Educational Administration: Theory and Practice
2023, 29(4), 1026 - 1042
ISSN: 2148-2403
https://kuey.net/

Research Article

Efficient And Accurate Technique For Improving ML Classifier Performance Using Feature Selection In Phishing Website Prediction

Anjaneya Awasthi[1*], Noopur Goel[2]

[1*]Research Scholar, Dept. of Computer Applications, VBS Purvanchal University, Jaunpur, UP, India. E-mail: anjaneyaawasthi@gmail.com
[2]Head, Dept. of Computer Applications, VBS Purvanchal University, , Jaunpur, UP, India

Citation: Anjaneya Awasthi (2023), Efficient And Accurate Technique For Improving ML Classifier Performance Using Feature Selection In Phishing Website Prediction, *Educational Administration: Theory and Practice, 29(4),* 1026 - 1042
Doi: 10.53555/kuey.v29i4.4369

ARTICLE INFO

ABSTRACT

Due to cyberattacks and various strategies, phishing websites are a problem on the Internet. One of these cyberattacks is phishing, in which the attacker pretends to be a trusted party to get sensitive and confidential information. Blacklisting, heuristic search, and visual similarity are just a few of the anti-phishing strategies that have been used to identify fraudulent activity. Machine learning (ML) techniques appear to be a beacon in the gloom of phishing websites, in contrast to these traditional methods, which take a long time to detect and have a high false rate. By introducing a novel features selection method in this article, it is possible to extract highly correlated features from datasets, thereby increasing the accuracy of classifiers over all features. Eight classifiers—Support vector machine (SVM kernel linear and rbf), Logistic regression (LR), Random forest (RF), Adaboost, Decision tree (DT), K-nearest neighbor (k-NN), and Gradient boosting (GBC)—as well as six feature selection techniques (Pearson, Chi-2, RFE, Logistics, Random Forest, and LightGBM) are used on phishing dataset with all features and feature selection methods. Comparing the results, it came to the conclusion that the random forest classifier and feature selection using the Chi-2 method have the potential to improve the model's accuracy. The accuracy of the proposed model reached as high as 96.99%.

1. Introduction

Both computer viruses and biological viruses are identical in nature. While computer viruses are nothing more than a small, malicious program that runs on a computer, biological viruses attack living cells [1]. Its goal was to copy itself to another computer and change the code of programs that could be run. It is harmful to all hosts it is on because it tends to spread quickly in human cells and computer programs. In software industry there might be numerous enemy of phishing programs that could be arrangement inside specific period [2]. These anti-phishing programs, on the other hand, are unable to detect all types of attacks because, rather than generating executable programs, they transfer fake web pages to end users in order to exploit their weaknesses and gain access to secret or sensitive data. These Phishing attacks operate in a similar way to viruses. Interaction causes biological viruses to become active [3]. Similar to interactions, phishing attacks occur. E-mail or messages from social networking sites are typically used for this kind of communication. Computer infections duplicate data by working behind the scenes without seeing the person in question. The victim's information is also retrieved by the phishing tools [4]. At this point, victims are asked to enter crucial information such as their credit card number, user name, password, and other personal information. Phishing attacks evolve in the same way that biological viruses change over time in order to escape detection by their victims [5]. This change focuses primarily on the fake website's visual similarity and the use of a reliable uniform resource locator (URL) or email. The loss—generally monetary loss— it is inevitable that the hacker will obtain important information from the victim if they trust one of these methods. According to a 2022 SlashNext report, the number of phishing attacks has increased by 61% since 2021. During the six months of analysis, 255 million attacks were identified in billions of emails, attachments, link-based URLs, browser channels, and mobile messages [6]. Figure 1 depicts the worldwide distribution of phishing URLs on infected devices prior to and during the year 2022.

Fig.1. Phishing URL distribution between 2020 and 2022

Since 2003, numerous agencies around the world have collaborated to reduce losses caused by phishing URLs. However, literacy is required to protect beside phishing attacks, and professionals' also as academic studies may play a significant role in preventing phishing attacks [7]. The goal of this study is to improve phishing website detection accuracy [8]. The process of detecting phishing websites is broken down into two distinct phases in this study. Eight machine-learning classifiers are used to classify the dataset in the first stage. In contrast, in the second stage, six feature selection algorithms and machine learning classifiers are used to represent the data. The goal was to choose prominent features that make classifiers with fewer features more accurate. Table 1 provides a description of the features that were chosen by various feature selection algorithms, as well as the abbreviations that are used to describe each feature in Table 2 and Table 3, which are presented in order.

Table1. 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_registeration_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

Table 2 Name 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_registeration_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 selected features	**20**	**20**	**20**	**12**	**13**	**11**

The leftover article is depicted as follows: The previous efforts to identify phishing websites are discussed in Section 2. Section 3 discusses the experimental diagram and its explanations. An explanation of the dataset and information about the attributes are included in Section 4. Section 5 discusses the experiment's results, while Sections 6 and 7 discuss the experiment's conclusion.

2. Literature Review

An active area of research is the classification of phishing websites using ML-based approaches using various supervised classification methods. This section discusses state-of-the-art machine learning-based phishing website detection techniques.

As a first attempt at modeling phishing attacks, a crude method of building feature sets from lists of words in URLs has been attempted as bag-of-words vectors [9]. Feng et al. [10] for phishing detection, a novel neural network is proposed. By implementing risk minimization principles, they enhance the network's generalizability. 11,055 samples marked as legitimate or phishing are available in the UCI repository for testing the proposed network's performance. In addition, the dataset specifies 30 features for each website, all of which are domain-based, exception-based, HTML and Javascript-based, and address bar-based. Muhammad et al. [11] by systematically extracting URL features and proposing hierarchical classifiers based on the extraction rules, we contributed to the automation of phishing URL detection tasks. It's important to note that despite the fact that using features from third-party services can make detection take longer, it actually makes detection more accurate [12]. They studied how well the proposed algorithm works on 1407 legitimate and 2119 phishing websites in the Alexa database3 and PhishTank2 respectively. Deep learning-based phishing detection has been the subject of extensive research due to the limitations of rule-based feature selection and modeling in terms of generalization performance to unobserved URLs [13]. Deep learning is a technique for fitting complex mapping functions that makes use of a large number of observations. The process of selecting features is automated using word-level features and variants based on recurrent neural networks [14, 15]. Muhammad et al. [16] proposed a brand-new self-structured NN for the purpose of identifying websites that are phishing. They assigned 17 signatures to 800 phishing and 600 legitimate websites taken from the archives of PhishTank and Millersmiles 4, a few of which were taken from third-party services. The power and generalizability of

