Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Employing cluster-based class decomposition approach to detect phishing websites using machine learning classifiers

Yousif Al-Tamimi[a] and Mohammad Shkoukani[a*]

[a]*Faculty of Information Technology, Applied Science Private University, Amman, Jordan*

C H R O N I C L E

Article history:
Received: April 10, 2022
Received in revised format: September 20, 2022
Accepted: October 8, 2022
Available online: October 8 2022

Keywords:
Phishing website
Machine learning
Class decomposition
Classification

A B S T R A C T

Phishing is an attack by cybercriminals to obtain sensitive information such as account IDs, usernames, and passwords through the use of the anonymous structure of the Internet. Although software companies are launching new anti-phishing tools that use blacklists, heuristics, visual methods, and machine learning-based methods, these products cannot prevent all phishing attacks. This research offers an opportunity to increase accuracy in the detection of phishing sites. This study develops a model using machine learning algorithms, specifically the decision tree and the random forest, due to their outperforming the rest of the classifiers and being accredited by researchers in this field to achieve the highest accuracy. The study is based on two phases: the first phase is to measure the accuracy of classifiers on the dataset in the usual way before and after feature selection. The second phase uses the class decomposition approach and measures the accuracy of classifiers in the dataset before feature selection and after feature selection to detect phishing sites. The class decomposition approach is a technique to improve the performance of classifiers by distributing each class into clusters and renaming the examples of each cluster with a new class. This provides a specific metric that more accurately predicts the level of phishing. Testing on a dataset containing 11,055 instances, 4,898 phishing, and 6,157 legitimate, each instance has 30 features. It achieved the highest accuracy in the first phase through the random forest algorithm by 96.9% before feature selection, and after feature selection, it was by 97.1%. In the second phase, the highest accuracy of both the decision tree and random forest classifiers was achieved by 100% with the two and four classes after feature selection. While before feature selection, the random forest algorithm achieved 100% with only the two classes.

1. Introduction

Phishing websites are one of the most difficult challenges that uneducated and inexperienced users face (McAnulty, 2021). Internet users cannot distinguish between legitimate websites and scam websites because scam websites are faked by attackers; fake websites look legitimate and genuine to users. When cloning the contents of any legitimate company, organization, or bank, scammers try to maintain similarities between the original and fake websites to deceive internet users. Phishing websites are not necessarily similar to a specific site but can be a new domain. When scammers send emails to their victims, including phishing website links, phishing websites prompt users to enter private data, and the scammers receive the victims' data and use this data for their illegal benefit. Most scammers do this for financial purposes by stealing the victim's account (Bitaab et al., 2020). Phishing attacks became successful due to a lack of user awareness. Because phishing attacks prey on

* Corresponding author.
E-mail address: m.shkokani@asu.edu.jo (M. Shkoukani)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2022.10.002

users' vulnerabilities, it's tough to prevent them, but it's critical to enhance phishing detection tools (Deshpande et al., 2021). One effective solution to prevent a phishing attack is to integrate security features with web browsers to raise alerts when an Internet user visits a phishing website (Tang et al., 2021). Phishing may be a fraud framework that uses a combination of social design and personal data for the extra. It may be possible to open credit passwords from simple elements by assuming the salient features of an individual or a trusted company in electronic correspondence. Phishing uses spoof messages created to look basic and directed to start from real blue sources like money-related organizations, online business goals, etc., to lure customers to go to fake destinations through the joins offered on phishing sites (Bhardwaj et al., 2021).

2. Related work

Hannousse and Yahiouche (2021) presented a General Strategy for Creating Repeatable and Extensible Datasets for Website Phishing Detection. In this paper they adopt an improved classification of website phishing features and systematically selected a total of 87 commonly known features. Random Forest was the most predictive classifier, according to the results. Filter-based classification with the gradual removal of less important characteristics outperformed casing technologies by up to 96.83 percent. Chiew et al. (2019) proposed a framework for a machine learning-based phishing detection system. The baseline features correctly distinguish 94.6% of phishing and legitimate websites using only 20.8% of the available data.

Subasi and Kremic (2020) used a variety of machine learning algorithms, and an intelligent framework for spotting phishing sites. Different classification algorithms were used to classify websites as real or fraudulent, and to create an intelligent system to detect phishing sites. The performance of machine learning systems was evaluated using classification accuracy, F-measurement, and the area under the receiver operating characteristic (ROC) curves (AUC). According to the results of experiments, Adaboost with SVM outperformed all other classification algorithms, reaching the highest accuracy of 97.61%.

Yang et al. (2017, 2019) introduced a new technology based on a Non-Inverse Online Sequencing Extreme Learning Machine (NIOSELM). The Sherman Morrison Woodbury equation was used to avoid matrix inversion. The researchers worked to fully describe a website. The NIOSELM algorithm considered three classes of properties. The overall detection performance appeared to be somewhat superior to those of other methods, especially in terms of training speed and detection accuracy. Niu et al. (2017) proposed a Decision Tree and Optimal Features based Artificial Neural Network (DTOF-ANN). The standard K-medoids clustering technique was first improved by an iterative selection of beginning centres. The best feature selection method was then created using the newly constructed feature evaluation index. Somesha et al. (2020) suggested a new categorization approach based on the extraction of heuristic traits. The retrieved characteristics were divided into three categories: URL masking features, third-party-based features, and hyperlink-based features. Furthermore, the proposed approach has a 99.57% accuracy. The disadvantage is that, because this strategy relies on third-party features, the website's ranking is influenced by the speed of the third-party services. This model is similarly reliant on the training set's quality and quantity.

Gautam et al. (2018) employed a method called correlation data mining. They proposed a categorization system based on criteria for detecting phishing sites. They concluded that the correlation classification method is superior to all other algorithms due to its easy rule transformation. They extracted 16 characteristics and reached an accuracy of 92.67%, however, this falls short of specificity, thus the suggested technique may be modified to get a high detection rate.

Liu et al. (2018) focused primarily on character frequency features. They worked on integrating statistical analysis of IP addresses with machine learning techniques to obtain high-quality results that are more accurate for classifying and detecting harmful IP addresses. To prove the effectiveness of the proposed algorithm, it outperformed six machine learning algorithms with an accuracy of 99.7% with a positive rate and a false rate of less than 0.4%.

Jain and Gupta (2018) proposed a new method for detecting and combating phishing that collects information exclusively on the client side. The proposed method is quick and dependable since it does not rely on a third party and instead extracts the features directly from the URL and source code. In this study, they were able to detect phishing sites with an overall accuracy of 99.09%. This research indicates that this technique has limitations because it can only recognize HTML web pages and cannot detect non-HTML web pages. Shirazi et al. (2017) created the "Fresh-Phish" open-source framework. This system may be used to produce machine learning data for phishing sites. They utilized a smaller feature set and Python to create a syntactic query for this. They created a huge labeled data set and examined multiple machine learning classifiers against it. The results show that machine learning classifiers had a high level of accuracy, which examined the length of time it takes to train the model. Yadollahi et al. (2019) proposed a true anti-phishing system. Web pages are characterized by this feature-rich online machine learning technology. The approaches are based on discriminative characteristics. There is no third-party service, thus the solution is entirely dependent on the consumer. Yang et al. (2017) proposed multidimensional feature phishing learning. This threshold is fixed for reducing the time. They have the highest accuracy by using CNN-LSTM.

Kamal et al. (2018) employed machine learning with features extracted from URLs used for phishing detection. According to the APWG, phishing grew in 2014 as a result of the domain name's license and independence. On the Weka platform, the Naive Bayes algorithm is utilized to categorize phishing sites. the Naive Bayes algorithm, Decision Tree, and Random Forest may produce an accuracy of up to 97.08% by employing Stacking, Bagging, and Boosting along.

Almadhoor (2021) focused on co-developing phishing site detection accuracy. As a result, the feature elicitation algorithm and 14 were chosen, as well as the group learning technique, which was based on multiple voting and parallelism and used a range of classification models, such as Random Forest, logistic regression, and among other things, current phishing detection techniques (prediction models) have an accuracy rate of between 70 and 92.52%, according to the research. The accuracy rate of the suggested model could be higher than the existing approaches for detecting phishing sites, according to experimental simulation. Tyagi et al. (2018) recommended several machine learning approaches for determining if a website is phishing or not. Python is thought to be able to detect 30 phishing site features. The generalized linear model (GLM) and the generalized additive model (GAM) were used to calculate the accuracy, which employed a decision tree, Random Forest, to improve accuracy. The PhishMon framework was suggested by Niakanlahiji et al. (2018) for identifying phishing web pages with great accuracy, distinguishing between real and fraudulent web pages.

