in Section 4. The results of the experiment are analyzed in Section 5, while Sections 6 and 7 conclude the study and discuss its implications.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.006

Literature Review

This section explores cutting-edge machine-learning techniques for detecting phishing websites. Initially, early methods involved constructing basic feature sets from URL word lists using the bag-of-words approach (Le et al., 2011). Feng et al. (2018) introduced a more advanced method by proposing a novel neural network optimized for phishing detection through risk minimization principles, enhancing the model's ability to generalize across different scenarios. They tested their model on a substantial dataset from the UCI repository, consisting of 11,055 samples labeled as either legitimate or phishing and featuring 30 different attributes per website, encompassing domains, exceptions, HTML, JavaScript, and address bar elements. Further advancing the field, Muhammad et al. (2012) focused on the systematic extraction of URL features and the development of hierarchical classifiers. Their method emphasizes the automation of phishing detection, highlighting that although incorporating third-party service features may slow down the process, it significantly improves the accuracy of detection (Sahingoz et al., 2019). This approach underscores a pivotal shift towards more precise and automated methods in phishing website classification. The research examined the effectiveness of a proposed algorithm by testing it on 1,407 legitimate and 2,119 phishing websites from the Alexa database and PhishTank, respectively. This work highlighted the constraints of traditional rule-based feature selection and modeling, particularly in generalizing to previously unseen URLs, prompting a shift towards deep learning-based phishing detection (Iuga et al., 2016). Deep learning, known for its ability to model complex functions using large datasets, automates feature selection (Zhao et al., 2018; Singh and Singh, 2023; Banerjee et al., 2023) using word-level features and techniques like recurrent neural networks (Bahnsen et al., 2017). Muhammad et al. (2014) further advanced this field by developing a novel self-structured neural network (NN) specifically for identifying phishing websites. They evaluated this network using 17 distinctive signatures derived from 800 phishing and 600 legitimate websites sourced from PhishTank and Millersmiles, incorporating some data from third-party services. Their studies demonstrated the robustness and adaptability of neural networks in detecting phishing. They explored a backpropagation-trained feedforward neural network (Mohammad et al., 2013; Dawn et al., 2023) for further classifying websites. A significant advancement in phishing detection involved focusing on character-level features from URLs, recognizing that language and sentiment can be discerned from character sequences (Zhang et al., 2015). This shift towards character-level analysis reduces the need for extensive feature selection or preprocessing, allowing researchers to optimize computational efficiency and structural design in deep learning models.

Jain and Gupta (2018) proposed a machine learning-based method to detect phishing websites, focusing exclusively on client-side features. They utilized 19 specific features derived from URLs and source code to assess their approach. The evaluation involved testing on 2,141 phishing pages from PhishTank and Openfish, alongside 1,918 legitimate pages from the Alexa database and several online payment and banking websites (Jain and Gupta, 2018). A key part of their study was examining the impact of data augmentation on phishing URL detection performance through the use of generative adversarial networks (GANs).

Despite the breadth of features used in various studies for phishing detection, it has been noted that some features may not be adequate for reliably identifying phishing attempts (Anand et al., 2018). The selection of the most effective features has not been a primary focus in the field. To address this, Rajab advocated for the use of correlated feature sets and information gain to enhance phishing site identification (Rajab, 2018). In an analysis using the UCI repository, information gain and correlation-based feature selection methods were used to identify the most relevant features—11 and 9 features were selected out of 30, respectively, across 11,055 samples. The effectiveness of these selected features was further validated using the data mining algorithm RIPPER, showcasing a methodical approach to refine phishing detection through strategic feature selection. Bu and Cho employed an unsupervised learning method to tackle phishing attacks, uncovering significant class imbalances in the classification of phishing URLs (Bu and Cho, 2021). In a similar vein, Babagoli et al. utilized a comparable dataset and recommended the use of decision trees and wrapper methods for feature selection, ultimately selecting 20 features (Le et al., 2018). They further enhanced their approach with a novel metaheuristic-based nonlinear regression technique to evaluate phishing site performance (Babagoli et al., 2018).

However, these feature selection methods depend heavily on the underlying data and require the setting of user-specified thresholds, which can significantly influence the final performance of the classification algorithms, especially when features are selected from data not seen during the training phase.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.006

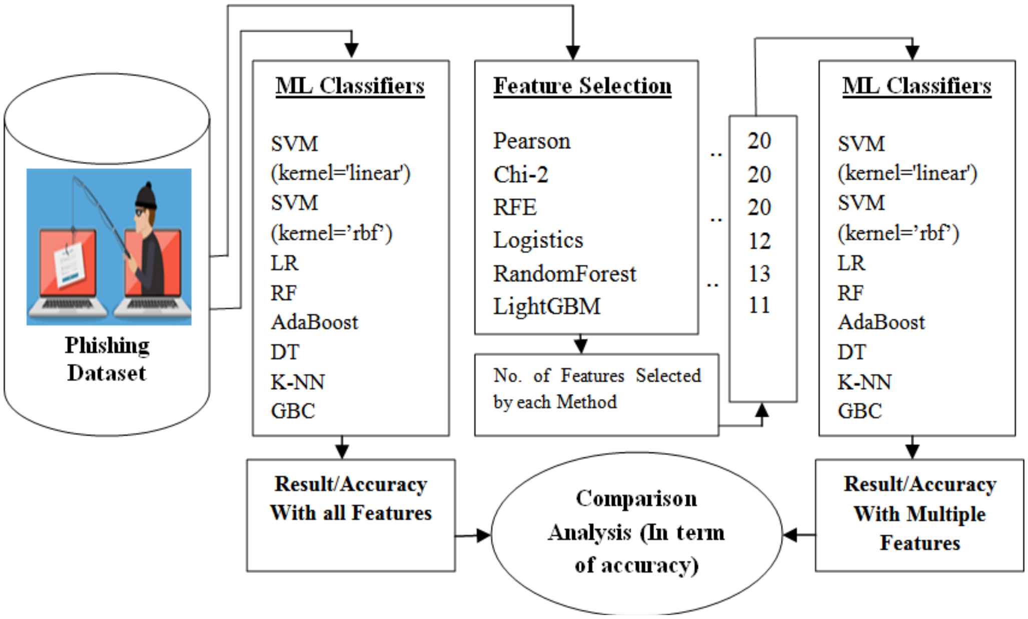


Figure 2. Experiment's flow diagram.

In a notable advancement, Microsoft developed a deep learning model that enhances phishing attack detection by integrating both character-level and word-level features (Tajaddodianfar et al., 2020). This model employs deep learning techniques like the self-attention mechanism to refine the URL feature set, making it one of the most accurate and reliable phishing detection methods currently available. Further improving upon this, Bu and Cho have optimized performance by using expert knowledge-based feature sets along with character- and word-level URL features (Bu and Cho, 2021). Additionally, first-order logic-based rules have been employed to correct outputs from deep learning classifiers, highlighting the ongoing efforts to refine the feature set for phishing detection. The integration of deep learning with traditional machine learning algorithms and genetic algorithms has also been explored for enhancing performance (Pal et al., 2023; Kumar et al., 2023; Yadav and Singh, 2023; Jain et al., 2024). For instance, Suleiman et al. (2019) have boosted the accuracy of various classifiers, including Naive Bayes (NB), k-Nearest Neighbors (k-NN), Decision Trees (DT), and Random Forests (RF), by incorporating evolutionary computation-based feature selection algorithms into these traditional machine learning frameworks. Similarly, Park et al. (2021) have leveraged genetic algorithms to improve the discovery rules, thereby increasing both the precision and recall of deep learning classifiers and enhancing overall detection performance. These developments underscore a multi-faceted approach to improving phishing website detection through innovative combinations of machine learning techniques.