neural networks in phishing detection are demonstrated by their experiments. They proposed a backpropagation-trained feedforward neural network for website classification in another work [17]. On the other hand, because it is established that language, as well as sentiment analysis, can be modeled from the sequences of characters that make up a string [18], the character-level features that make up URLs are selected as the key features. Since character sequence feature sets require less feature selection or preprocessing, deep learning-based research focuses on optimizing computation and structure. Using only client-side features, a machine learning-based approach for identifying phishing websites was proposed by Jain and Gupta [19]. Using 19 features extracted from URLs and source code, they evaluated their method on 2,141 phishing pages from PhishTank and Openfish, 1,918 legitimate pages from the Alexa database, and a number of online payment and banking sites by verifying the effect of data enhancement on the performance enhancement of virtual phishing URL generation by using generative adversarial networks (GANs) [20]. Although each of the aforementioned studies suggests a variety of features for detecting phishing websites, a few of these characteristics may not be sufficient to identify instances of phishing [21]. There has been little focus on selecting the best features for detecting phishing websites. Rajab suggests using correlated feature sets and information gain to identify phishing sites. The UCI repository's results show that 11 and 9 features were chosen by IG and CFS, respectively, with 30 features assigned to 11,055 samples. The efficacy of the classification based on the selected features was also evaluated using the data mining technique RIPPER. Bu and Cho [22] use an unsupervised learning approach to filter phishing attacks and find significant class imbalances in phishing URL classification. The authors Babagoli et al. [23] used a dataset that was similar and suggested using decision trees and wrapper methods to select features, which led to the selection of 20 features [24]. They utilize a novel metaheuristic-based nonlinear relapse calculation to assess phishing location execution. However, the feature selection methods used in these studies still rely on the data and necessitate user-specified thresholds that should be established. The classification algorithm's final performance is influenced by these thresholds, particularly when selecting features from out-of-sample training data in practice. According to research streams, Microsoft developed a deep learning model that better detects phishing attacks by utilizing character-level and word-level features [25]. Based on the enhancement of the URL feature set and deep learning operations of the self-attention mechanism, the most accurate and reliable of the currently available phishing detection methods is. Utilizing expert knowledge-based feature sets and character- and word-level URL features, Bu and Cho also improve performance [26]. First-order logic-based phishing attack detection rules successfully correct the output of deep learning classifiers and address the need to optimize the phishing detection feature set. Deep learning, conventional machine algorithms, and genetic algorithm-based combinatorial search have previously been combined for improved performance. Suleiman et al. [27] improved the accuracy of NB classifiers, k-NN classifiers, DT and RF classifiers by incorporating evolutionary computation-based feature selection algorithms into traditional machine-based algorithm-based phishing website detection tasks. Park et al. [28] improvement of discovery rules in light of hereditary calculation, amplifying the exactness and review of profound learning classifiers, and further developing identification execution.

3. Experimental Methodology

In this section, we talk about the proposed experiment in which a number of machine learning classifiers were used before and after the feature selection process [29]. There are six feature selection methods in the interval between before and after feature selection. The optimal number of features was selected by each of the feature selection algorithms. The proposed method's overall architecture, which compares the results obtained before and after feature selection, as well as the number of features selected by each algorithm, is depicted in Figure 2, which shows the architecture in its entirety. In this section, all ML classifiers have been described in brief as well as all feature selection algorithms.

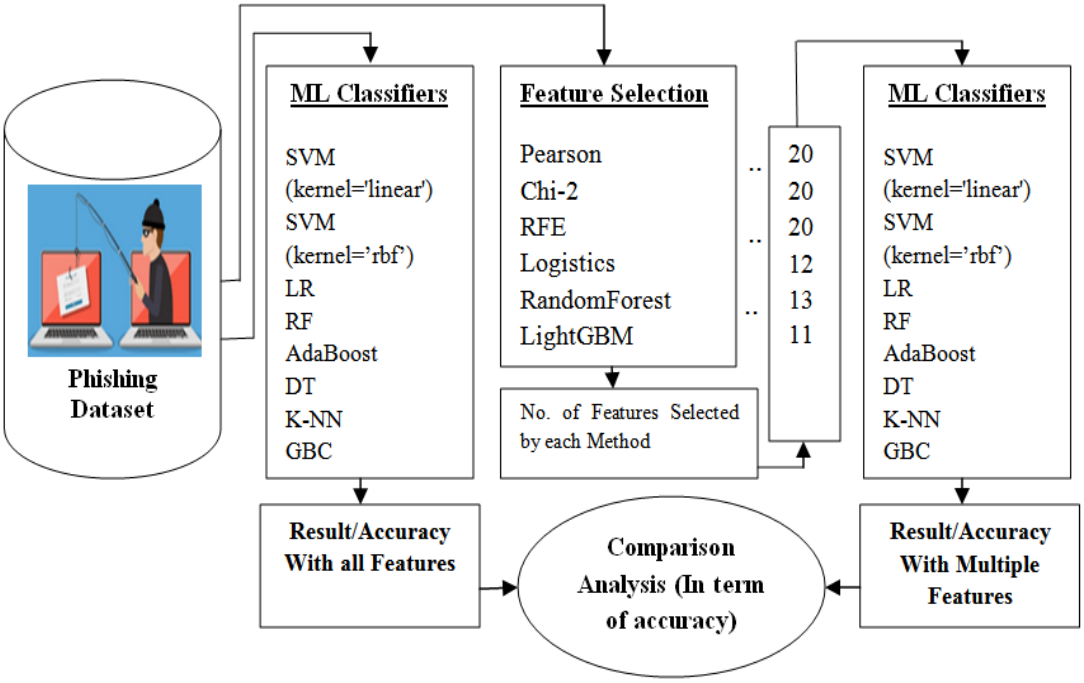

Fig.2. Experiment's flow diagram

3.1. Machine Learning Classifiers

Self-teaching algorithms are the focus of the artificial intelligence (AI) subfield known as machine learning. In machine learning, professionals employ a wide range of algorithms, including classifiers [30]. There are two main models in the classifier category: both with and without supervision. Classifiers learn to distinguish between unlabeled and labeled data in the supervised model. They are able to recognize patterns as a result of this training and, in the end, function independently without the use of labels. Pattern recognition is used by unsupervised algorithms to classify unlabeled datasets with increasing precision [31]. By automating the analysis and classification steps, AI tools with classification functionality simplify this procedure.