Gutierrez et al. (2018) proposed Create a Semi-Automated Phishing Classification Feature (SAFE-PC) to detect whether a web page is phishing or legitimate. Kaytan and Hanbay (2017), for detecting misleading websites, suggested an extreme machine learning classification technique. The categorization of phishing websites in this article was based on “URL request” and “Website redirect”. The 10-fold validation method was used to assess performance. Niu et al. (2017) for high-accuracy email phishing detection, used the Cuckoo-Search SVM (CS-SVM) model. Increases the accuracy of categorization. CS-SVM was used to extract 23 characteristics to develop a hybrid classifier. Cuckoo-Search (CS) was used with Vector Machine help in the hybrid classifier to enhance parameter selection for the Radial Basis function (RBF). It achieved greater accuracy than an SVM classifier using RBF in this case. The method proposed by Singh et al. (2015) used machine learning-based feature extraction for phishing detection. They employed the Adaline and Backpropagation algorithms, as well as SVM, to enhance web page recognition and ranking. Adaline was compared against SVM, which had a score of 99.14 percent, for a superior outcome. Verma and Gautam (2020) used a random forest model in detecting a website as dangerous or safe. Their model was based on many classifiers and the random forest model. The classification algorithms used were Random Forest, Decision Tree, J48, Support Vector Machines, Naive Bayesian, Neural Network, Logistic Regression, Lazy K Star, and the C4.5 algorithm.

3. Materials and methods

3.1. The Datasets

The Dataset for training: This is a hurdle that any researcher on this subject tackles. However, even though many articles on using data mining techniques to predict phishing sites have been published recently, a generally reliable training data set has yet to be published, possibly due to a lack of agreement in the literature about the specific features that characterize phishing sites, making it difficult to create a dataset that covers all possible features. According to the University of California, Irvine, essential qualities that have been demonstrated to be sound and useful in predicting phishing sites are highlighted (UCI) (Babagoli et al., 2018). It contains 11,055 instances, 4,898 phishing, and 6,157 legitimate, and each instance has 30 features.

The major characteristics of a phishing site are based on the elements it provides, which may be divided into four categories based on how well it functioned (Mohammad et al., 2013). They are features that are based on the address bar. This means that the term signifies that the address bar itself is displaying a potentially dangerous or fraudulent website. Sub-types such as utilizing an IP address in the address bar; long URL to disguise the suspicious section; URL shortening; the presence of the "@" sign in the URL; redirect with the "//" tag; and other elements that show in the address bar are among the things that can be learned about this category. The aberrant characteristics category is the second. Anomalies of many sorts, such as the request URL, Anchor URL, links in the meta>, script>, and link> tags, server form handlers sending information to e-mail, and so on, And the URL isn't what you'd expect. Redirecting websites, customizing the status bar, disabling right-click, and using popup and iframe redirection are all HTML and JavaScript-based features. The domain-based attributes category includes domain age, DNS records, website traffic, page rank, Google index, and other comparable qualities that can be used to identify phishing sites.

Just as (1) indicates that the site is legitimate, (-1) indicates that the site is phishing, and (0) indicates that the site is suspicious, such as the URL is classified as "Suspicious" because it has one subdomain or if the domain name in SFHs is different from the domain name of the webpage. As in the following Table 1.

Table 1
List of phishing website features.

phishing website features						
Type	No	Feature	Name	Description	Rule	Value
Address bar-based	1	IP address	Using IP	Having IP address in URL	If The Domain Part has an IP Address → Phishing Otherwise→ Legitimate	−1, 1
	2	URL length	Long URL	Long URL to hide the suspicious part	URL length < 54 → feature = Legitimate else if URL length ≥ 54 and ≤ 75 → feature = Suspicious Otherwise → feature = Phishing	−1, 0, 1
	3	Shortening service	Short URL	Using URL shortening services “TinyURL”	Tiny URL → Phishing Otherwise→ Legitimate	−1, 1
	4	@ Symbol	Symbol@	URL’s having @ symbol	URL Having @ Symbol → Phishing Otherwise → Legitimate	−1, 1
	5	“//” redirecting	Redirecting //	Having “//” within the URL path for directing	The Position of the Last Occurrence of "//" in the URL > 7 → Phishing Otherwise → Legitimate	−1, 1
	6	Prefix suffix	Prefix Suffix	Adding prefix or suffix separated by (-) to the domain	Domain Name Part Includes (-) Symbol → Phishing Otherwise → Legitimate	-1, 1
	7	Sub domain	Sub Domains	Sub domain and multi sub domain	Dots In Domain Part =1 → Legitimate Dots In Domain Par t = 2 → Suspicious Otherwise → Phishing	-1, 0, 1
	8	SSL final state	HTTPS	Existence of HTTPS and validity of the certificate	Use https and Issuer Is Trusted & and Age of Certificate ≥ 1 Years → Legitimate Using https and Issuer Is Not Trusted → Suspicious Otherwise→ Phishing	-1, 0, 1
	9	Domain registration	DomainRegLen	Expiry date of domains/Domain registration length	Domains Expires on ≤ 1 years → Phishing Otherwise→ Legitimate	-1, 1
	10	Favicon	Favicon	Favicon loaded from a domain It's a visual reminder of the website's identity.	Favicon Loaded From External Domain→ Phishing Otherwise→ Legitimate	-1, 1
	11	Port	NonStdPort	Using non-standard port	Port is of the Preferred Status → Phishing Otherwise → Legitimate	-1, 1
	12	HTTPS token	HTTPS Domain URL	The existence of HTTPS token in the domain part of URL	Using HTTP Token in Domain Part of The URL → Phishing Otherwise → Legitimate	-1, 1
Abnormal-based	13	Request URL	Request URL	Request URL within a webpage/Abnormal request	Request URL < 22% → Legitimate Request URL ≥ 22% and 61%→ Suspicious Otherwise→feature=Phishing	-1, 1
	14	URL of anchor	AnchorURL	URL within <a> tag/Abnormal anchor	URL Of Anchor <31% → Legitimate% of URL Of Anchor ≥31% And≤67% → Suspicious Otherwise→ Phishing	-1, 0, 1
	15	Links in tags	LinksInScriptTags	Links in <Meta>, <Script> and <Link> tags	Links in "<Meta>"," <Script>"and"<Link>" <17% → Legitimate Linksin<Meta>","<Script>" and "<Link>" ≥17% And≤81% → Suspicious Otherwise→ Phishing	-1, 0, 1
	16	SFH	ServerForm-Handler	Server Form Handler	SFH is "about: blank" Or Is Empty → Phishing SFH Refers To A Different Domain→ Suspicious Otherwise → Legitimate	-1, 0, 1
	17	Email	Info Email	Submitting information to E-mail	Using "mail()" or "mailto:" Function to Submit User Information→ Phishing Otherwise → Legitimate	-1, 1
	18	Abnormal URL	Abnormal URL	Host name is included in the URL/Who is	The Host Name Is Not Included In URL→Phishing Otherwise→ Legitimate	-1, 1
HTML and JavaScript-based	19	Redirecting	Website Forwarding	Number of times a website has been redirected	Redirect Page ≤ 1 → Legitimate of Redirect Page ≥ 2 & And < 4 → Suspicious Otherwise → Phishing	−1, 0, 1
	20	On mouseover	Status Bar Cust	On mouse over changes status bar/Status bar customization	on Mouseover Changes Status Bar → Phishing It Doesn’t Change Status Bar → Legitimate	-1, 1
	21	Right click	Disable Right Click	Disable Right Click	Right Click Disabled → Phishing Otherwise → Legitimate	-1, 1
	22	Pop-up window	Using popup Window	Using Pop-up window	Pop-up Window Contains Text Fields→ Phishing Otherwise → Legitimate	-1, 1
	23	Iframe redirection	IframeRedirection	Using Iframe	Using iframe→ Phishing Otherwise → Legitimate	−1, 1
Domain-based	24	Age of domain	Age of Domain	Minimum age of a legitimate domain is 6 months	Age Of Domain ≥ 6 months → Legitimate Otherwise → Phishing	−1, 1
	25	DNS record	DNSRecording	Existence of DNS record for the domain	no DNS Record For The Domain → Phishing Otherwise→ Legitimate	−1, 1
	26	Website traffic	Website Traffic	Being among top 100,000 in Alexa rank	Website Rank < 100,000 → Legitimate Website Rank > 100,000 → Suspicious Otherwise → Phish	−1, 0, 1
	27	Page rank	PageRank	Having a page rank greater than 0.2	PageRank <0.2 → Phishing Otherwise → Legitimate	−1, 1
	28	Google index	Google Index	Website indexed by Google	Webpage Indexed by Google → Legitimate Otherwise → Phishing	−1, 1
	29	Link reference	LinksPoitingToPage	Number of links pointing to a page	Link Pointing to The Webpage = 0 → Phishing Link Pointing to The Webpage > 0 and ≤ 2 → Suspicious Otherwise → Legitimate	−1, 0, 1
	30	Statistical report	Status Report	Top 10 domain and top 10 Ips from Phish Tank	Host Belongs to Top Phishing IPs or Top Phishing Domains → Phishing Otherwise → Legitimate	−1, 1
	31	Result	class	Phishing or legitimate		−1, 1