Experimental Methodology

In this section, the focus is on the experimental setup where various machine learning classifiers were evaluated both before and after implementing a feature selection process (Cai et al., 2017). Six different feature selection methods were utilized, each determining the optimal number of features to use for enhancing classifier performance. The architecture of the proposed method is illustrated comprehensively in Figure 2, which depicts the comparative results obtained from the classifiers with and without feature selection, as well as the specific number of features selected by each feature selection algorithm. This section provides a concise overview of all the machine learning classifiers and feature selection algorithms used in the experiment. This description aims to offer a clear understanding of how each classifier and feature selection method contributes to the overall effectiveness of the phishing detection process, highlighting the improvements in performance achieved through the strategic reduction of features.

Machine Learning Classifiers

Algorithms called machine learning classifiers are used to group data according to input characteristic into predetermined groups or categories (Awasthi and Goel, 2021c). Support Vector Machine (SVM) is a common classifier; it finds the best hyperplane to divide classes; Logistic Regression (LR) models the probability of class membership using the logistic function; Random Forest (RF) constructs multiple decision trees and combines their outputs for improved accuracy; AdaBoost divides data into branches based on feature values and combines weak classifiers to form a strong classifier (); Decision Tree (DT) divides data into branches; K-Nearest

Neighbors (K-NN) classifies based on the majority class among the nearest neighbors; and Gradient Boosting Classifier (GBC) builds models sequentially to correct the errors of the previous ones.

Support vector machine (SVM)

An effective supervised learning approach for regression and classification problems is called a Support Vector Machine (SVM). It operates by determining which hyperplane in the feature space best divides the data points of various classes (Taher et al., 2018). Using kernel functions such as linear, polynomial, and radial basis function (RBF) to shift the input space into higher dimensions where a linear separator is more effective, SVM can handle both linear and non-linear data. Maximizing the margin, or the distance, between the closest support vector data points from each class and the hyperplane is the aim. This maximizing enhances the model's resilience to novel, untested data and its capacity for generalization. SVM works very well in high-dimensional areas and situations.

Logistic Regression (LR)

For binary classification problems, one popular statistical technique is logistic regression (LR). In contrast to linear regression, which forecasts continuous results, logistic regression (LR) models the likelihood that an input falls into a certain class (Thabtah et al. 2019). The logistic function, sometimes referred to as the Sigmoid function, is used to achieve this. It maps expected values to a probability range between 0 and 1. By using maximum likelihood estimation to estimate the coefficients for each input feature, the model is able to ascertain how each feature affects the result. Effectiveness, readability, and simplicity are the main benefits of logistic regression, particularly when there is a linear connection between the target variable's log-odds and its characteristics (Josephine et al., 2021). It's often used in situations like forecasting binary results for things like spam identification and illness presence.

Random Forest (RF)

In order to generate many decision trees during training and provide the mode of the classes (for classification) or mean prediction (for regression) of the individual trees, Random Forest (RF) is an ensemble learning technique used for classification and regression problems. It combines the ideas of random feature selection, which selects a random subset of features for splitting at each node in the tree, and bagging (Sun et al., 2017), which creates several subsets of the dataset by random sampling with replacement. By averaging out individual tree biases, this method enhances the model's generalization and decreases overfitting. Because every tree in the forest has received individual training, the combined result is a forecast that is more reliable and accurate. Random Forest is renowned for its exceptional precision and capacity to manage the dataset.

AdaBoost

Adaptive Boosting, or AdaBoost, is an ensemble learning technique that builds a strong classifier by aggregating the results of many weak classifiers. It operates by gradually training on the dataset weak classifiers, which are usually decision trees (Bansal et al., 2022). Every classifier concentrates on the examples that the preceding ones misclassified. Each training instance receives a weight throughout this procedure, which raises the weight of examples that are erroneously categorized so that later classifiers will give them more consideration. The final model is a robust classifier that decreases overfitting and increases accuracy, which is derived from the weighted sum of the weak classifiers. Continuing this iterative approach, each classifier focuses on the challenging cases to produce a strong final model that includes the best features of all the weak ones.

Decision Tree (DT)

Decision trees are a popular method in supervised machine learning, where they are used to model decisions and their possible consequences, similar to a flowchart. This algorithm splits the data into branches at decision nodes, which represent tests on certain attributes. Each split is based on the attribute that results in the most distinct separation of the data into groups based on the target variable (Gøttcke et al., 2021). The structure of a decision tree includes two main elements: decision nodes and leaves. Decision nodes are the points where the data is split. Each decision node represents a question based on an attribute, with the branches from the node answering this question. Leaves, on the other hand, represent the final outcomes or decisions.

k-Nearest Neighbor (k-NN)

The k-nearest neighbor classifier is one method for nonparametric supervised machine learning. It relies on distance: It classifies objects according to the classes of their closest neighbors. The most common application for KNN is classification, but it can also be used to solve regression issues. Labels in the training set serve as a guide for learning in a supervised model (Tekouabou et al., 2020). Check out our in-depth explanation of the principles of supervised learning for a better understanding of how it works. It is suitable for data where the relationship between the independent variable and the dependent variable is not a straight line, rather than simple models like linear regression.

Gradient boosting classifier (GBC)

In Gradient Boosting, each successive predictor aims to improve upon its predecessor by reducing the prediction error. Unlike traditional methods where a predictor is fit directly to the data, Gradient Boosting takes a unique approach by fitting a new predictor to the residual errors made by previous predictors (Awasthi and Goel, 2022). This iterative process begins with an initial prediction based on the dataset, often calculated by taking the logarithm of the probability of the target feature. Typically, this is done by dividing the number of true outcomes by the number of false outcomes. Each new predictor then focuses on correcting the mistakes of the preceding model, refining the overall prediction accuracy with each iteration. This strategy of incrementally correcting errors makes Gradient Boosting a powerful technique for building highly accurate predictive models.

Feature Selection Algorithms

Feature selection in machine learning is the process of taking out features that are noisy, redundant, or unnecessary in order to find the most relevant subset of the original set. This procedure is essential for enhancing classifier accuracy since it concentrates on the most important elements. Six different feature selection techniques were used in this research to determine which traits were most important and relevant. Through the removal of less valuable data, these techniques sought to improve the classifier's accuracy (Abdul Khalek et al., 2019). Table 1 provides a comprehensive list of all the features selected using these various feature selection techniques. By applying these methods, the study seeks to streamline the dataset, thereby optimizing the performance of the machine learning models used for classification.

Pearson correlation

Pearson Correlation creates a matrix measuring the linear association between features, providing values from -1 to 1. It evaluates the relationship between each feature and the target variable to identify the feature with the greatest impact on the target (Ali et al., 2019).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

r =Where n is the number of records in the dataset, x is the average value of the sample attribute, x is the ith value of the variable, and y is the target variable. 1 indicates a correlation, -1 indicates a correlation, and 0 indicates no correlation.

