An Empirical Feature Selection Approach for Phishing Websites Prediction with Machine Learning

Pankaj Bhowmik[1](✉), Md. Sohrawordi[1], U. A. Md. Ehsan Ali[1], and Pulak Chandra Bhowmik[2]

[1] Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur 5200, Bangladesh
pankaj.cshstu@gmail.com, ehsan_cse@hstu.ac.bd
[2] Department of Computer Science and Engineering, Stamford University Bangladesh, Dhaka 1217, Bangladesh

Abstract. The proportion of phishing attacks has soared worldwide amid the Covid-19 crisis since people started using the internet more actively. Browsing phishing websites can cause immense damage to user privacy. In this article, investigating the attributes of URLs to detect the possible legitimate and phishing websites, we presented a feature selection framework that improves the efficacy of machine learning models. In feature selection, considering the filter and wrapper method, we introduced an empirical hybrid framework that comprises two phases. To derive the accumulative feature subset, in the early stage, we performed a function perturbation ensemble using four filter techniques. Finally, to select the best features, we employed the wrapper method, in which the feature subset is passed into a statistical model to perform a p-value test (conforming 95% confidence). We used two phishing datasets, and applying this proposed hybrid ensemble framework, we derived only 45.95% of the initial features from each dataset. Thereafter, the optimized (hyperparameters) models such as Artificial Neural Network, XGBoost Classifier, Random Forest Classifier are applied to conduct 10-folds cross-validation on Data-I, the XGBoost Classifier outran with the accuracy of 96.08%. Besides, the XGBoost model performed prediction on Data-II, achieved a notable accuracy of 97.29%.

Keywords: Phishing detection · Empirical feature selection · Machine learning

1 Introduction

During the COVID-19 lockdown, the Internet has become truly essential to perform everyday tasks. Several facilities and services are getting digitalized daily. Besides, people are embracing this trend because of the conveniences they get out of it. Particularly, online marketing and banking transactions achieved utmost popularity. But, hackers always try to break the security protocols of the Internet to steal confidential information of users. They attempt to seduce people and mug private data through forged

A. K. M. M. Islam et al. (Eds.): ICBBDB 2021, CCIS 1550, pp. 173–188, 2022.
https://doi.org/10.1007/978-3-031-17181-9_14

websites [1]. One of the common ways to do this is the phishing attack, a well-known cybercrime. Cyber-criminals usually replicate the contents of original websites to make the users believe that they are surfing authentic information. Besides, attackers often try to install malware by forwarding spam emails or links on social media to take control over a user system. Spam links redirect the users to phishing websites while they click on them unknowingly. As a result, people disclose their private data e.g., passwords, credit card info to hackers. The phishers usually target specific organizations such as banks, govt. database centers, defense and law enforcement agencies, and people such as celebrities, govt. officials. Phishing attack has become considerably sophisticated lately and one of the routine cyber crimes these days [2].

In these circumstances, phishing attacks are becoming a burning concern for cyber-security departments all over the world. To address this cynical issue, researchers and cyber-security experts are constantly working hard—besides, they proposed different possible solutions [3]. Phishing attacks detection based on 'Blacklist' is one of the popular preventive approaches, web browsers use this list to warn users about a potential phishing website. The 'Blacklist' contains the universal resource locator (URL) of all known phishing websites. If a surfed URL is listed on the Blacklist, then it's a phishing website and legitimate otherwise, the browsers provide warnings accordingly. But the drawback is, as tons of phishing websites are developing in a daily manner, and if the 'Blacklist' is not updated, browsers will not be able to detect the newly generated phishing URLs. Besides, hackers deploy dynamic methods to crack the 'Blacklist' approach which can be a major threat, and they also develop mirror URLs to exploit the security loopholes. However, the 'Whitelist' approach works oppositely. It contains a list of legitimate websites, and the browsers allow only the listed sites to pass through the system gateway [4]. Besides, in the heuristic-based technique, which is an extended variant of the listing-based method, URLs and other features are extracted from websites and compared with ground-list to decide websites legitimacy [5, 6]. On the other hand, machine learning (ML) based phishing attacks detection is becoming more dominating lately. In ML approaches, lists of features are extracted from URLs to predict phishing sites, as a result, these methods can effectively combat the dynamic changes of phishing attacks. The traditional ML algorithms and neural networks are doing great jobs to detect phishing websites. These robust ML models perform significantly well and are much reliable [1, 7].

In URL-based phishing websites detection, many features are extracted but not all of them are equally important. Consequently, researchers have introduced several feature selection methods to rank and select the best features from the feature space. To cite an example, Chiew et al. [8] established a novel feature selection framework named hybrid ensemble feature selection which has two phases: data perturbation and function perturbation. The data perturbation cycle derived the primary and secondary feature subsets gradually, and the final features are obtained from the function perturbation ensemble cycle. The combination of both these cycles results in a hybrid ensemble feature selection. Waleed Ali [9], on the other hand, experimented with two feature selection approaches for phishing website detection. The performance of the proposed ML models revealed that wrapper-based feature selection outperformed the filter method. Barbara Pes et al. [10] established an ensemble-based substantial approach for feature selection. They experimented with the data perturbation strategy using a set of feature selection methods. In general, they found that the ensemble feature subsets provide good outcomes, particularly in terms of stability.