3.2 Data Preprocessing

Outliers indicate extreme or unreasonable data samples that are very far from the rest of the group.

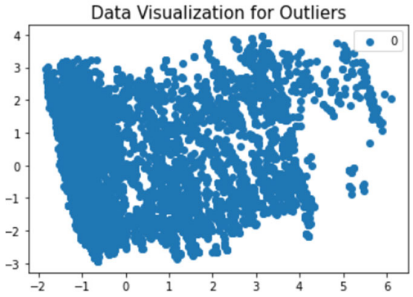


Fig. 1. Detecting outliers in our dataset

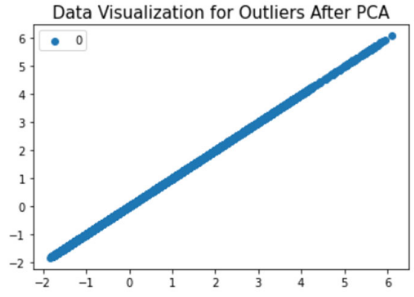


Fig. 2. The shape of our dataset after converting it to a vector

The values in our data set are shown to be close to and not far from the center, reasonable, and not outliers, as shown in Fig. 1 and Fig. 2, so outliers are not identified, thus we leave them as they are. As shown above, no outliers were found, so there is nothing to remove from the dataset at this stage.

3.3 Model and Dataset Selection

After finishing several procedures to test out the dataset and make sure that it is free of any failings, abnormalities, or noises. The data seems to be quite balanced, for the next step is to remove the "Id" column because it is not from the data set and then split the dataset into 70% for training and 30% for testing, which is the best combination after several attempts with different ratios for training and testing. The next step was to apply several machine learning algorithms to the prepared data set, and the best algorithms that gave the highest accuracy were selected from the experiments we conducted on the data set and from previous studies, namely: Random Forest and Decision Tree. The raw data set was passed to the two models by statistical representation. As shown in Fig. 3, (1-legit, negative 1-phishing).

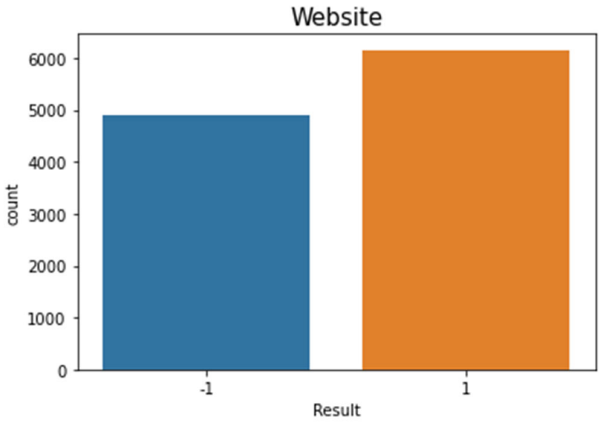


Fig. 3. Bar graph of the dataset used for website phishing. The dataset contains 6,157 legitimate and 4,898 phishing websites.

3.4 The Proposed Model

To achieve the objectives of the study, we designed the research methodology in three phases, as shown in Fig. 4: In the first phase, after selecting the dataset, which contains 30 features to determine whether the website is phishing or not, we adopted two machine learning classifiers, namely, the Decision Tree (DT) and the Random Forest (RF), which are among the most commonly used in this domain, to train and test them on the dataset. After that, the accuracy of the two models is measured using evaluation measures: accuracy, precision, and recall. To ensure the correctness of the algorithms' workflow, K-fold-10 was used for both classifiers and matching the results. Feature selection algorithms are applied to the entire dataset in proportion to this phase to give the best results. We tried the following models (XGBClassifier, Decision Tree Classifier, Random Forest Classifier, Logistic Regression, Ada Boost Classifier), and the expectations for the models for the best features were (3, 4, 6, 13, 11) respectively, and according to our tests, we found 25 features that give the best accuracy. The accuracy of the two models is also measured using evaluation metrics.

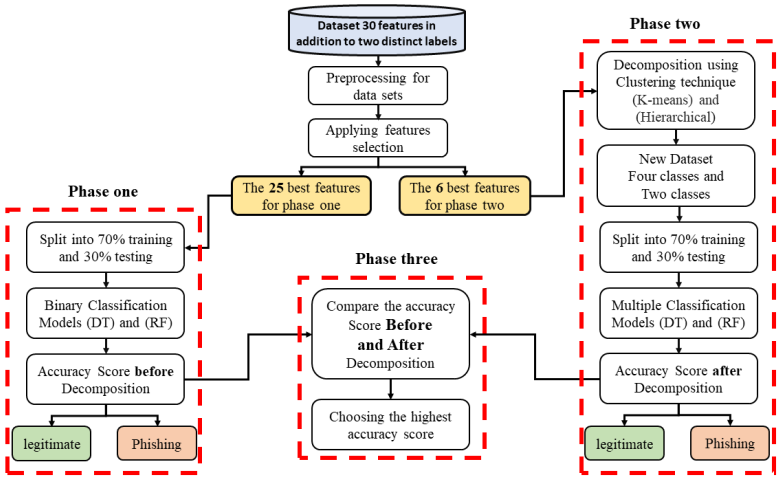


Fig. 4. Theoretical workflow and prediction methodology

The second phase, the effect of the class decomposition approach on the performance of machine learning models, is employed to improve the prediction accuracy of phishing websites. This approach uses different clustering techniques such as K-means clustering and hierarchical clustering. to produce a new data set of several classes that are compatible with a multi-label classification. Feature selection algorithms are applied as mentioned above to the dataset to discover the appropriate features at this phase to give the best results. And according to our tests, we found 6 features that give the best accuracy. The two models indicated will be trained, but this time they will work on the multiple classifications with the several new classes of data and be tested, and the accuracy of the models will be measured using the same evaluation measures, namely accuracy, precision, and recall. Finally, in the third phase, the accuracy results obtained in the first and second phases are compared to see if there is an improvement in the accuracy of the prediction or not, as this study assumes.

3.5 Feature selection

Feature selection has an important role in the analysis and training of data. The feature selection method helps in improving the accuracy of the prediction model so that it reduces the number of features that aren't of importance to those that are considered critical in influencing the prediction. On the other hand, this method leads to an increase in processing speed. Specifically, this method helps to narrow down the features of the raw dataset by keeping only the relevant and useful ones. Thus, the feature selection algorithm (Rao et al., 2019) will show which features rank high in feature importance and which do not. However, if the data is subject to feature selection, then the loss of information does not have an extreme effect.

Classification faces a feature selection problem. The reason we resort to feature selection algorithms is that the default significance contained within the two models is not always reliable. The most common mechanism for calculating feature import, which is the mechanism used in the Random Forest Classifier and Random Forest Regressor, is the average decrease in the impurity mechanism. about genetic significance. The average decrease in impurity significance for a feature is calculated by measuring how effectively the feature reduces uncertainty or variance for classifiers when decision trees are generated within the Random Forest Classifier. The problem is that this mechanism, while quick, does not always provide an accurate picture of the importance of the feature selection.