3.1.1. Support vector machine (SVM)

Finding a hyperplane or N features in an N-dimensional space that clearly classifies the data points is the goal of the SVM algorithm [32]. There are numerous possible hyperplanes from which the two classes of data points can be distinguished. Our goal is to locate a plane with the greatest margin, or distance between data points from both classes. In order to classify subsequent data points with greater confidence, increasing the margin distance to its maximum provides some reinforcement. However, a kernel trick emerges in problems involving non-linear SVMs. In higher-dimensional space, a kernel is a function that converts the original nonlinear problem into a linear one.

3.1.2. Logistic Regression (LR)

In machine learning, a classification method called logistic regression is utilized [33]. Logistic functions are used to model the dependent variable. Because the dependent variable is dichotomous, there are only two possible categories. Logistic regression is included in supervised learning. Supervised learning occurs when an algorithm uses a labeled dataset to learn and analyze the training data. There are inputs and anticipated outputs in these labeled datasets. Supervised learning also includes regression and classification.

3.1.3. Random Forest (RF)

In machine learning, random forest is a supervised learning method for classification and regression algorithms. It is a classifier that improves the dataset's predicted accuracy by averaging the results of several decision trees applied to various subsets of the dataset [34]. It makes a "forest" from a collection of decision trees that are typically trained by "bagging." The fundamental premise of bagging techniques is that output can be enhanced by combining multiple learned models.

3.1.4. AdaBoost

Adaboost, or Adaptive Boosting, is a 1996 ensemble boosting classifier proposed by Yoav Freund and Robert Schapire. To improve classifier accuracy, it combines multiple classifiers. AdaBoost is a method for iterative ensembles [35]. By combining multiple underperforming classifiers, the AdaBoost classifier creates a powerful classifier with high accuracy. The basic idea of Adaboost is to train data samples and set the weights of classifiers in iteration so that they can accurately predict abnormal observations.

3.1.5. Decision Tree (DT)
Decision trees, a type of supervised machine learning, involves successively partitioning data based on particular parameters [36]. Two entities that can be used to interpret a tree are the decision nodes and the leaves. Leaves are used to represent decisions or outcomes. Additionally, decision nodes split the data.

3.1.6. 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 [37]. 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. Check out our in-depth explanation of the principles of supervised learning for a better understanding of how it works. The model's training step does not include any parameter fine-tuning because it is non-parametric. K is a hyperparameter, but it can be thought of as an algorithm parameter in some way. It is chosen by hand and stays the same during training and inference. Also non-linear is the k-nearest neighbor algorithm. 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.

3.1.7. Gradient boosting classifier (GBC)
In Gradient Boosting, each predictor tries to make its predecessor better by lowering the error. Gradient Boosting, on the other hand, fits a new predictor to the residuals of previous predictors rather than fitting a predictor to the data at iteration [38]. This is an intriguing concept. In order to make an initial prediction based on the data, the algorithm will determine the logarithm of the probability of the target feature. This is typically determined by dividing the number of true values by the number of false values.

3.2. Feature Selection Algorithms
In machine learning, feature selection removes features that are redundant, noisy, or irrelevant to select the most relevant subset of the original set. Six distinct feature selection methods are utilized to select the most prominent and relevant features in order to enhance the classifier's accuracy [39]. Table 1 lists all of the features chosen using various feature selection methods.

3.2.1. Pearson correlation
Using Pearson Correlation, a correlation matrix is created that measures the linear association between two features and provides values between -1 and 1 for the degree of correlation [40]. It computes the association between each feature and the target variable to determine the degree to which two features are dependent on one another. The feature with the greatest impact on the target can be identified.

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

Where n is the number of records in the dataset, x is the average value of the sample attribute, x is the i^{th} value of the variable, and y is the target variable. 1 indicates a correlation, -1 indicates a correlation, and 0 indicates no correlation.

3.2.2. Ch-2
The chi-2 test was used to verify the independence of attributes in statistical models [41]. 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

3.2.3. 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 [42]. 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.

3.2.4. Logistic Regression (LR)

Logistic regression seeks to establish a connection between characteristics and the likelihood of a particular outcome. The only difference between a Logistic Regression model and a Linear Regression model is that, in place of a linear function, the Logistic Regression model makes use of a more complex cost function known as the Sigmoid function or logistic function [43]. The term "logistic regression" can be,

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

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.

3.2.5. Random Forest (RF)

A supervised model called Random Forest employs both decision trees and bagging [44]. 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.

3.2.6. LightGBM

A gradient boosting framework called Light GBM makes use of a tree-based learning algorithm [45]. 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.

4. Experimental Setup

The used dataset comes from the Kaggle Repository's Phishing website dataset [46]. There are 32 features in the phishing dataset; the feature with the name Index has been removed because it only contains serial numbers. Table 2 shows that of the 31 features, there are 30 independent features and 1 dependent feature. The Result is the final feature, indicating whether the website is phishing (1) or legitimate (0). There are 4898 legitimate websites and 6157 phishing websites, as depicted in Figure 3.