Chi-2

The chi-2 test was used to verify the independence of attributes in statistical models (Li et al., 2022). The model measures the difference between expected and actual responses. A lower Chi-2 value indicates that the variables are less dependent on one another, while a higher value indicates a greater correlation. The null hypothesis is based on the initial assumption that the attributes are distinct from one another. The following formula is used to determine the value of the expected result:

$$E_i = P(x_i \cap y_i) = P(x_i) \times P(y_i)$$

The following expression can be used to calculate the chi-square:

$$\chi 2 = \sum_{i=1}^{n} \frac{O_i - E_i}{E_i}$$

Where, i → range from 1 to n,

n → dataset records,

O_i → actual outcome,

E_i → the expected outcome

Recursive feature elimination (RFE)

The individual properties of features and how they interact with one another are the primary focus of the fundamental methods for selecting features. Based on variance and the correlation between them, some examples of methods that remove unnecessary features include variance thresholding and pairwise feature selection. However, a more practical strategy would choose features based on how they affect the performance of a particular model. By removing features one at a time until the optimal number of features are left, it reduces model complexity. Recursive Feature Elimination, also known as RFE Feature Selection, is a method of selecting features that cuts down on the complexity of a model by picking the most important ones and removing the weaker ones (Chen et al., 2018). The selection procedure eliminates these less important characteristics one at a time until it reaches the optimal number required for optimal performance. The model's dependencies and collinear ties are then removed by recursively removing a small number of features per loop. The number of features reduced by recursive feature elimination results in an increase in model efficiency.

Logistic Regression (LR)

Logistic regression establishes a relationship between predictor variables and the probability of an outcome using the Sigmoid function instead of a linear function like in Linear Regression (Alsouda et al., 2019). This makes it suitable for binary classification tasks, where it models the probability of a particular class.

$$log\, log \left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x$$

Where, $\frac{p(x)}{1-p(x)}$ → odd term and $log\ log\left(\frac{p(x)}{1-p(x)}\right)$ →logit or log-odds function.

Random Forest (RF)

A supervised model called Random Forest employs both decision trees and bagging (Awasthi and Goel, 2022). The idea is to resample the training dataset using a technique called "bootstrap". Fit a decision tree with each sample containing a random subset of the original columns. Based on its ability to increase the purity of its leaves, each Random Forest tree is able to determine the importance of features. The importance of this feature increases with leaf purity. This is done for each tree, averaged over all trees, and then normalized to 1 at the end. As a result, the random forest's importance scores all add up to 1.

LightGBM

A gradient boosting framework called Light GBM makes use of a tree-based learning algorithm (Rufo et al., 2021). The tree is grown vertically by Light GBM and horizontally by another algorithm. As a result, Light GBM creates trees one layer at a time.

Experimental Setup

The used dataset comes from the Kaggle Repository's Phishing website dataset (Phishing website dataset | Kaggle, https://www.kaggle.com/datasets). The phishing dataset has 32 features; the feature with the name Index has been removed because it only contains serial numbers. Table 2 shows that of the 31 features, 30 are independent and 1 is dependent. The Result is the final feature, indicating whether the website is phishing (1) or legitimate (0). As depicted in Figure 3, there are 4898 legitimate websites and 6157 phishing websites.

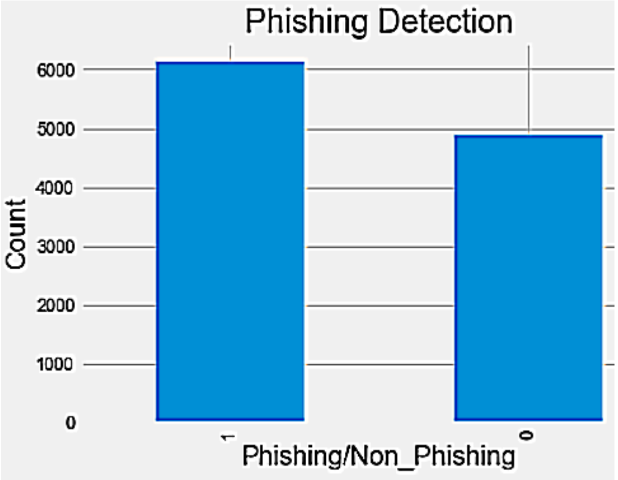


Figure 3. Phishing and legitimate websites.

Results

The findings in this manuscript are based on analyses performed both prior to and following feature selection. By comparing these results, we can evaluate if utilizing a reduced set of features leads to better performance compared to using the full feature set. We begin by examining the outcomes derived from all features (f1, f2, ..., f30), as detailed in Table 3. These results encompass evaluations of accuracy, recall, precision, F1-score, and confusion matrices, along with the feature correlation matrix and ROC curve analysis. These thorough evaluations serve as the cornerstone of our conclusions.

Results before feature selection

A correlation matrix was first constructed to examine the relationships between the coefficients of various variables (Qiu et al., 2021). This matrix summarizes the phishing dataset and helps identify and visualize patterns within the data. It illustrates the correlation between all 31 pairs of feature values in a tabular format, with variables displayed in rows and columns. The correlation coefficient for each pair can be found in the corresponding cell of the table. Additionally, the correlation matrix is often used alongside other types of statistical analysis.

Figure 4 demonstrates that the ranks of the 12 features—f5, f4, f1, f9, f2, f11, f6, f19, f8, f7, f16, and f18—are highly correlated. In the next step, we applied various machine learning classifiers to our dataset with all features. As previously mentioned, a range of classifiers was used to predict accuracy based on the dataset. Table 4 presents the results of several experiments involving machine learning-based classification of the dataset's features. For the evaluation and comparison of the learning algorithms, the dataset was divided into two parts: 80% was used for training and 20% for testing. To ensure the robustness of our evaluation, K-fold cross-validation was employed to validate the dataset. This method allowed for a comprehensive assessment of each algorithm's performance by repeatedly training and testing on different subsets of the data, thus minimizing the potential for bias and improving the reliability of our results. After training, the dataset was tested using different machine learning classifiers. At this stage, various algorithms were applied to distinguish between phishing and non-phishing website URLs. The dataset performed well across the eight machine learning classifications. This initial stream experiment, conducted before feature selection, aimed to obtain results from straightforward classification. Both RF and DT classifiers achieved the highest accuracy—96.06%—on the test dataset, resulting in a tie. Table 4 illustrates the training and testing outcomes across various classifiers.