Where feature selection aims to identify a subset of highly variable features, in other words, different features that can distinguish samples belonging to different classes are identified. That is, for the problem of feature selection for a classification, given the availability of naming information, the importance of the features is evaluated as a distinction between different classes. For example, the feature Ai is said to be related to the class Bj if Ai and Bj are effectively related and from the heatmap to determine which features are most related to the result variable, as shown in Fig. 5.

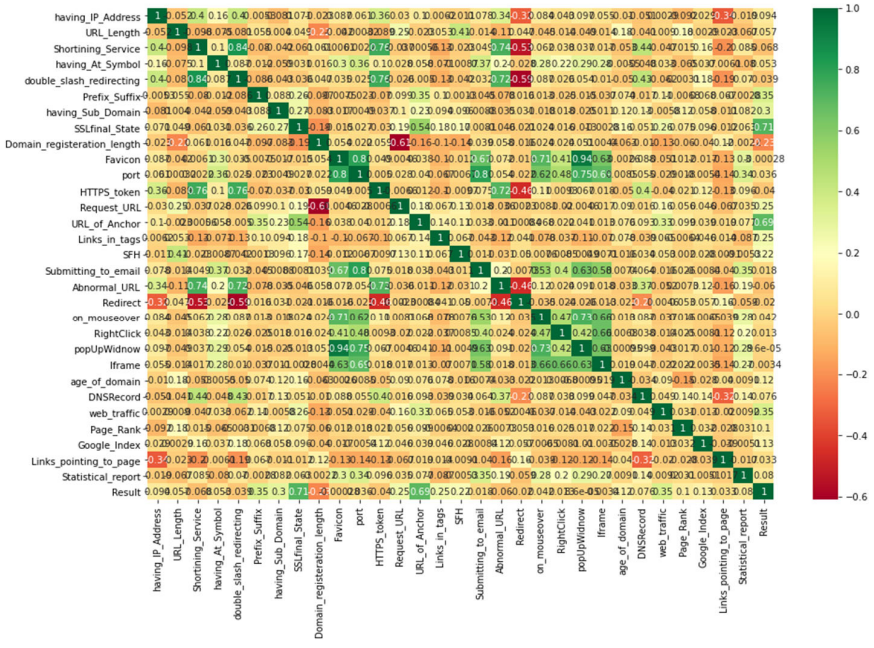
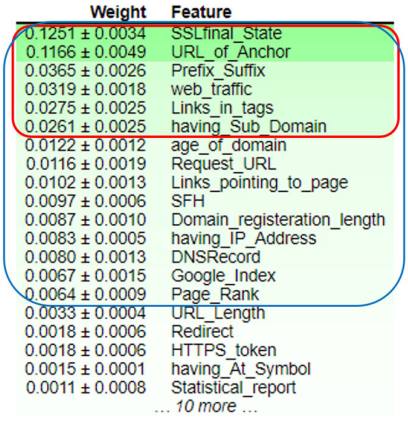


Fig. 5. Heatmap to identify which features are most related to the result variable

If we look at the last row, the output range above, we will see how the features are linked to the output range, where, 'SSLFinal_State', 'URL_OF_ANCHOR', 'are intensely associated with the output domain, while 'RightClick', 'popUpWidnow', 'Iframe' appears to be less connected to the output domain. This is one of the ways to determine the data-related features using a single variable selection technology and the importance of the feature and link matrix.

Classification algorithms, in all their forms, cannot always be applied directly to a data set. The data must be processed first. Preprocessing may include performing some functions and reducing dimensions as well.

The feature selection algorithms used in our outline are to identify the most significant features in the data set based on association with the result, and they are (eli5 PermutationImportance, SelectByShuffling, ExtraTreesClassifier, RandomForestClassifier, SelectFromModel) with different behaviors and methods of work, but the results of calculating the most important features of the data set for all of them are six features that take the lead, and they are ['Prefix_Suffix', 'having_Sub_Domain', 'SSLfinal_State', 'URL_of_Anchor', 'Links_in_tags', 'web_traffic'] Then they come after them in terms of importance, with nine features. They are also common to all, but they are less important, and thus the decrease in importance for the rest of the features, which is 15 features, This is followed by another data set of 25 features, as shown in Fig. 6 and Fig. 7.



Weight	Feature
0.1251 ± 0.0034	SSLfinal_State
0.1166 ± 0.0049	URL_of_Anchor
0.0365 ± 0.0026	Prefix_Suffix
0.0319 ± 0.0018	web_traffic
0.0275 ± 0.0025	Links_in_tags
0.0261 ± 0.0025	having_Sub_Domain
0.0122 ± 0.0012	age_of_domain
0.0116 ± 0.0019	Request_URL
0.0102 ± 0.0013	Links_pointing_to_page
0.0097 ± 0.0006	SFH
0.0087 ± 0.0010	Domain_registeration_length
0.0083 ± 0.0005	having_IP_Address
0.0080 ± 0.0013	DNSRecord
0.0067 ± 0.0015	Google_Index
0.0064 ± 0.0009	Page_Rank
0.0033 ± 0.0004	URL_Length
0.0018 ± 0.0006	Redirect
0.0018 ± 0.0006	HTTPS_token
0.0015 ± 0.0001	having_At_Symbol
0.0011 ± 0.0008	Statistical_report
… 10 more …	

Fig. 6. Using Permutation Importance from eli5.sklearn, extracting the top features for the dataset

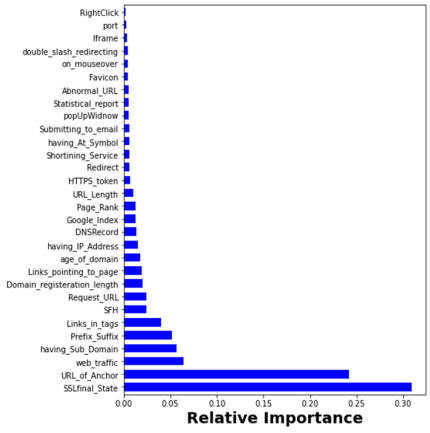


Fig. 7. Using Extra Trees classifier extracting the top features for the dataset

Many features in datasets are unnecessary or redundant. They must be pruned or filtered with the goal of identifying the target objects. The goal of feature selection is to find a subset of features that "improve the learner's ability to classify patterns," thus improving accuracy.

3.6 K-fold Cross Validation

Cross-validation is necessary in evaluating the designed model. Since the classifier used is trained on a specific set of data, this results in a high classification accuracy of the used data set only. Therefore, a method must be used to confirm the validity of the technique used. Cross-validation will not improve the finishing classification accuracy, but it does provide reliability to the classifier employed and can be generalized to other objective datasets. Datasets are randomly divided into separate k-folds of approximately equal size, and each fold is used to test the induced model. The classifier calculates the average by means of an accuracy of k. Thus, in our work, a k-fold technique was used to validate the K Decision Tree and K Random Forest. For K, it uses 10 folds.

3.7 Clustering Methods

On the phishing website datasets, unsupervised clustering was done using two distinct algorithms: K-Means clustering and agglomerative clustering. The Linkage Ward, Euclidean Distance, was the distance metric employed in both techniques. The algorithm employs the so-called linkage criteria to select which distance to utilize between sets of observed data for each sort of cluster. "Ward" reduces the variance of the combined clusters. The shortest distance between two places is known as the Euclidean distance.

3.7.1 K-Means Clustering

Starting with an initial partitioning, a partitioning-based unsupervised clustering algorithm reallocates data points by transferring them from one cluster to another. The cluster centers are initialized as K points in this technique. Every point in the dataset is allocated to the cluster to which it is closest in each iteration. The cluster center is then reset to the cluster set's mean, and the clustering iteration continues until convergence is reached.

3.7.2 Ward Agglomerative Clustering

By iteratively separating or combing the data points, hierarchical clustering creates the groupings. The clusters are generated in agglomerative clustering (a method within the wider family of hierarchical clustering methods) by iteratively merging smaller clusters starting from a single data point until the requisite number of clusters is obtained. Within all clusters, the Ward's distance minimizes the overall inter-cluster sum of squared distances.

3.7.3 Optimal Number of Clusters

Traditional (Hierarchal, K-mean) clustering algorithms detect the number of clusters by hand as input by the user but are impracticable. The user may possibly be a beginner, not know how many classes are suitable for the database, or not be aware of the kind of the dataset. The reason for using these algorithms is their popularity, their frequent use in previous studies, their ease of use, and their speed of performance. Our work suggests using the elbow method to determine the number of classes without relying on the user. There are several branches of the elbow method, and we used the most famous of them.