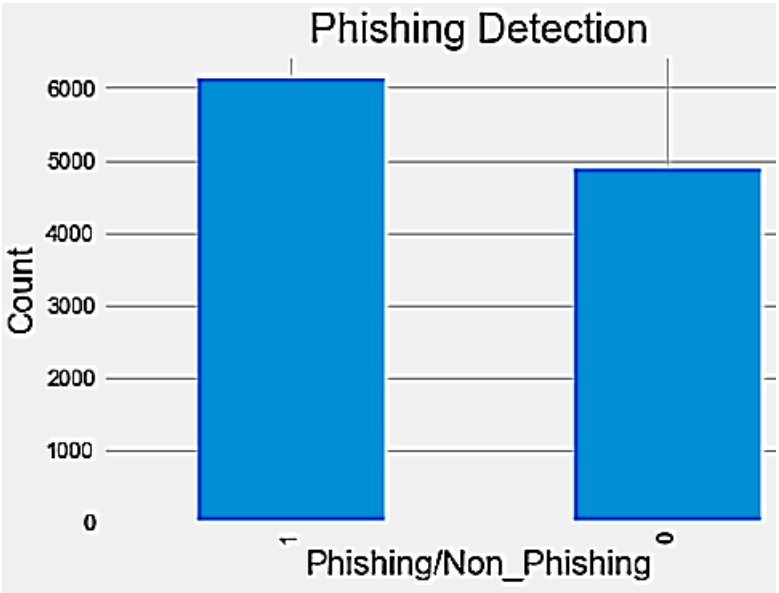

Fig.3. Phishing and legitimate websites

5. Result

The results presented in this manuscript are based on both before and after feature selection. By comparing these results, we can see if the result based on fewer features could be better than the result based on all features. First, we discuss results based on all features, such as (f1, f2….f30), as shown in Table 3. The accuracy, recall, precision, f1-score, and confusion matrices, as well as the correlation matrix of features and ROC curve analysis, formed the basis for these actual results [47].'

5.1. Result based on before feature selection

A correlation matrix is first constructed between the coefficients of various variables [48]. In order to summarize a phishing dataset and identify and visualize patterns in the provided data, the matrix illustrates the correlation between all 31 pairs of feature values in a table. The variables are displayed in rows and columns in each feature. The correlation coefficient can be found in any cell in a table. Additionally, other kinds of statistical analysis are frequently used in conjunction with the correlation matrix. Figure 4 shows that the ranks of the 12-features f5, f4, f1, f9, f2, f11, f6, f19, f8, f7, f16, and f18 are highly correlated.

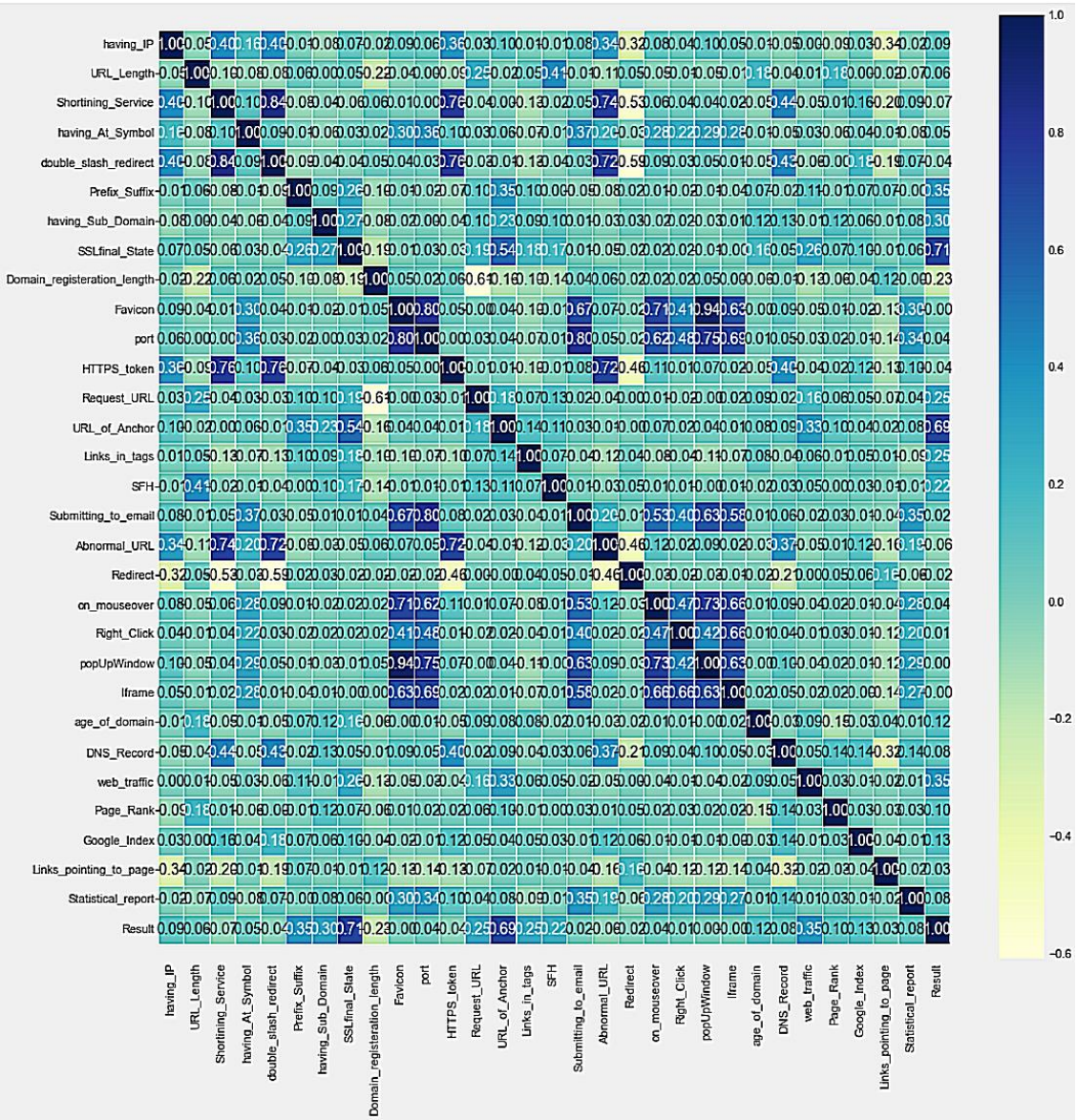

Fig.4. Correlation matrix

On our dataset with all features, we used various machine learning classifiers in the subsequent step. As previously stated, an assorted classifier was used to predict the accuracy of the classifiers using the dataset. A number of experiments involving machine learning-based classification based on our dataset's features are presented in Table 4. The dataset is divided using a machine learning technique for evaluating and comparing learning algorithms. Training accounts for 80% of the dataset, while testing accounts for 20%. K-fold cross-validation validates the dataset. The dataset goes through the testing phase after being trained using various machine learning classifiers. At this point, various machine learning algorithms are also applied to the

particular data. The distinction in our instance was made between phishing and non-phishing website URLs. The dataset performed well in comparison to the eight machine learning classifications. That was the first stream experiment that utilized before feature selection to obtain results from simple classification. In this instance, the accuracy-based result with both RF and DT had the highest accuracy on the test dataset—96.06 percent—so we can call it a tie.. Table 4 depicts the training and testing outcomes based on various classifiers. Figure 5 depicts the corresponding outcome.