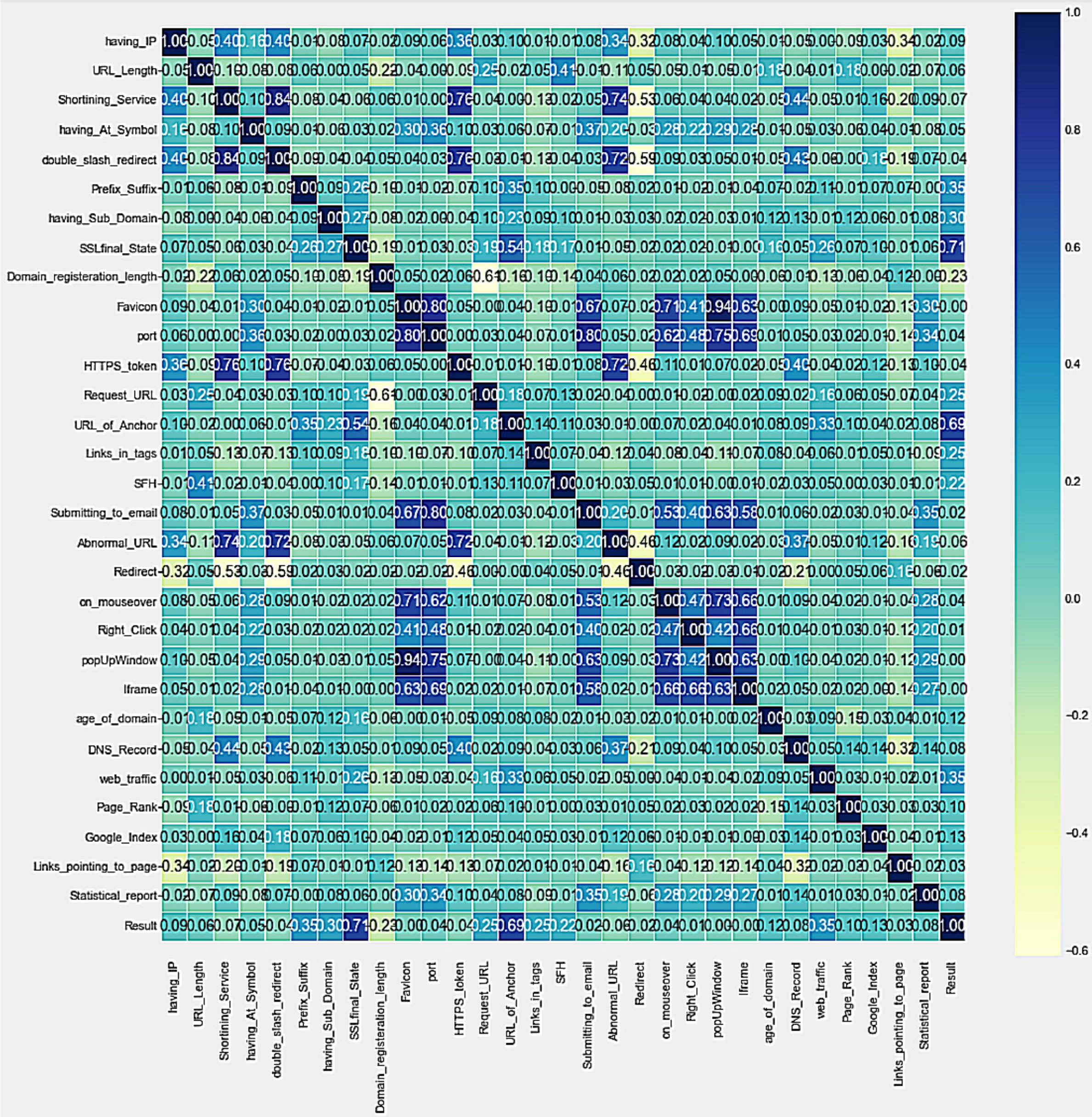


Figure 4. Correlation matrix.

Table 4. Accuracy (Train and Test) of the classifiers with all features.

Accuracy	SVM (kernel='linear')	SVM (kernel='rbf')	LR	RF	Ada Boost	DT	K-NN	GBC
Train	92.84%	95.41%	92.94%	99.06%	93.96%	99.06%	96.55%	95.28%
Test	92.85%	94.71%	92.40%	96.74%	93.58%	95.97%	94.08%	95.07%

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.006

Figure 5 depicts the corresponding outcome. To evaluate Recall, Precision, Specificity, Accuracy, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve), the confusion matrix was employed. The confusion matrix is a table that displays the four possible outcomes of predicted and actual values: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). This matrix is crucial for calculating various performance metrics that provide a comprehensive understanding of the model's effectiveness.

Table 6 presents the precision, recall, and F1 scores for phishing and legitimate URLs across both the training and testing datasets. Additionally, the confusion matrix scores have been extracted for these datasets. On the test dataset, the RF classifier stands out with a precision of 96.31%, a recall of 98.00%, and an F1-score of 97.15%. The validation score for the RF classifier is also very similar to that of the DT classifier. Furthermore, the confusion matrix-based results are closely aligned with those of the DT classifier.

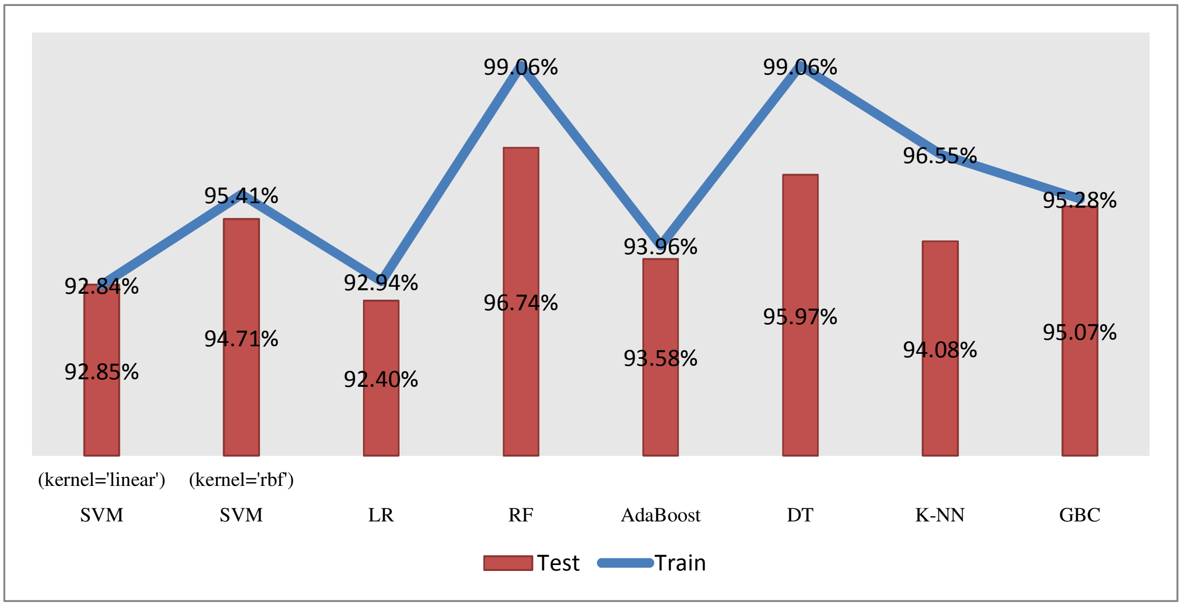
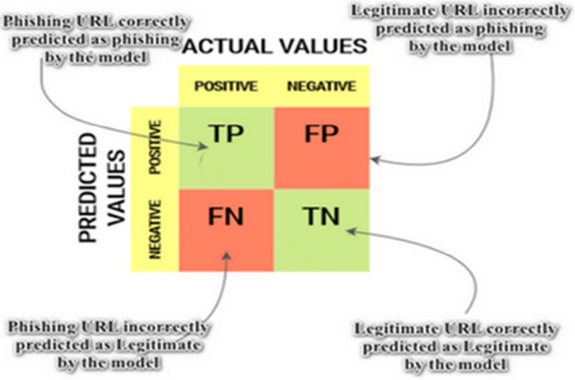


Figure 5. Visualization of classifier accuracy across all features during training and testing.

Table 5. Metrics for validation used in the experiment.

Validation measures	Using formula
Precision	$\frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positve\ (FP)}$
Recall	$\frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$
F1-score	$2 \times \frac{Precision \times Recall}{Precison + Recall}$
Confusion Matrix	

Phishing URL correctly predicted as phishing by the model
ACTUAL VALUES
Legitimate URL incorrectly predicted as phishing by the model
POSITIVE
NEGATIVE
PREDICTED VALUES
POSITIVE
NEGATIVE
TP
FP
FN
TN
Phishing URL incorrectly predicted as Legitimate by the model
Legitimate URL correctly predicted as Legitimate by the model

Table 6. Performance Metrics (Train and Test) of the classifiers with all features.