3.7.3.1 Elbow Method

The elbow method (Kodinariya et al., 2013). is a method for determining the best number of clusters to choose based on heuristics like inter-cluster and intra-cluster similarity. The number of clusters is iteratively raised from 2 to 12, and the elbow, or an optimal number of clusters, is picked at the point where the graph of the cost function has the largest curvature. The elbow approach was used to discover the best number of clusters for K-means clustering using the distortion score as the cost function, and the silhouette score was used to guarantee that the intra-cluster similarity was optimal. The reasons for both scores may be found further down this page. The hierarchical clustering was split into the same number of clusters as the elbow method to compare the results of the two procedures.

3.7.3.1.1 Distortion Score

This is a branch of the elbow method, and this measure provides information about the total difference in mass. It is calculated as the mean sum of the squared distances between the centers. Fig. 8 shows the appropriate clusters for our data set according to this scale, which consists of four classes.

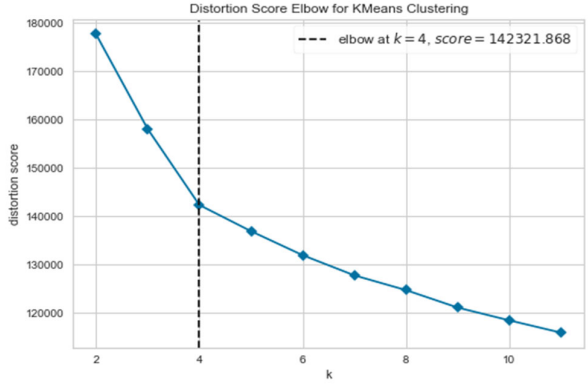


Fig. 8. Distortion Score Elbow for K-Means

3.7.3.1.2 Silhouette Score

This is also one of the branches of the elbow method, and the information this scale provides is information about the overall similarity in mass. It is calculated as the ratio of the average distance within one cluster and the distance next to the nearest other cluster. When the categories are dense, the result is higher and well separated. Fig. 9 shows the silhouette score chart as a result of fitting the K-Means model on the data set with the change of K value from 2 to 12. The production of the evaluation for the elbow method was to represent the elbow line at point 2. It can be incidental that the similarity within the cluster for K = 4 The samples were off center (Silhouette Score 4 Classes = 0.129) and when the intra-cluster similarity was set at K = 2, the illustrations were off center (Silhouette Score 2 Classes = 0.237), and k signifies the optimal number of clusters to prevent data overfitting and get well-rounded groups.

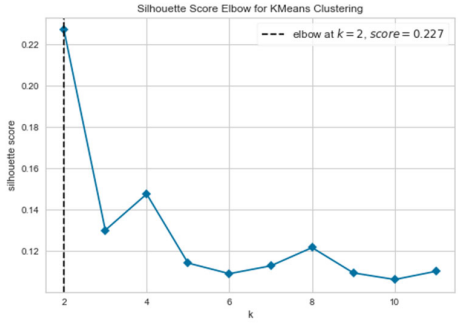


Fig. 9. Silhouette score Elbow for K-Means

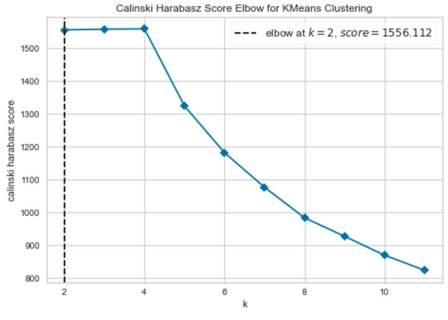


Fig. 10. Calinski Harabasz score Elbow for K-Means

3.7.3.1.3 Calinski Harabasz

This is also one of the branches of the Elbow method, which is known as the variance ratio criterion in the dataset, and it gives an evaluation of the model. This metric provides information about the mass in case its label is not known. The ground truth. It links the clusters with a higher score and, in turn, defines the clusters better. The denser and well separated the clusters, the higher the accuracy of determining K, so the score of this scale is known as the ratio of the average dispersion between clusters and the dispersion within the cluster (dispersion is defined as the sum of the squared distances within the cluster for all the clusters), Figure 10 shows the Calinski-Harabasz score chart.

3.8 Multiclass classification

The classification stage, which is the last stage in our proposed model, starts after we finished the clustering stage, whose work was to generate several groups, between them strong ties, from the original data set, and the optimal number of clusters was two classes and four classes, which was determined through our experiments on the phishing web sites data set based on the Elbow method. We take these outputs (two classes and four classes) and make them the inputs to the two classification algorithms, namely the decision tree and the random forest, and then we calculate their accuracy, comparing them with traditional classification algorithms' accuracy before clustering.

3.9 Performance Evaluation Metrics

To evaluate the performance, we used a set of equations to measure the effectiveness of the classifiers in our model (Rana & Venkata Suryanarayana, 2020). Metrics have been used to evaluate the effects of the experiments, such as accuracy, recall, precision, and the confusion matrix.

$$\text{Accuracy}=(TP+TN)/(TP+TN+FP+FN) \quad (1)$$

$$\text{Recall}=TP/(TP+FN) \quad (2)$$

$$\text{Precision}=TP/(TP+FP) \quad (3)$$

3.9.1 Multiclass Averaging

The macro-averaged method is based on the micro-averaged method, but it works to find the rate locally for each class instead of globally, as in micro-averaged, or with respect to both labels and examples (micro average) (Koyejo et al., 2014). Classification metrics where both the fact and approximation columns are influences for the binary and multiclass case (Pillai et al., 2017).

3.9.1.1 Macro averaging

Macro averaging breaks down multi-class forecasts into numerous sets of binary predictions, calculates the metric for each binary state, and then averages the results (Grandini et al., 2020). The formula representation is equation 10. The average macro simplifies the problem by allowing many comparisons versus all in the multi- class scenario. For each "relevant" column, precision is determined. This procedure is repeated for every subsequent level. After that, the results are averaged (Grandini et al., 2020).

$$P\ \text{macro}=(Pa + Pb + ...+Pn)/(K) \quad (4)$$

where a, b, …, n: Classes, K: number of classes

3.9.1.2 Micro averaging

Micro averaging calculates a single measure rather than the k measurements that are averaged together, using the full data set as an aggregated result (Sagala, 2022).

$$P\ \text{micro}=(TPa + TPb + ...+TPn)/((TPa+TPb+\cdots+TPn)+(FPa+FPb+\cdots+FPn)) \quad (5)$$

Instead of each class being given equal weight, each observation is given equal weight in this situation. This increases the power of the groups with the most observations (Abdulhamit Subasi & Kremic, 2020).

3.10 Confusion Matrix

The confusion matrix is used by constructing a 2×2 matrix to visualize the efficiency of a binary supervised learning problem. The instances in a predicted class are shown in each line in the matrix and each column displays the instances in the actual class (Grandini et al., 2020). It is used to assess a classifier or model's ability to distinguish the dataset's classes. TP and TN denote correctly classified data, while FN and FP denote incorrectly classified data. TP and TN are more accurately classified than FN and FP by the accurate classifier or model. The resulting matrix is composed of four values (Subasi & Kremic, 2020). The Confusion Matrix is shown as follows in Table 2 Confusion Matrix.

Table 2
Confusion Matrix

	Positive (Actual)	Negative (Actual)
Positive (Predict)	TP	FP
Negative (Predict)	FN	TN

4. Experimental results and discussion

The experiments in this research were done using python 3. The experiments were carried out using a Windows 10.1 64-bit laptop with a Core-i7 processor, 2.30 GHz, and 16 GB of RAM.

4.1 The Experimental Results

This section includes the results of the experiments of each classifier that were conducted on each dataset in the two phases. They were conducted on the sub-set of the dataset and on the two phases. Depending on each phase and the feature selection appropriate for it, we will show results for each case.

4.1.1 Traditional classification results

The first phase is traditional classification. After the preprocessing operation is done on our dataset containing (342705 samples), it becomes ready for the classification phase, where each of the data points passes to two models, the Decision Tree (DT) and the Random Forest (RF), to train and test the dataset on them. Accuracy was calculated for the two models using evaluation measures, which are accuracy, recall, and precision. This is depicted in Table 3 and in Fig. 11.