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

Accuracy	**SVM (kernel='linear')**	**SVM (kernel='rbf')**	**LR**	**RF**	**AdaBoost**	**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 %

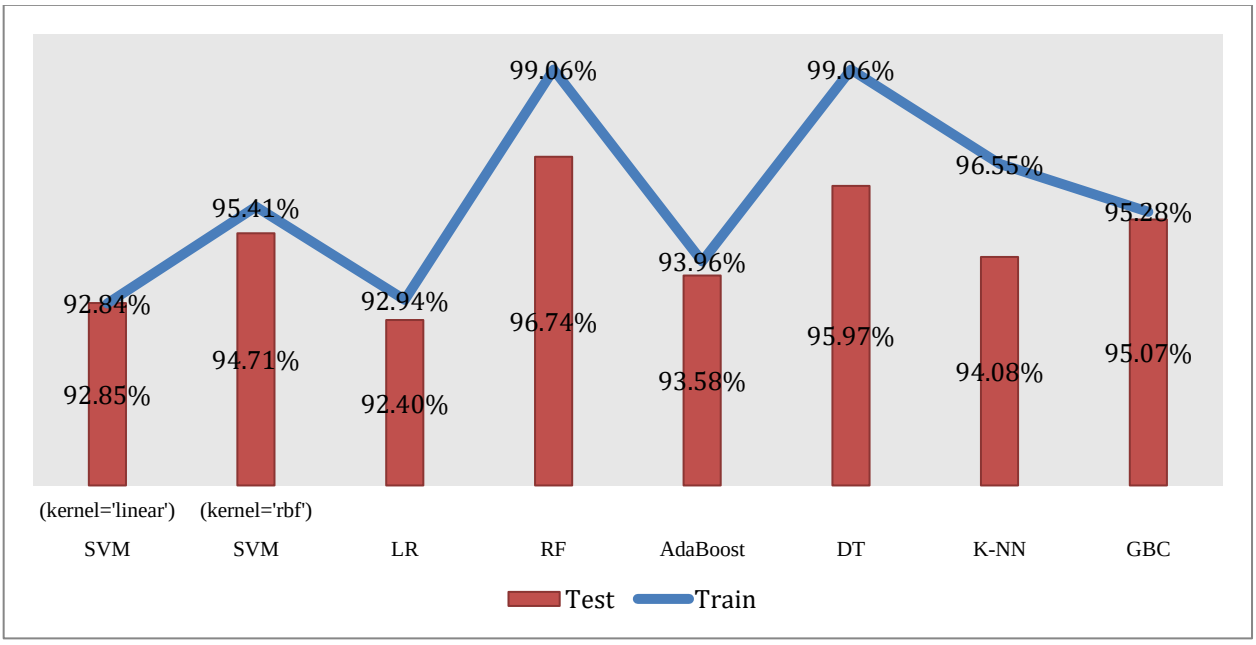

Fig.5. Visualization of classifier accuracy across all features during training and testing

The effectiveness of the proposed model was then assessed with the help of four well-known validation techniques. These performance metrics are listed in Table 5. We can use precision measure to evaluate the ratio of correctly predicted observations to positive observations in our experiments. The recall test looks at the proportion of correctly predicted positive observations to all actual class observations. The F-measure is a precision and recall weighted average. For the purposes of measuring Recall, Precision, Specificity, Accuracy, and AUC-ROC curves, the confusion matrix is a table with four distinct combinations of predicted and actual values.

Table5. 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

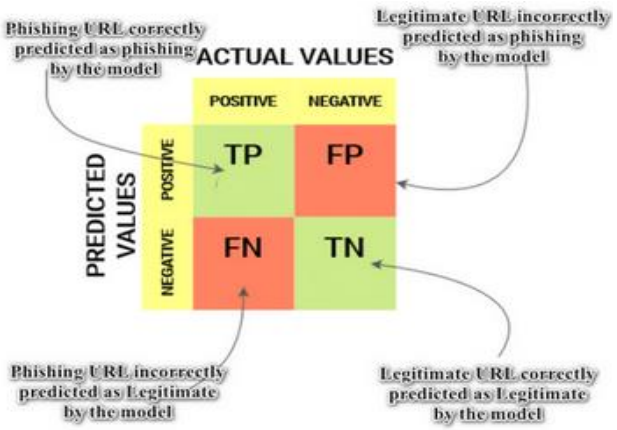

Table 6 displays the phishing and legitimate URL precision, recall, and f1 scores for the training and testing datasets. The training and testing datasets have also had the confusion matrix score extracted. On the test dataset, the only winner is the RF classifier, with precision, recall, and f1-score of 96.31 percent, 98.00 percent, and 97.15 percent, respectively. The validation score is very similar to RF when compared to DT. The confusion matrix-based result is also very close to DT.

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	[[3573369] [2644638]]	[[86195] [631192]]
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	[[3685257] [1494753]]	[[88274] [431212]]
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	[[3579363] [2614641]]	[[86492] [761179]]
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]]	[[90947] [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	[[3624318] [2164686]]	[[87482] [601195]]

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]]	[[91046] [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]]	[[88472] [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]]	[[89165] [44 1211]]

An aggregate measure of performance across all possible classification thresholds is provided by the ROC (AUC) curve. As shown in the preceding results, accuracy, precision, recall, the F1-score, and the confusion matrix all have very close scores; consequently, we require additional clarification regarding results based on these metrics. The Figure 6 is calculated using these metrics. RF has a higher ROC (AUC) score than DT.