Classifiers	precision				recall				f1-score				Confusion Matrix	
	Train (%)		Test (%)		Train (%)		Test (%)		Train (%)		Test (%)		Train	Test
	0	1	0	1	0	1	0	1	0	1	0	1		
SVM (kernel='linear')	93.11	92.63	93.18	92.61	92.63	94.61	90.06	94.98	91.86	93.61	91.59	93.78	[[3573 369] [264 4638]]	[[861 95] [63 1192]]
SVM (kernel='rbf')	96.11	94.87	95.35	94.24	93.48	96.96	92.25	96.57	94.77	95.90	93.77	95.39	[[3685 257] [149 4753]]	[[882 74] [43 1212]]
LR	93.20	92.74	91.91	92.76	90.79	94.67	90.37	93.94	91.98	93.70	91.13	93.34	[[3579 363] [261 4641]]	[[864 92] [76 1179]]
RF	99.25	98.90	97.32	96.31	98.63	99.40	95.08	98.00	98.94	99.15	96.19	97.15	[[3888 54] [29 4873]]	[[909 47] [25 1230]]
AdaBoost	94.37	93.64	93.57	93.57	91.93	95.59	91.42	95.21	93.13	94.60	92.48	94.39	[[3624 318] [216 4686]]	[[874 82] [60 1195]]
DT	99.00	99.10	95.48	96.34	98.88	99.20	95.18	96.57	98.94	99.15	95.33	96.45	[[3898 44] [39 4863]]	[[910 46] [43 1212]]
K-NN	96.64	96.48	93.74	94.32	95.58	97.32	92.46	95.29	96.11	96.90	93.10	94.80	[[3768 174] [131 4771]]	[[884 72] [59 1196]]
GBC	95.67	94.98	95.29	94.90	93.65	96.59	93.20	96.49	94.65	95.78	94.23	95.69	[[3692 250] [167 4735]]	[[891 65] [44 1211]]

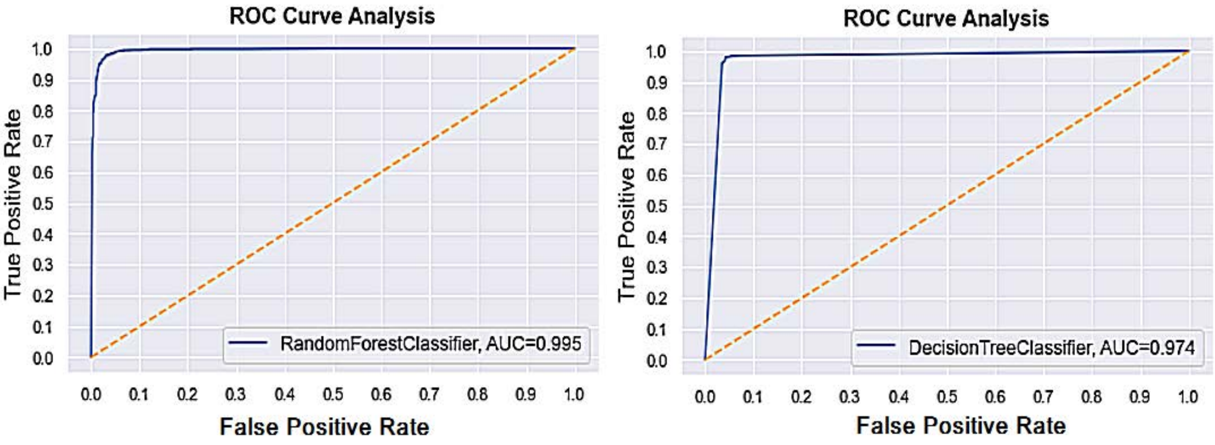


Figure 6. RF and DT ROC (AUC) curves.

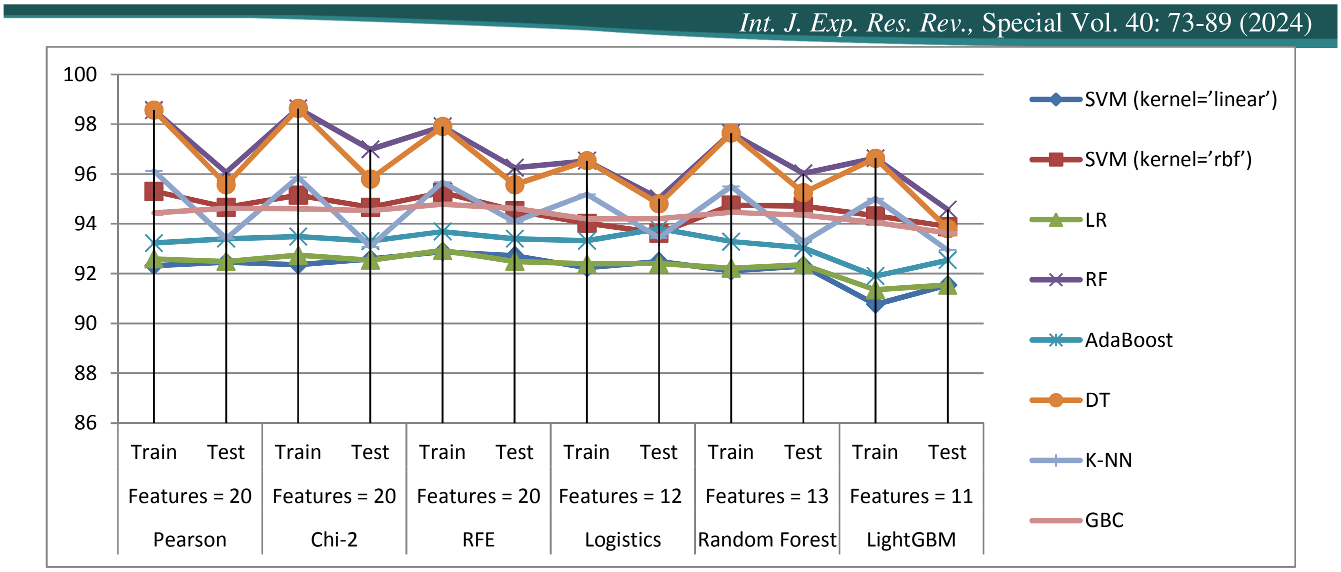


Figure 7. Accuracy of the classifiers for various feature counts.

Table 7. Classifier accuracy (Train and Test) for various feature selections.

Classifiers	Pearson Features = 20		Chi-2 Features = 20		RFE Features = 20		Logistics Features = 12		Random Forest Features = 13		LightGBM Features = 11	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
SVM (kernel=’linear’)	92.32	92.45	92.35	92.58	92.88	92.72	92.24	92.49	92.11	92.31	90.76	91.54
SVM (kernel=’rbf’)	95.30	94.66	95.15	94.66	95.26	94.53	94.02	93.62	94.75	94.71	94.32	93.89
LR	92.59	92.49	92.74	92.54	92.93	92.49	92.39	92.40	92.21	92.36	91.35	91.54
RF	98.56	96.07	98.63	96.99	97.91	96.25	96.53	95.02	97.64	96.02	96.63	94.57
AdaBoost	93.22	93.40	93.49	93.31	93.69	93.40	93.32	93.80	93.28	93.03	91.90	92.54
DT	98.56	95.57	98.63	95.79	97.91	95.57	96.53	94.80	97.64	95.25	96.63	93.80
K-NN	96.12	93.40	95.87	93.08	95.67	94.08	95.18	93.44	95.50	93.26	95.01	92.94
GBC	94.43	94.62	94.60	94.53	94.78	94.62	94.19	94.21	94.46	94.35	94.05	93.62

The ROC (AUC) curve offers a comprehensive measure of performance across all classification thresholds. As demonstrated in the previous results, metrics such as accuracy, precision, recall, the F1-score, and the confusion matrix have very similar scores. Therefore, additional clarification of the results based on these metrics is necessary. Figure 6, derived from these metrics, shows that the RF classifier achieves a higher ROC (AUC) score compared to the DT classifier.