Table 3
Traditional classification results

		Accuracy	Precision	Recall
All Dataset	**Decision Tree**	0.963	0.960	0.972
	Random Forest	0.969	0.960	0.985

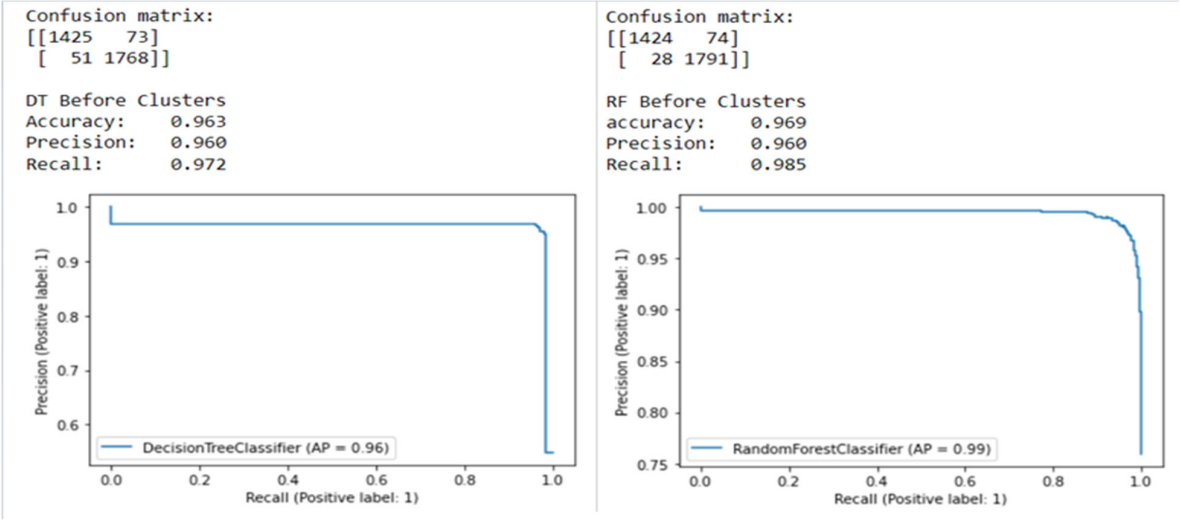


Fig. 11. Traditional classification results

4.1.2 Traditional classification with feature selection

After performing the process of feature selection from our dataset five features dropped, namely ('RightClick', 'Iframe', 'port', 'on_mouseover', 'double_slash_redirecting'), the remaining dataset becomes the one that represents our dataset that contains (287430 samples), where each of the data points (after reducing from 30 features to 25 features) passes into the two models, the Decision Tree (DT) and the Random Forest (RF), to train and test the data set on them. Accuracy for the two models was calculated. This is illustrated in the table below 4 and in Fig. 12.

Table 4
The top twenty-five features in a traditional classification resulted in results

		Accuracy	Precision	Recall
Top (25) Features	**Decision Tree**	0.965	0.963	0.973
	Random Forest	0.971	0.961	0.987

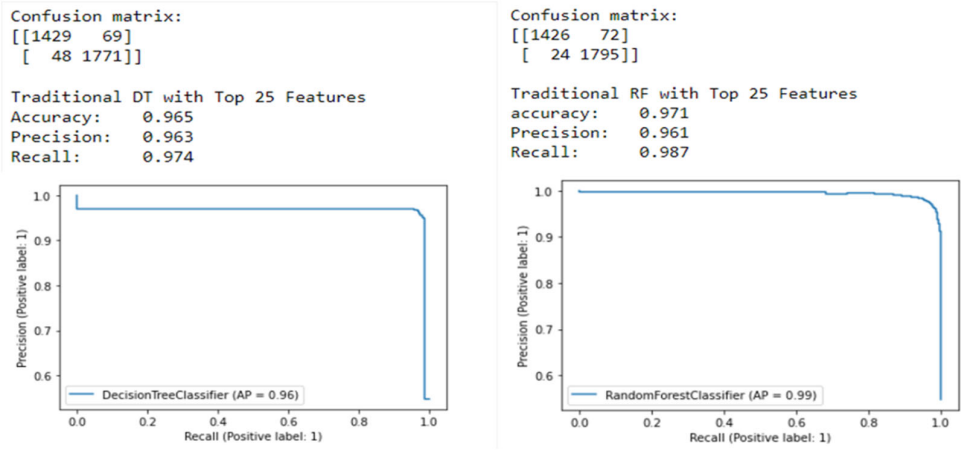


Fig. 12. The traditional classification with the top twenty-five features yields results.

4.1.3 Traditional classification with K-fold

Cross-validation is better used for the data. It is a potent tool. Sometimes we may overlook and use the same data at different stages of the workflow (such as training and testing). This may produce positive results, but in most cases, they are illusory or cause strange side effects. Using cross-validation, we can test all the data samples. For each sample, we make the prediction using our models (Decision Tree and Random Forest), which have not been trained on it (i.e., the sample). We can use all the data samples for both training and testing while maintaining the condition of verifying the models using samples that have never been seen before. By using cross-validation, we can get more measurements to better understand and make clearer decisions, both at the level of our algorithms and our data. In our current experiment, we used the K-fold-10 once with all the data and the twenty-five strongest distinguishing features. To report classification results, this shows the two tables 5 and 6.

Table 5
Use K-fold 10 with all data

		K-fold	Accuracy	Precision	Recall
All Dataset	**Decision Tree**	**Maximum**	0.975	0.974	0.980
		Minimum	0.958	0.943	0.981
	Random Forest	**Maximum**	0.983	0.982	0.987
		Minimum	0.966	0.953	0.985

When using K-fold 10 above, we find that the resulting values are very meaningful for all datasets through the degree of measurement, which is the accuracy. We conclude that our work was reliable in terms of recording reading accuracy for the previous two models.

Table 6
Use K-fold 10 with the top twenty-five features

		K-fold	Accuracy	Precision	Recall
Top (25) Features	**Decision Tree**	**Maximum**	0.975	0.974	0.980
		Minimum	0.953	0.937	0.978
	Random Forest	**Maximum**	0.983	0.984	0.985
		Minimum	0.967	0.951	0.988

When the amount of data decreases, we notice a speed in performance. which gives reliability to the algorithms we use.

4.2 Clustering Results

In this approach, we used two different clustering techniques, such as K-mean clustering and hierarchical clustering. Following that, new data sets consisting of two and four classes were generated, and this clustering is considered optimal, Fig. 13 and Fig. 14 show the new data format after the clustering process.

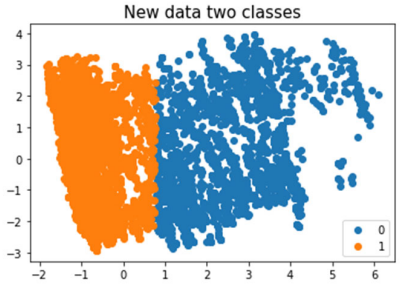


Fig. 13. The new data format was divided into two classes

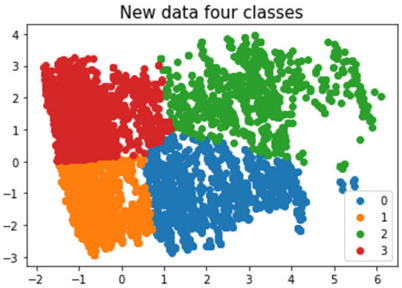


Fig. 14. The new data format was divided into four classes

4.2.1 Multiclass classification Results

After completing the clustering process and producing new data sets and making sure that no sample is lost from the original data set, The new dataset will be the input for the multi-label classification process. Another concept, output labels are a purpose of the inputs. As part of our proposed work to improve the detection of phishing websites, we used macro averaging to calculate the measure for each binary condition and then average the results. The multiple classification models are Decision Tree (DT) and Random Forest (RF). The models mentioned were trained and tested on the new data set once with K-mean clustering and again with hierarchical clustering. We expected that this proposed approach would be more accurate than the result of traditional phishing classifiers (the product of a two-label classification), and indeed, after conducting the experiment, the accuracy was very good, which is what we needed before this study. The accuracy of the two models was calculated using rating scales for accuracy, recall, and precision. We notice an improvement in accuracy when the classifier works with two classes. This is shown in Table 7.