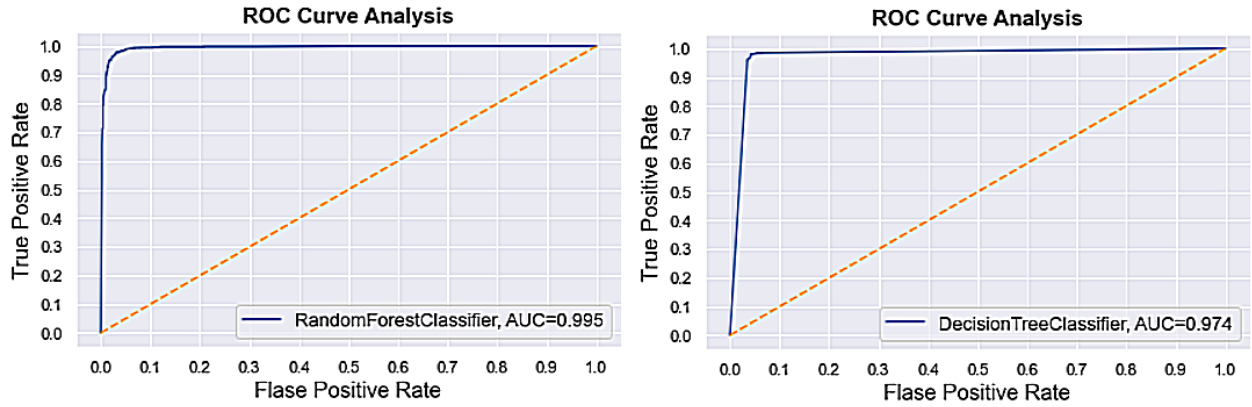

Fig.6. RF and DT ROC (AUC) curves

5.2. Result based on after feature selection

In a wide range of applications, the feature selection algorithms have received increasing attention. Using these algorithms, simulate "survival of the fittest" evolution to search the solution space. The score obtained by various feature selection algorithms on various numbers of features is shown in Table 7 from the simulation result. Multiple scores are produced by the eight classifiers based on their training and testing results (accuracy) on fewer features. When these scores are compared, we can see that the Ch-2 feature selection algorithm gave RF the highest testing accuracy—96.99%—at 20 numbers of features. RF classifier, on the other hand, achieves the second highest score (96.25 percent gain) when using the same number of features, 20.

Table7. 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

Figure 7 is the conclusion of the data presented in Table 7, which provides a summary of the previous findings.

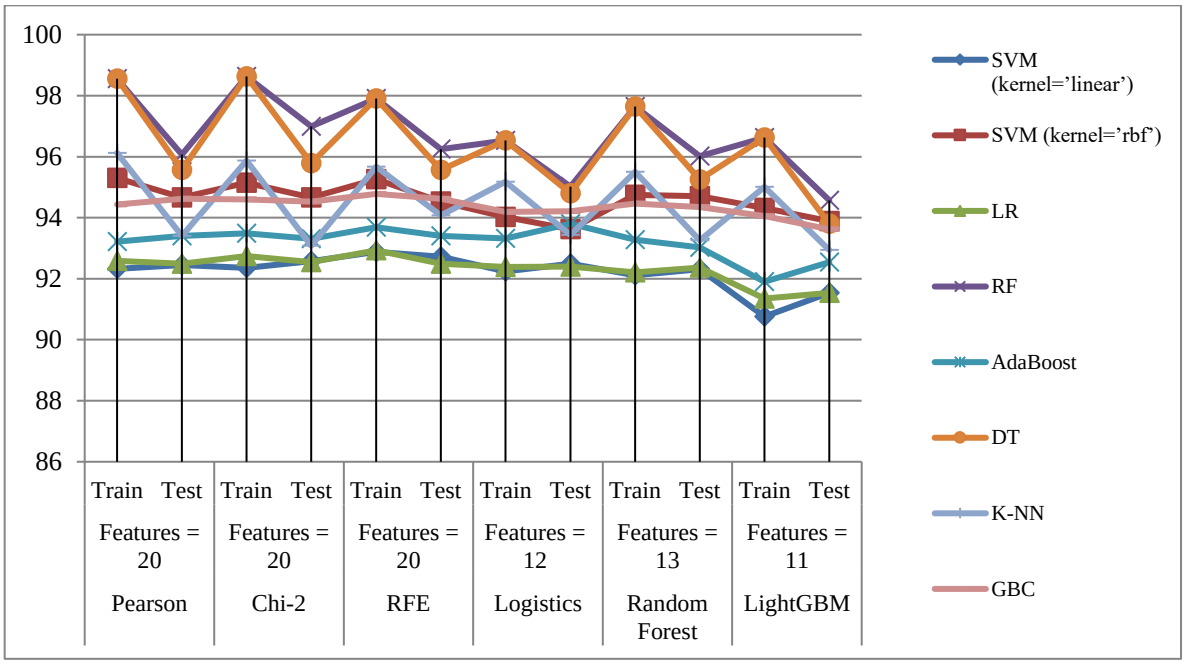

Fig.7. Accuracy of the classifiers for various feature counts

According to Table 7, the testing accuracy of RF is the highest when all of the classifiers with varying numbers of features are compared. As can be seen in Table 6, RF is the high-scoring classifier, so we created Table 8, which shows the concentric result based on RF at various features and a comparison with all features (WFS). In addition, a comparison is made in Figure 8 to show the classifiers' testing accuracy across all features and different numbers of features.