Results after feature selection

Feature selection algorithms have garnered significant attention across a wide range of applications. These algorithms simulate a "survival of the fittest" evolution to search the solution space. Table 7 displays the scores obtained by various feature selection algorithms for different numbers of features from the simulation results. Multiple scores are produced by the eight classifiers based on their training and testing results (accuracy) on fewer features. When comparing these scores, it is evident that the Chi-2 feature selection algorithm provided the RF classifier with the highest testing accuracy—96.99%—using 20 features. Conversely, the RF classifier achieved the second-highest score (96.25%) using 20 features with a different feature selection method.

Figure 7 is the conclusion of the data presented in Table 7, which provides a summary of the previous findings. According to Table 7, In this research, the objective was to identify the most effective URL-based features for phishing detection by employing six different feature selection algorithms: Pearson, Chi-square (Chi-2), Logistic Regression (Logistics), Random Forest, Light Gradient Boosting Machine (Light GBM), and Recursive Feature Elimination (RFE). Each of these algorithms selected a set number of features from an initial pool, demonstrating an enhancement in detection accuracy

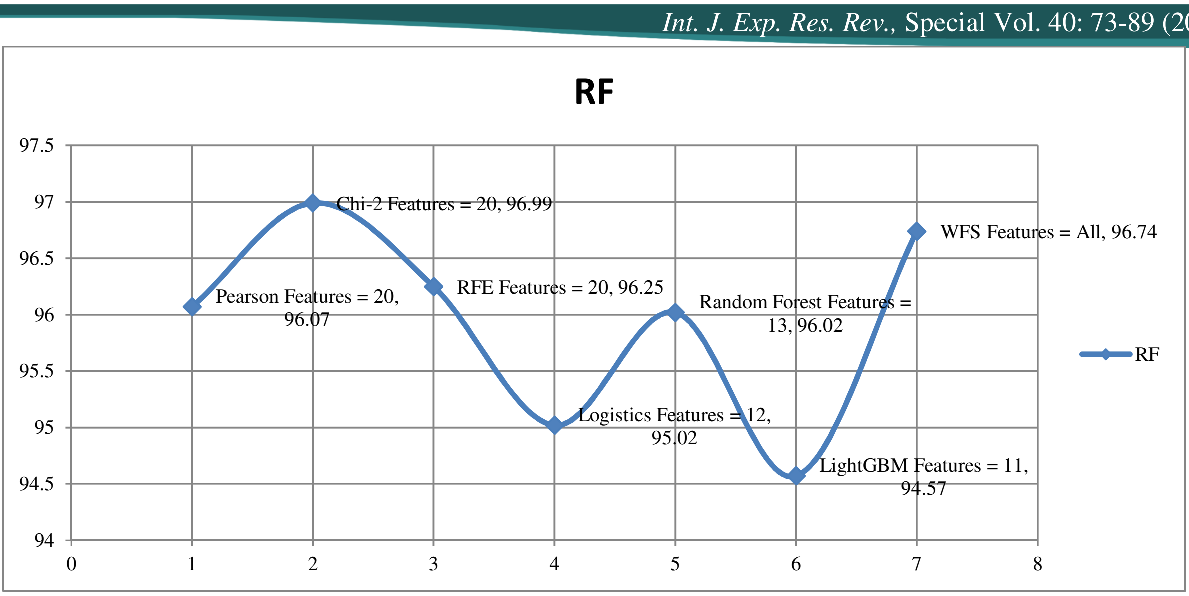


Figure 8. A test of the accuracy of the classifiers with all features and different numbers of features

Table 8. Classifier accuracy (Test) for various feature selections

Test	Pearson Features = 20	Chi-2 Features = 20	RFE Features = 20	Logistics Features = 12	Random Forest Features = 13	LightGBM Features = 11	WFS Features = All
RF	96.07	96.99	96.25	95.02	96.02	94.57	96.74
*WFS → without feature selection							

when used with various classifiers such as SVM (both linear and radial basis function), Logistic Regression (LR), Random Forest (RF), AdaBoost, Decision Tree (DT), K-Nearest Neighbors (K-NN), and Gradient Boosting Classifier (GBC). The feature selection processes resulted in a range of features being chosen by each algorithm, with Pearson, Chi-2, and RFE selecting up to 20 features each, and Light GBM, Random Forest, and Logistic Regression selecting fewer—12, 13, and 11 features respectively. The selected features were then used to train and test the classifiers, and the accuracies were recorded. RF stood out, achieving the highest accuracy of 96.74%, indicating superior performance over other classifiers. The validation metrics including precision, recall, f1-score, and a confusion matrix further reinforced RF's efficacy. The research highlighted the impact of feature selection on the efficiency of phishing detection models. For instance, using the Chi-2 method, the RF classifier achieved a high accuracy of 96.99% with just 20 features, compared to 96.94% accuracy with all 31 features, showing that feature reduction can still preserve or even enhance model performance. This strategy not only simplifies the model but also optimizes computational efficiency without compromising detection capability. The findings suggest that a carefully selected subset of features can effectively support robust phishing detection, underscoring the importance of feature selection in building efficient security models in the cyber domain. Table 8 presents the classifier accuracy (Test) for various feature selections. Using Pearson with 20 features, the accuracy is 96.07. For Chi-2 with 20 features, it is 96.99. RFE with 20 features yields an accuracy of 96.25. Logistic regression with 12 features results in an accuracy of 95.02. Random Forest with 13 features achieves an accuracy of 96.02, while LightGBM with 11 features gives 94.57. WFS with all features provides an accuracy of 96.74.

Discussion

In this study, the primary goal was to enhance the detection of phishing websites by selecting the most effective URL-based features using a variety of feature selection algorithms. These algorithms, detailed in Table 1, include Pearson correlation, Chi-square (Chi-2), Logistic regression, Random Forest (RF), Light Gradient Boosting Machine (Light GBM), and Recursive Feature Elimination (RFE). By automating the feature selection process, these methods significantly improved detection accuracy. Table 3 highlights the number of features selected by each method. The efficiency of feature selection was demonstrated through the application of several classifiers, including Support Vector Machine (SVM) with both linear and radial basis function (rbf) kernels, Logistic Regression (LR), RF, AdaBoost, Decision Tree (DT), K-Nearest Neighbors (K-NN), and Gradient Boosting Classifier (GBC). Each classifier was evaluated on the entire phishing dataset, with performance metrics such as precision, recall, f1-score, and confusion matrix presented in Table 6. Among these, RF achieved the highest accuracy of 96.74%, validating the effectiveness of the feature selection algorithms. Notably, Pearson, Chi-2, and RFE each selected 20 out of a possible 30 features (Table 7), resulting in high testing accuracies of 96.07%, 96.99%, and 96.25%, respectively.

Table 9. Comparing our approach to that of recent studies, Where NR→ Not Reported.