Table 7
Classification results after the clustering process for all data

		K-means			Hierarchal		
		Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Four classes	**Decision Tree**	0.989	0.990	0.987	0.985	0.989	0.989
	Random Forest	0.993	0.994	0.990	0.991	0.993	0.994
Two classes	**Decision Tree**	0.9988	0.9979	0.9979	0.9994	0.9991	0.9991
	Random Forest	0.9991	0.9995	0.9973	1.0000	1.0000	1.0000

4.2.2 Comparison of the two phases before feature selection

To compare the two phases, the difference in the improvement in accuracy is evident in the second phase. The results are also shown in Table 8.

Table 8
Comparison of the accuracy in the first and second phases before feature selection

		Decomposition approach			Features	Traditional algorithms		
		Accuracy	**Precision**	**Recall**		**Accuracy**	**Precision**	**Recall**
All features Four classes	**Decision Tree**	0.988	0.991	0.991	**All features**	0.964	0.963	0.971
	Random Forest	0.991	0.994	0.994		0.970	0.961	0.985
All features Two classes	**Decision Tree**	0.9994	0.9991	0.9991	**All features**	0.964	0.963	0.971
	Random Forest	1.0000	1.0000	1.0000		0.970	0.961	0.985

The results of the study show the superiority of the two clusters over the four clusters, and this is generally due to the nature of the dataset we worked on, and also the superiority of the hierarchical clustering over the k-mean clustering in the two clusters due to the additional functions of the hierarchical clustering that helped the classifiers predict better. The comparison shows the superiority of the second phase over the first phase through the drawing shown in Fig. 15.

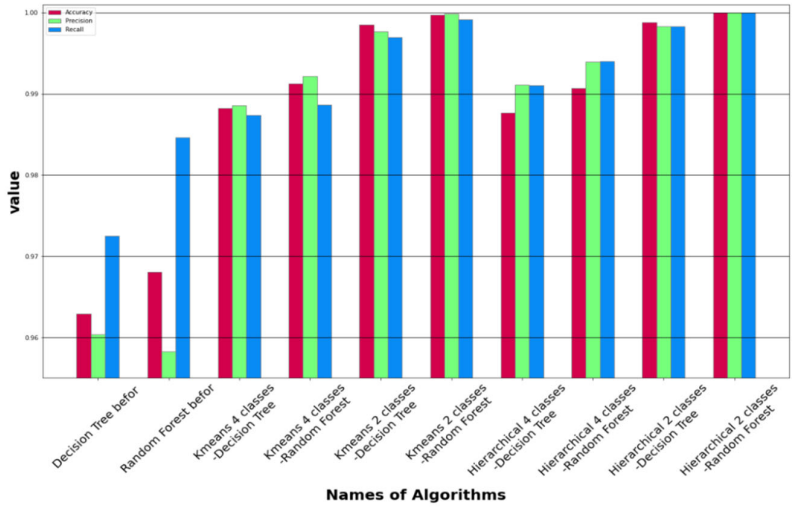


Fig. 15. The accuracy comparison in the first and second phases before the feature selection

4.3 Clustering Results after feature selection

After conducting the process of feature selection from the standard dataset, these features become the dataset containing (77385 samples). We will apply the clustering algorithms (K-Means and hierarchal) to the reduced dataset (30 to 6 features) and the outputs (2 and 4 classes), The two figures 16 and 17 show the reduced dataset format after the clustering process.

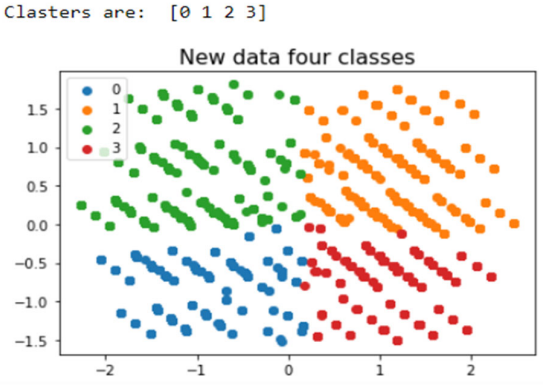


Fig. 16. The reduced dataset format was divided into four classes

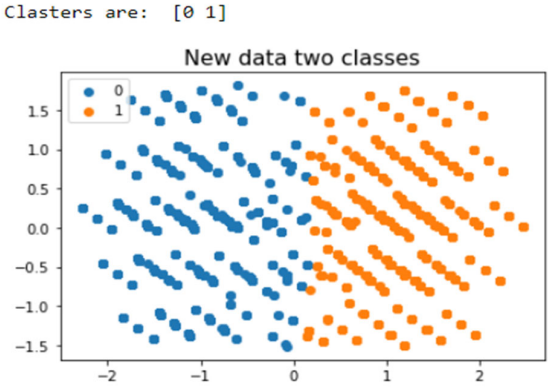


Fig. 17. The reduced dataset format was divided into two classes

4.3.1 Multiclass classification Results with feature selection

The data above represents our dataset after feature selection (six features), which in turn represents the inputs to both multi-classification algorithms, which are the decision tree and the random forest. After the application of the two algorithms to the data classes, the accuracy of the two models was achieved using measures of accuracy for accuracy, precision, and recall. The results are displayed in Table 9.

Table 9
Classification results after the clustering process with the top six features

		K-means			Hierarchal		
		Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Top (6) Features Four classes	**Decision Tree**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Top (6) Features Two Classes	**Decision Tree**	0.9997	0.9997	0.9996	1.0000	1.0000	1.0000

4.3.2 Comparison of the two phases after feature selection

To compare the two phases, after features selection, the improvement in accuracy is evident in the second phase. The results are also shown in Table 10.

Table 10
Comparison of the accuracy in the first and second phases after feature selection

		Decomposition approach With features selection			The best features selection	Traditional algorithms With features selection		
		Accuracy	Precision	Recall		Accuracy	Precision	Recall
Top (6) Features Four classes	Decision Tree	1.0000	1.0000	1.0000	Top (25) Features	0.964	0.963	0.971
	Random Forest	1.0000	1.0000	1.0000		0.970	0.961	0.985
Top (6) Features Two classes	Decision Tree	1.0000	1.0000	1.0000	Top (25) Features	0.964	0.963	0.971
	Random Forest	0.9997	0.9997	0.9996		0.970	0.961	0.985

The results of the study after selecting the top six features show the superiority of the four clusters over the two clusters. This is generally due to the nature of the selection of the feature set we worked on, and the superiority of both hierarchical grouping and average K grouping in the four clusters over the two clusters was due to the additional functions of each of them that helped the classifiers to better predict and each of them the nature of their work. While the hierarchical clustering in the two clusters exceeded the average K clustering due to the way the clusters were calculated with higher accuracy after the selection of features from the four clusters, this gave additional functions that helped the classifiers predict better, and the decision tree classifier was superior after the selection of features over the forest random classifier. The comparison displays, after features selection, the pre-eminence of the second phase over the first phase through the sketch shown in Fig. 18.

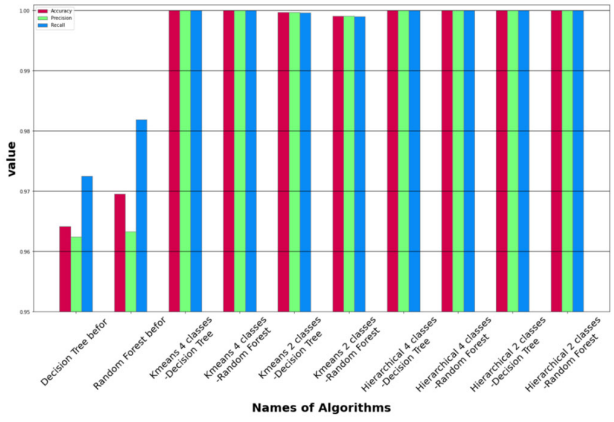


Fig. 18. The accuracy comparison in the first and second phases after the feature selection

4.4 Comparison of our results in previous studies

Comparing the results of our study with the results of previous studies that used the same standard data set that we used in our study and illustrating the tools used in the detection of accuracy are shown in the Table 11.