Table8. 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

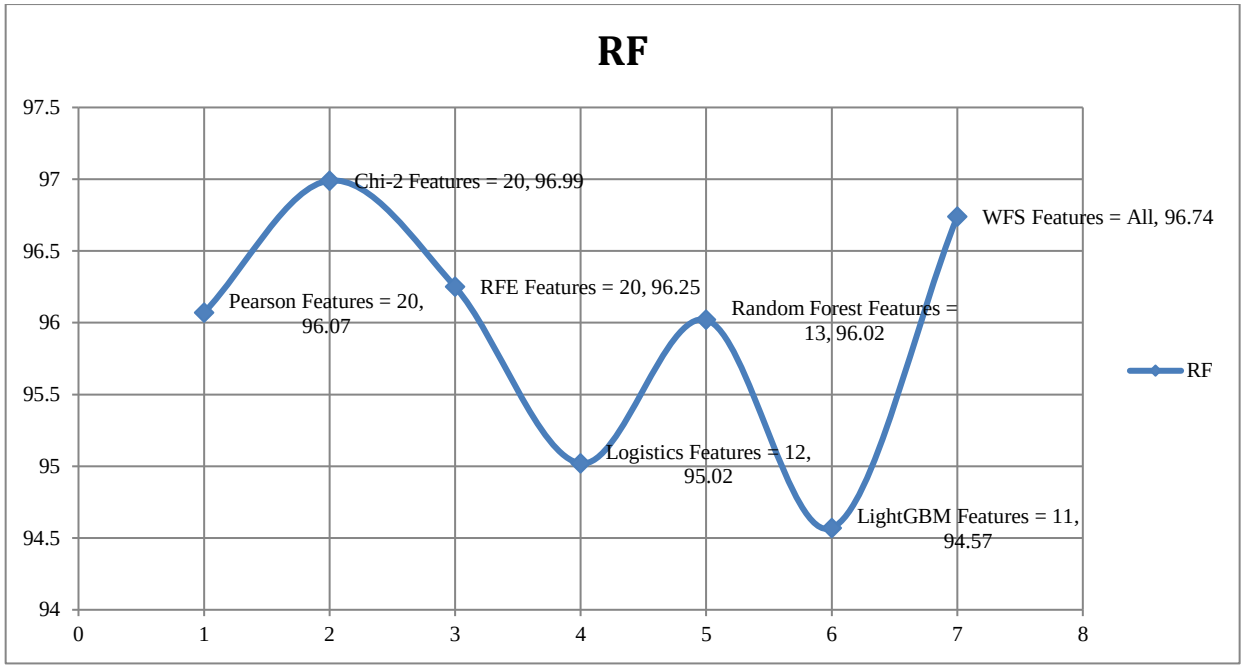

Fig.8. A test of the accuracy of the classifiers with all features and different numbers of features

6. Discussion

The goal of the feature selection process in this study was to pick the best URL-based features from Table 1. We used Pearson, Chi-2, Logistics, Random Forest, Light GBM, and RFE as feature selection algorithms at this point. Utilizing feature selection algorithms allowed for an improvement in detection accuracy. When employing SVM (linear and rbf), LR, RF, AdaBoost, DT, K-NN, and GBC as classifiers, the feature selection algorithms select the features listed in Table 3 automatically. As shown in Table 3, Pearson, Ch-2, and RFE each selected a maximum of 20 features. While Light GBM, Random forest, and Logistic each selected 12, 13, and 11 features. Estimates of the training and testing accuracies (Tables 4 and 5) were calculated using the phishing dataset with its entire features, precision, recall, f1-score, and confusion matrix (Table 6). In terms of accuracy, RF performed better than any other participant in this experiment (96.74 percent), and the validation metrics also performed well for RF. Additionally, we utilized six feature selection algorithms, each of which, when applied, selected distinct features as shown in Table 3. The Pearson, Chi-2, and RFE feature selection methods have chosen a maximum of 20 features from a total of 30 in Table 7. When compared to seven other classifiers, Pearson, Chi-2, and RFE have testing accuracy of 96.07%, 96.99%, and 96.25%, respectively. Table 7 shows that in this experiment with the phishing dataset, only the RF classifier with 20 features selected using the Chi-2 feature selection method performed better. The Chi-2 method selects the features f1, f2, ..., f12 and f14, f15, ..., f21, i.e. 20 features in total. In the final step, we compared the RF classifier's results for a variety of features to those for the entire phishing dataset (see Table 8). The RF classifier has the highest accuracy, at 96.94% with all 31 features and 96.99% with 20 features. Figure 8 depicts the comparative diagram produced by our experiment.

We compared the proposed model to other machine learning models that are being studied at the moment. The proposed model improves the detection system's accuracy, as shown by the obtained results. In Table 9, the creators presented a phishing recognition model by using highlight determination and consolidating as a pre-handling step for the dataset. By utilizing feature selection classifiers and reducing the number of selected features in our proposed work, we attempted to maximize accuracy.

Table9. 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. [49]	Hybrid Ensemble Feature Selection (HEFS), Cumulative Distribution Function gradient (CDF-g) algorithm	48 with dataset 1	96.17%	NR	NR	NR
Chiew et al. [49]	Hybrid Ensemble Feature Selection (HEFS), Cumulative Distribution Function gradient (CDF-g) algorithm	10 with dataset 1	94.60%	NR	NR	NR
Chiew et al. [49]	Hybrid Ensemble Feature Selection (HEFS), Cumulative Distribution Function gradient (CDF-g) algorithm	30 with dataset 2	94.27%	NR	NR	NR
Chiew et al. [49]	Hybrid Ensemble Feature Selection (HEFS), Cumulative Distribution Function gradient (CDF-g) algorithm	5 with dataset 2	93.22%	NR	NR	NR
(Zhu et al. [50]	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

7. Conclusion and Future Work

Website phishing is an effective attack that can lead to the disclosure and unauthorized use of sensitive information by Internet users. Phishers aim to steal sensitive information from naive users, such as usernames and passwords, bank account information, and credit card numbers. For the purpose of detecting spoofed websites, we have identified and examined the most critical features in this article. In order to select the features that are most useful for detecting website phishing, we suggested six feature selection strategies. Furthermore, we propose a phishing attack detection strategy based on eight machine learning algorithms, where the RF classifier achieves the highest accuracy for all features and less. Our phishing detection method can classify phishing websites in real time and deliver superior results to those of the existing methods. In subsequent work, we will expand our approach from URL-only to webpage content-based techniques in order to be able to inspect and analyze webpage data after the webpage has been rendered and downloaded to the user's computer. We believe that combining technologies based on web content and URLs will add an additional layer of security.

Funding: There is no financial support for this work.

Conflicts of Interest: Authors have disclosed no competing interests.

References

[1] Franjić, S. (2020). Cybercrime is Very Dangerous Form of Criminal Behavior and Cybersecurity. Emerging Science Journal, 4, 18-26.
[2] Srinivas, J., Das, A. K., & Kumar, N. (2019). Government regulations in cyber security: Framework, standards and recommendations. Future generation computer systems, 92, 178-188.
[3] Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020, June). Feature selections for the classification of webpages to detect phishing attacks: a survey. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-9). IEEE.
[4] Oest, A., Safei, Y., Doupé, A., Ahn, G. J., Wardman, B., & Warner, G. (2018, May). Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In 2018 APWG Symposium on Electronic Crime Research (eCrime) (pp. 1-12). IEEE.