Author	Method	No. of Features	Accuracy	Precision	Recall	F1 score
Chiew et al. (2019)	Hybrid Ensemble Feature Selection (HEFS) Cumulative Distribution Function gradient (CDF-g) Algorithm	48 with Dataset 1	96.17%	NR	NR	NR
Chiew et al. (2019)	Hybrid Ensemble Feature Selection (HEFS) Cumulative Distribution Function gradient (CDF-g) Algorithm	10 with Dataset 1	94.60%	NR	NR	NR
Chiew et al. (2019)	Hybrid Ensemble Feature Selection (HEFS) Cumulative Distribution Function gradient (CDF-g) Algorithm	30 with Dataset 2	94.27%	NR	NR	NR
Chiew et al. (2019)	Hybrid Ensemble Feature Selection (HEFS) Cumulative Distribution Function gradient (CDF-g) Algorithm	5 with Dataset 2	93.22%	NR	NR	NR
Zhu et al. (2019)	OFS-NN neural network	30	96.44%	94.78%	99.02%	96.85%
Ours	RF	30	96.74%	96.31%	98.00%	97.15%
Ours	RF with Chi-2 Feature Selection	20	96.99%	NR	NR	NR

Conclusion and Future Work

Website phishing is a serious cyber threat that targets unsuspecting internet users, aiming to capture sensitive personal information such as usernames, passwords, and financial details. In our research, we explore effective methods for identifying fake websites, focusing on critical features that distinguish these sites. We introduce six strategies for selecting the most informative features to aid in the detection of phishing attempts. Additionally, we developed a strategy for detecting phishing websites using eight different machine-learning algorithms. Among these, the Random Forest (RF) classifier was found to be the most accurate, providing high detection rates even with limited features. Our method is capable of identifying phishing sites in real time, offering better performance compared to existing solutions. Future work will enhance our model by incorporating webpage content analysis once a webpage is fully loaded on a user's device, providing a more robust defense by combining URL-based and content-based detection techniques.

Conflicts of Interest

Authors have disclosed no competing interests.

References

Abdul-Khalek, R., Ball, R. D., Carrazza, S., Forte, S., Giani, T., Kassabov, Z., ... & Wilson, M. (2019). A first determination of parton distributions with theoretical uncertainties. *The European Physical Journal C*, 79(10), 1-6. https://doi.org/10.1140/epjc/s10052-019-7364-5

Ali, L., Rahman, A., Khan, A., Zhou, M., Javeed, A., & Khan, J. A. (2019). An automated diagnostic system for heart disease prediction based on ${\chi^{2}} $ statistical model and optimally configured deep neural network. *IEEE Access, 7*, 34938-34945. https://doi.org/10.1109/ACCESS.2019.2904800

Alsouda, Y., Pllana, S., & Kurti, A. (2019). Iot-based urban noise identification using machine learning: performance of SVM, KNN, bagging, and random forest. *In Proceedings of the International Conference on Omni-layer Intelligent Systems,* pp. 62-67. https://doi.org/10.1145/3312614.3312631

Anand, A., Gorde, K., Moniz, J.R.A., Park, N., Chakraborty, T., & Chu, B.T. (2018). Phishing URL detection with oversampling based on text generative adversarial networks. *In Proceedings of the 2018 IEEE International Conference on Big Data* (Big Data), Seattle, WA, USA. pp. 1168-1177. https://doi.org/10.1109/BigData.2018.8622547

Awasthi, A., & Goel, N. (2021a). Phishing Website Prediction: A Comparison of Machine Learning Techniques. Springer, Singapore, *In Data Intelligence and Cognitive Informatics*, pp. 637-650. https://doi.org/10.1007/978-981-15-8530-2_50

Awasthi, A., & Goel, N. (2021b). Phishing Website Prediction: A Machine Learning Approach. Springer, Singapore, *In Progress in Advanced Computing and Intelligent Engineering*, pp. 143-152. https://doi.org/10.1007/978-981-33-4299-6_12

Awasthi, A., & Goel, N. (2021c). Generating Rules to Detect Phishing Websites Using URL Features. IEEE, *In 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology* (ODICON), pp. 1-9. https://doi.org/10.1109/ODICON50556.2021.9429003

Awasthi, A., & Goel, N. (2022). Phishing website prediction using base and ensemble classifier techniques with cross-validation. *Cybersecurity, 5*(1), 1-23. https://doi.org/10.1186/s42400-022-00126-9

Babagoli, M., Aghababa, M.P., & Solouk, V. (2018). Heuristic nonlinear regression strategy for detecting phishing websites. *Soft Computing, 23*, 4315-4327. https://doi.org/10.1007/s00500-018-3084-2

Bahnsen, A.C., Bohorquez, E.C., Villegas, S.; Vargas, J., & González, F.A. (2017). Classifying phishing URLs using recurrent neural networks. *In Proceedings of the 2017 APWG Symposium on Electronic Crime Research* (eCrime), Scottsdale, AZ, USA. pp. 1-8. https://doi.org/10.1109/ECRIME.2017.7945048Banerjee, M., Goyal, R., Gupta, P., &

Tripathi, A. (2023). Real-Time Face Recognition System with Enhanced Security Features using Deep Learning. *Int. J. Exp. Res. Rev.*, *32*, 131-144. https://doi.org/10.52756/ijerr.2023.v32.011

Bansal, M., Goyal, A., & Choudhary, A. (2022). A comparative analysis of K-Nearest Neighbour, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decision Analytics Journal*, 100071. https://doi.org/10.1016/j.dajour.2022.100071

Bu, S.J., & Cho, S.B. (2021). Deep character-level anomaly *detection based on a convolutional autoencoder for zero-day phishing URL detection. Electronics, 10*, 1492. https://doi.org/10.3390/electronics10121492

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing, 300*, 70-79. https://doi.org/10.1016/j.neucom.2017.11.077

Chen, H., Gilad-Bachrach, R., Han, K., Huang, Z., Jalali, A., Laine, K., & Lauter, K. (2018). Logistic regression over encrypted data from fully homomorphic encryption. *BMC Medical Genomics*, *11*(4), 3-12. https://doi.org/10.1186/s12920-018-0397-z

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences, 484*, 153-166. https://doi.org/10.1016/j.ins.2019.01.064

Dawn, N., Ghosh, T., Ghosh, S., Saha, A., Mukherjee, P., Sarkar, S., Guha, S., & sanyal, T. (2023). Implementation of Artificial Intelligence, Machine Learning, and Internet of Things (IoT) in revolutionizing Agriculture: A review on recent trends and challenges. *Int. J. Exp. Res. Rev.*, *30*, 190-218. https://doi.org/10.52756/ijerr.2023.v30.018

Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. (2018). The application of a novel neural network in the detection of phishing websites. *Journal of Ambient Intelligence and Humanized Computing, 2018*. https://doi.org/10.1007/s12652-018-0786-3

Franjić, S. (2020). Cybercrime is Very Dangerous Form of Criminal Behavior and Cybersecurity. *Emerging Science Journal, 4*, 18-26. https://doi.org/10.28991/esj-2020-SP1-02

Gøttcke, J. M. N., Zimek, A., & Campello, R. J. (2021). Non-parametric semi-supervised learning by Bayesian label distribution propagation. Springer, Cham., *In International Conference on Similarity Search and Applications*, pp. 118-132. https://doi.org/10.1007/978-3-030-89657-7_10

Gupta, S., Cherukuri, A. K., Subramanian, C. M., & Ahmad, A. (2022). Comparison, Analysis and Analogy of Biological and Computer Viruses. Springer, Singapore, *In Intelligent Interactive Multimedia Systems for e-Healthcare Applications*, pp. 3-34. https://doi.org/10.1007/978-981-16-6542-4_1

Iuga, C., Nurse, J.R., & Erola, A. (2016). Baiting the hook: Factors impacting susceptibility to phishing attacks.