Table 11
Comparing the results of our study with the results of previous studies using the same dataset

		Phishing Website Detection			
ID	Authors	Machine Learning Techniques	Accuracy	Recall	Precision
1	Mustafa Kaytan et al. 2017 (KAYTAN et al., n.d.)	Extreme learning machine (ELM), NN	95.05%	-	-
2	Sheikha Verma et al. 2020 (Verma & Gautam, n.d.)	Random Forest, Decision Tree, J48, Support Vector Machines, Naive Bayesian, Neural Network, Logistic Regression, Lazy K Star, and the C4.5 algorithm.	97.25%	-	-
3	Yousif Jabbar our study 2022	Random Forest, Decision Tree, K-mean, and Hierarchical	100%	100%	100%

5. Conclusion and future work

This study can be summed up as providing a new way to increase the accuracy of detecting phishing sites through machine learning algorithms. Our work was divided into two phases:

- In the first phase, we used traditional algorithms before and after the application of feature selection algorithms, and the highest accuracy we recorded in this phase was 97% with the random forest classification algorithm. Our main goal in this study is the second phase in which evidence from the experimental results shows a significant improvement in accuracy. We have applied the clustering technique in both algorithms (k-means clustering and hierarchical clustering) before the classification. The algorithms used in this research are the decision tree and random forest, which are very suitable.
- The elbow method was very useful as it helped to detect the required number of sets by using the k-means algorithm. This was appropriate for the dataset. Also, the accuracy of classifiers can be improved by applying the selection of a set of sub-features before the clustering technique, and the classification time can also be reduced.
- A The results were achieved in the second phase, and the highest accuracy was recorded before feature selection at 100% with a random forest classification algorithm with hierarchical clustering of two classes. The results in the second phase achieved the highest accuracy after feature selection with 100% with the decision tree classification algorithm in all its cases (two and four classes), except 99.97% with K-means clustering of two classes, and scored 100% with the random forest classification algorithm with four classes and 99.97% with hierarchical clustering of two classes.
- In conclusion, the second phase made a vast difference from the first phase.

For future work in this area, it is hoped that some future work can determine the feasibility of the proposals below.

- The use of (Web Scraping) through a program that simulates human Internet browsing to collect (30 features) from the URL, in addition to the output column that determines whether the website is phishing or legitimate, is considered a preliminary data set.
- Before applying the classification algorithms, the implementation of feature selection algorithms and then clustering algorithms must be completed. The searcher recommends using the algorithms used in this research because they are useful in the discovery of the site's phishing or legitimacy.
- Implement the proposed system for other types of primary data sets.
- Educate and train users besides detecting phishing attacks. Previous studies have shown that 61% of website users were not familiar with phishing detection tools.

Acknowledgment

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

References

Almadhoor, L. (2021). Social media and cybercrimes. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(10), 2972-2981.

Babagoli, M., Aghababa, M. P., & Solouk, V. (2019). Heuristic nonlinear regression strategy for detecting phishing websites. *Soft Computing, 23*(12), 4315-4327.

Bhardwaj, A., Al-Turjman, F., Sapra, V., Kumar, M., & Stephan, T. (2021). Privacy-aware detection framework to mitigate new-age phishing attacks. *Computers & Electrical Engineering, 96*, 107546.

Bitaab, M., Cho, H., Oest, A., Zhang, P., Sun, Z., Pourmohamad, R., ... & Ahn, G. J. (2020, November). Scam pandemic: How attackers exploit public fear through phishing. In 2020 APWG *Symposium on Electronic Crime Research* (eCrime) 1-10.

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.

Deshpande, A., Pedamkar, O., Chaudhary, N., & Borde, S. (2021). Detection of Phishing Websites using Machine Learning, *International Journal Of Engineering Research & Technology, 10*.

Gautam, S., Rani, K., & Joshi, B. (2018). Detecting phishing websites using rule-based classification algorithm: a comparison. *In Information and Communication Technology for Sustainable Development*, 21-33.

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756.

Gutierrez, C. N., Kim, T., Della Corte, R., Avery, J., Goldwasser, D., Cinque, M., & Bagchi, S. (2018). Learning from the ones that got away: Detecting new forms of phishing attacks. IEEE *Transactions on Dependable and Secure Computing, 15*(6), 988-1001.

Hannousse, A., & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence, 104*, 104347.

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.

Kamal, G., & Manna, M. (2018). Detection of phishing websites using naïve Bayes algorithms. *International Journal of Recent Research and Review, 11*(4), 34-38.

Kaytan, M., & Hanbay, D. (2017). Effective classification of phishing web pages based on new rules by using extreme learning machines. *Computer Science, 2*(1), 15-36.

Koyejo, O. O., Natarajan, N., Ravikumar, P. K., & Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. *Advances in neural information processing systems, 27*.

Liu, C., Wang, L., Lang, B., & Zhou, Y. (2018, January). Finding effective classifier for malicious URL detection. *In Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*, 240-244.

McAnulty, B. L. (2021). Phishing Attacks: A Plan to Educate Employees and Mitigate Risks (Doctoral dissertation, Utica College).

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.

Niakanlahiji, A., Chu, B. T., & Al-Shaer, E. (2018, November). Phishmon: A machine learning framework for detecting phishing webpages. *In 2018 IEEE International Conference on Intelligence and Security Informatics*, 220-225.

Niu, W., Zhang, X., Yang, G., Ma, Z., & Zhuo, Z. (2017, December). Phishing emails detection using CS-SVM. *In 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC),* 1054-1059.

Pillai, I., Fumera, G., & Roli, F. (2017). Designing multi-label classifiers that maximize F measures: State of the art. *Pattern Recognition, 61*, 394-404.

Rana, V. K., & Suryanarayana, T. M. V. (2020). Performance evaluation of MLE, RF and SVM classification algorithms for watershed scale land use/land cover mapping using sentinel 2 bands. *Remote Sensing Applications: Society and Environment, 19*, 100351.

Rao, R. S., Vaishnavi, T., & Pais, A. R. (2019). PhishDump: A multi-model ensemble based technique for the detection of phishing sites in mobile devices. *Pervasive and Mobile Computing, 60*, 101084.

Sagala, N. T. (2022, January). Comparative Analysis of Grid-based Decision Tree and Support Vector Machine for Crime Category Prediction. *In 2021 International Seminar on Machine Learning, Optimization, and Data Science* (ISMODE), 184-188.

Shirazi, H., Haefner, K., & Ray, I. (2017, August). Fresh-phish: A framework for auto-detection of phishing websites. *In 2017 IEEE international conference on information reuse and integration* (IRI), 137-143.

Singh, P., Maravi, Y. P., & Sharma, S. (2015, February). Phishing websites detection through supervised learning networks. *In 2015 international conference on computing and communications technologies* (ICCCT), 61-65.

Somesha, M., Pais, A. R., Rao, R. S., & Rathour, V. S. (2020). Efficient deep learning techniques for the detection of phishing websites. *Sādhanā, 45*(1), 1-18.

Subasi, A., & Kremic, E. (2020). Comparison of adaboost with multiboosting for phishing website detection. *Procedia Computer Science, 168*, 272-278.

Tang, L., & Mahmoud, Q. H. (2021). A survey of machine learning-based solutions for phishing website detection. *Machine Learning and Knowledge Extraction, 3*(3), 672-694.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Tyagi, I., Shad, J., Sharma, S., Gaur, S., & Kaur, G. (2018, February). A novel machine learning approach to detect phishing websites. *In 2018 5th International conference on signal processing and integrated networks* (SPIN), 425-430.

Ubing, A. A., Jasmi, S. K. B., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Phishing website detection: An improved accuracy through feature selection and ensemble learning. *International Journal of Advanced Computer Science and Applications, 10*(1).

Verma, S., & Gautam, A. K. (2020). A Survey on Phishing Detection and The Importance of Feature Selection In Data Mining Classification Algorithms.

Yadollahi, M. M., Shoeleh, F., Serkani, E., Madani, A., & Gharaee, H. (2019, April). An adaptive machine learning based approach for phishing detection using hybrid features. *In 2019 5th International Conference on Web Research* (ICWR), 281-286.

Yang, L., Zhang, J., Wang, X., Li, Z., Li, Z., & He, Y. (2021). An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. *Expert Systems with Applications, 165*, 113863.

Yang, P., Zhao, G., & Zeng, P. (2019). Phishing website detection based on multidimensional features driven by deep learning. *IEEE access, 7*, 15196-15209.

Zhu, E., Ju, Y., Chen, Z., Liu, F., & Fang, X. (2020). DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features. *Applied Soft Computing, 95*, 106505.