[5] Gupta, S., Cherukuri, A. K., Subramanian, C. M., & Ahmad, A. (2022). Comparison, Analysis and Analogy of Biological and Computer Viruses. In Intelligent Interactive Multimedia Systems for e-Healthcare Applications (pp. 3-34). Springer, Singapore.
[6] Zhong, C., & Sastry, N. (2017). Systems applications of social networks. ACM Computing Surveys (CSUR), 50(5), 1-42.
[7] Awasthi, A., & Goel, N. (2021). Phishing Website Prediction: A Machine Learning Approach. In Progress in Advanced Computing and Intelligent Engineering (pp. 143-152). Springer, Singapore.
[8] Awasthi, A., & Goel, N. (2021, January). Generating Rules to Detect Phishing Websites Using URL Features. In 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON) (pp. 1-9). IEEE.
[9] Le, A.; Markopoulou, A.; Faloutsos, M. Phishdef: Url names say it all. In Proceedings of the 2011 Proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 191–195.
[10] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han, and J. Wang, "The application of a novel neural network in the detection of phishing websites," Journal of Ambient Intelligence and Humanized Computing, 2018.
[11] Mohammad, R.M.; Thabtah, F.; McCluskey, L. 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, 10–12 December 2012; pp. 492–497.
[12] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," Expert Systems with Applications, vol. 117, pp. 345–357, 2019
[13] Iuga, C.; Nurse, J.R.; Erola, A. Baiting the hook: Factors impacting susceptibility to phishing attacks. Hum.-Cent. Comput. Inf. Sci. 2016, 6, 8. [CrossRef]
[14] Bahnsen, A.C.; Bohorquez, E.C.; Villegas, S.; Vargas, J.; González, F.A. Classifying phishing URLs using recurrent neural networks. In Proceedings of the 2017 APWG Symposium on Electronic Crime Research (eCrime), Scottsdale, AZ, USA, 25–27 April 2017; pp. 1–8.
[15] Zhao, J.; Wang, N.; Ma, Q.; Cheng, Z. 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, 3–5 July 2018; pp. 385–394.
[16] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, no. 2, pp. 443–458, 2014.
[17] R. Mohammad, T. McCluskey, and F. A. Thabtah, "Predicting phishing websites using neural network trained with back-propagation." World Congress in Computer Science, Computer Engineering, and Applied Computing, 2013.
[18] 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, 7–12 December 2015; pp. 649–657.
[19] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," Telecommunication Systems, vol. 68, no. 4, pp. 687–700, 2018.
[20] Anand, A.; Gorde, K.; Moniz, J.R.A.; Park, N.; Chakraborty, T.; Chu, B.-T. 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, 10–13 December 2018; pp. 1168–1177.
[21] M. Rajab, "An anti-phishing method based on feature analysis," in Proceedings of the 2nd International Conference on Machine Learning and Soft Computing. ACM, 2018, pp. 133–139.
[22] Bu, S.-J.; Cho, S.-B. Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection. Electronics 2021, 10, 1492. [CrossRef]
[23] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," Soft Computing, pp. 1–13, 2018.
[24] Le, H.; Pham, Q.; Sahoo, D.; Hoi, S.C. URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv 2018, arXiv:1802.03162.
[25] Tajaddodianfar, F.; Stokes, J.W.; Gururajan, A. 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, Spain, 4–8 May 2020; pp. 2857–2861.
[26] Bu, S.-J.; Cho, S.-B. Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection. Electronics 2021, 10, 1492. [CrossRef]
[27] Suleman, M.-T.; Awan, S.M. Optimization of URL-based phishing websites detection through genetic algorithms. Autom. Control. Comput. Sci. 2019, 53, 333–341. [CrossRef]
[28] Park, K.-W.; Bu, S.-J.; Cho, S.-B. Evolutionary optimization of neuro-symbolic integration for phishing URL detection. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Bilbao, Spain, 22–24 September 2021; pp. 88–100.
[29] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300, 70-79.
[30] Awasthi, A., & Goel, N. (2021). Phishing Website Prediction: A Comparison of Machine Learning Techniques. In Data Intelligence and Cognitive Informatics (pp. 637-650). Springer, Singapore.
[31] Chaurasia, V., & Chaurasia, A. (2023). Novel Method of Characterization of Heart Disease Prediction Using Sequential Feature Selection-Based Ensemble Technique. Biomedical Materials & Devices, 1-10.

[32] Taher, S. A., Akhter, K. A., & Hasan, K. A. (2018, September). N-gram based sentiment mining for bangla text using support vector machine. In 2018 international conference on Bangla speech and language processing (ICBSLP) (pp. 1-5). IEEE.
[33] 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.
[34] Josephine, P. K., Prakash, V. S., & Divya, K. S. (2021). Supervised Learning Algorithms: A Comparison. Kristu Jayanti Journal of Computational Sciences (KJCS), 01-12.
[35] 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.
[36] 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.
[37] Gøttcke, J. M. N., Zimek, A., & Campello, R. J. (2021, September). Non-parametric semi-supervised learning by Bayesian label distribution propagation. In International Conference on Similarity Search and Applications (pp. 118-132). Springer, Cham.
[38] 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.
[39] Awasthi, A., & Goel, N. (2022). Phishing website prediction using base and ensemble classifier techniques with cross-validation. Cybersecurity, 5(1), 1-23.
[40] 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.
[41] 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.
[42] 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.
[43] 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.
[44] Alsouda, Y., Pllana, S., & Kurti, A. (2019, May). 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).
[45] 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.
[46] Phishing website dataset | Kaggle, https://www.kaggle.com/datasets/akashkr/phishing-website-dataset?select=dataset.csv. Accessed 8th January 2023.
[47] Chaurasia, V., Pandey, M. K., & Pal, S. (2022). Chronic kidney disease: A prediction and comparison of ensemble and basic classifiers performance. Human-Intelligent Systems Integration, 1-10.
[48] Qiu, P., & Niu, Z. (2021). TCIC_FS: Total correlation information coefficient-based feature selection method for high-dimensional data. Knowledge-Based Systems, 231, 107418.
[49] 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.
[50] 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.