Hum. Cent. Comput. Inf. Sci., 6, 8. https://doi.org/10.1186/s13673-016-0065-2

Jain, A.K., & Gupta, B.B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems, 68*(4), 687-700. https://doi.org/10.1007/s11235-017-0414-0

Jain, P., Thada, V., & Motwani, D. (2024). Providing Highest Privacy Preservation Scenario for Achieving Privacy in Confidential Data. *International Journal of Experimental Research and Review, 39*(Spl Volume), 190-199. https://doi.org/10.52756/ijerr.2024.v39spl.015

Josephine, P. K., Prakash, V. S., & Divya, K. S. (2021). Supervised Learning Algorithms: A Comparison. *Kristu Jayanti Journal of Computational Sciences* (KJCS), pp. 01-12. https://doi.org/10.59176/kjcs.v1i1.1259

Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020). Feature selections for the classification of webpages to detect phishing attacks: a survey. IEEE, *In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications* (HORA), pp. 1-9. https://doi.org/10.1109/HORA49412.2020.9152934

Kumar, A., Dutta, S., & Pranav, P. (2023). Supervised learning for Attack Detection in Cloud. *Int. J. Exp. Res. Rev.*, *31*(Spl Volume), 74-84. https://doi.org/10.52756/10.52756/ijerr.2023.v31spl.008

Le, A., Markopoulou, A., & Faloutsos, M. (2011). Phishdef: Url names say it all. *In Proceedings of the 2011 Proceedings IEEE INFOCOM*, Shanghai, China, pp. 191-195. https://doi.org/10.1109/INFCOM.2011.5934995

Le, H., Pham, Q., Sahoo, D., & Hoi, S.C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv 2018, arXiv:1802.03162.

Li, L., Ching, W. K., & Liu, Z. P. (2022). Robust biomarker screening from gene expression data by stable machine learning-recursive feature elimination methods. *Computational Biology and Chemistry, 100*, 107747. https://doi.org/10.1016/j.compbiolchem.2022.107747

Mohammad, R., McCluskey, T., & Thabtah, F.A. (2013). Predicting phishing websites using neural network trained with back-propagation. *World Congress in Computer Science, Computer Engineering, and Applied Computing*.

Mohammad, R.M., Thabtah, F., & McCluskey, L. (2012). An assessment of features related to phishing websites using an automated technique. *In Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions*, London, UK. pp. 492-497.

Mohammad, R.M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications, 25*(2), 443-458, 2014. https://doi.org/10.1007/s00521-013-1490-z

Oest, A., Safei, Y., Doupé, A., Ahn, G. J., Wardman, B., & Warner, G. (2018). Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. IEEE, *In 2018 APWG Symposium on Electronic Crime Research* (eCrime), pp. 1-12. https://doi.org/10.1109/ECRIME.2018.8376206

Pal, R., Pandey, M., Pal, S., & Yadav, D. (2023). Phishing Detection: A Hybrid Model with Feature Selection and Machine Learning Techniques. *Int. J. Exp. Res. Rev.*, 36, 99-108. https://doi.org/10.52756/ijerr.2023.v36.009

Park, K.W., Bu, S.J., & Cho, S.B. (2021). Evolutionary optimization of neuro-symbolic integration for phishing URL detection. *In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems*, Bilbao, Spain. pp. 88-100. https://doi.org/10.1007/978-3-030-86271-8_8

Phishing website dataset | Kaggle, https://www.kaggle.com/datasets/akashkr/phishing-website-dataset?select=dataset.csv. Accessed 8th January 2023.

Qiu, P., & Niu, Z. (2021). TCIC_FS: Total correlation information coefficient-based feature selection method for high-dimensional data. *Knowledge-Based Systems, 231*, 107418. https://doi.org/10.1016/j.knosys.2021.107418

Rajab, (2018). An anti-phishing method based on feature analysis," in Proceedings of the 2nd International Conference on Machine Learning and Soft Computing. *ACM*, 133-139. https://doi.org/10.1145/3184066.3184082

Rufo, D. D., Debelee, T. G., Ibenthal, A., & Negera, W. G. (2021). Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM). *Diagnostics, 11*(9), 1714. https://doi.org/10.3390/diagnostics11091714

Sahingoz, O.K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from urls. *Expert Systems with Applications, 117*, 345-357. https://doi.org/10.1016/j.eswa.2018.09.029

Singh, D., & Singh, S. (2023). Precision fault prediction in motor bearings with feature selection and deep learning. *Int. J. Exp. Res. Rev.*, *32*, 398-407. https://doi.org/10.52756/ijerr.2023.v32.035

Srinivas, J., Das, A. K., & Kumar, N. (2019). Government regulations in cyber security: Framework, standards and recommendations. *Future Generation Computer Systems, 92*, 178-188. https://doi.org/10.1016/j.future.2018.09.063

Suleman, M.T., & Awan, S.M. (2019). Optimization of URL-based phishing websites detection through genetic algorithms. Autom. Control. *Comput. Sci., 53*, 333-341. https://doi.org/10.3103/S0146411619040102

Sun, J., Fujita, H., Chen, P., & Li, H. (2017). Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble. *Knowledge-Based Systems, 120*, 4-14. https://doi.org/10.1016/j.knosys.2016.12.019

Taher, S. A., Akhter, K. A., & Hasan, K. A. (2018). N-gram based sentiment mining for bangla text using support vector machine. IEEE, *In 2018 international conference on Bangla speech and language processing* (ICBSLP), pp. 1-5.

Tajaddodianfar, F., Stokes, J.W., & Gururajan, A. (2020). Texception: A character/word-level deep learning model for phishing URL detection. *In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), Barcelona, pp. 2857-2861. https://doi.org/10.1109/ICASSP40776.2020.9053670

Tekouabou, S. C. K., Cherif, W., & Silkan, H. (2020). Improving parking availability prediction in smart cities with IoT and ensemble-based model. *Journal of King Saud University-Computer and Information Sciences.*

Thabtah, F., Abdelhamid, N., & Peebles, D. (2019). A machine learning autism classification based on logistic regression analysis. *Health information science and systems, 7*(1), 1-11. https://doi.org/10.1007/s13755-019-0073-5

Yadav, R., & Singh, R. (2023). Enhancing Software Maintainability Prediction Using Multiple Linear Regression and Predictor Importance. *Int. J. Exp. Res. Rev.*, *36*, 135-146. https://doi.org/10.52756/ijerr.2023.v36.013

Zhang, X., Zhao, J., & LeCun, Y. Character-level convolutional networks for text classification. *In Proceedings of the Advances in Neural Information Processing Systems*, Montreal, QC, Canada, pp. 649-657.

Zhao, J., Wang, N., Ma, Q., & Cheng, Z. (2018). Classifying malicious URLs using gated recurrent neural networks. In Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Matsue, Japan. pp. 385-394. https://doi.org/10.1007/978-3-319-93554-6_36

Zhong, C., & Sastry, N. (2017). Systems applications of social networks. *ACM Computing Surveys* (CSUR), *50*(5), 1-42. https://doi.org/10.1145/3092742

Zhu, E., Chen, Y., Ye, C., Li, X., & Liu, F. (2019). OFS-NN: an effective phishing websites detection model based on optimal feature selection and neural network. *IEEE Access, 7*, 73271-73284. https://doi.org/10.1109/ACCESS.2019.2920655

How to cite this Article:

Anjaneya Awasthi and Noopur Goel (2024). An Approach for Efficient and Accurate Phishing Website Prediction Using Improved ML Classifier Performance for Feature Selection. *International Journal of Experimental Research and Review*, *40*(spl.), 73-89.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.006