In this article, we propose a hybrid feature selection framework to derive the best features from phishing datasets and thus to detect phishing websites effectively and with ease. The feature selection framework has two phases—at the beginning phase, a set of filter methods are applied to select the primary feature subsets from the dataset and then obtained the secondary (accumulative) feature subset with the function perturbation ensemble. In the final phase, the wrapper method is applied where the secondary feature subset fed is in a statistical model (Bi-directional elimination) to select the best features with a 'p-value test' ensuring 95% confidence. We used two latest phishing datasets of Grega Vrbančič for the experiment, available on Mendeley. In this proposed study, we have attempted to find out answers to the following research questions:

RQ-1: How does the Hybrid Feature Selection approach boost the efficacy of ML models to perform better?

RQ-2: Can the proposed framework outrun the previous research findings?

2 Related Studies

Over the years, scholars and cyber-security experts have developed several methods to combat phishing attacks. Until now, the performance of the ML-based approaches reflects their superiority over the conventional methods. In this section, we shed some light on the contemporary ML-based solutions introduced by the researchers to fight against phishing attack.

Based on the datasets we used in this study, Vrbančič et al. presented a method to address the parameter setting issue of deep neural networks (DNN). They applied swarm intelligence meta-heuristic algorithms (bat algorithm, firefly algorithm) to optimize DNNs parameters. They used four phishing datasets for their experiment and achieved promising outcomes in classifying phishing websites—the proposed firefly method outplayed. Considering Vrbančič's small and full dataset, the firefly method showed an accuracy rate of 90.17% and 94.39% respectively, in cross-validation [11].

Detecting phishing websites based on URLs, ensemble learning models showed better performance compared to the individual traditional algorithms. Mohammed Al-Sarem et al. [2] proposed an optimized ensemble method considering the stacking approach. They experimented with six ML classifier models, and the parameters of those models were optimized using a genetic algorithm (GA). Following that, the models were ranked to select the best three classifiers to perform stacking ensemble. However, the ensemble model with SVM as meta-learner showed an accuracy of 97.39% on Vrbančič full_data. In another study, Yazan Ahmad Alsariera et al. [7] developed meta-learner models in which Extra-tree classifier is considered as the base classifier. The LogitBoost-Extra Tree model achieved the highest accuracy of 97.58% with a false-positive rate of 0.018 in cross-validation. To deal with the dynamism of phishing attacks, Adeyemo et al. [12] proposed an ensemble-based approach that combines the tree induction and logistic regression techniques. They integrated the bagging and boosting methods with the base algorithm ‘Logistic Model Tree’ to build more effective models. However, they used two phishing datasets and achieved a minimum accuracy of 97.18% with the proposed models.

Neural network models tend to be highly efficient for detecting phishing websites. The existing studies showed that these models can predict phishing and legitimate websites mostly with above 90% accuracy, and provide a lower false-positive rate. A deep learning approach is conducted by Somesha et al. [1] to predict the legitimacy of websites based on URLs. They applied the Information Gain feature selection method to rank the features and selected the 10 best features for their experiment. Among the three deep learning models (i.e., LSTM, DNN, CNN), the LSTM outperformed with securing an accuracy of 99.57%. In another experiment [13], to find out the best performing algorithm Vaitkevicius and Marcinkevicius used three phishing datasets and eight widely used machine learning algorithms. The findings of the study showed that multilayer perceptron (ANN) and ensemble-based algorithms performed better. Apart from Neural Networks, the tree-based ensemble algorithms performed significantly well in phishing websites prediction. In particular, the Random Forest, Extra Trees, Gradient Tree Boosting showed promising outcomes [7].

There are tons of open-source phishing datasets available for research purposes, and the datasets contain lots of features. However, all the features in a dataset are not reasonably significant, and those features can affect an ML model’s performance inversely. Consequently, the feature selection methods can be applied to derive the best features from datasets [8, 14, 15]. Waleed Ali experimented with wrapper and filter-based feature selection approaches on a phishing dataset to build effective ML models. He established 7 ML models and made a comparative analysis of their performance. His study concludes that the ML models considering wrapper-based feature selection surpassed the same models with filter method, and without feature selection [9]. In a study, Chiew et al. [8] established a hybrid ensemble feature selection, in which a set of filter methods are applied to derive feature subsets. They determined the cut-off ranks for selecting features with considering the gradient changes. Besides, the proposed feature selection framework selected only 20.8% of features from the data but achieved promising outcomes from the ML models.

Since the phishing attacks are dynamic in nature, after exploring the previous researches—in general, we can reach a consensus that ML-based methods are much reliable in addressing these cynical attacks. In this article, we introduced a hybrid feature selection framework to boost the performance of proposed models. It’s a two-phase hybrid feature selection approach in which filter and wrapper methods are applied accordingly. After selecting the best attributes with this hybrid ensemble approach, the study performed cross-validation and prediction on the datasets. The study experimented with three optimized machine learning models i.e., RFC, XGBC, ANN.

3 Proposed Research Methodology

In this article, we propose an empirical feature selection approach which is more like a hybrid ensemble of the best features from the dataset. In this section, we discussed the overall proposed methodology applied in this study for phishing websites detection. The structure of this methodology has three distinct layers, namely Feature Engineering, Cross-Validation, and Perform Prediction (testing). In this study, we use two phishing website datasets of Vrbančič that are publicly available in Mendeley. We tagged the Vrbančič small_dataset as 'Data-I' and Vrbančič full_dataset as 'Data-II'. The framework of the proposed methodology is illustrated in Fig. 1.

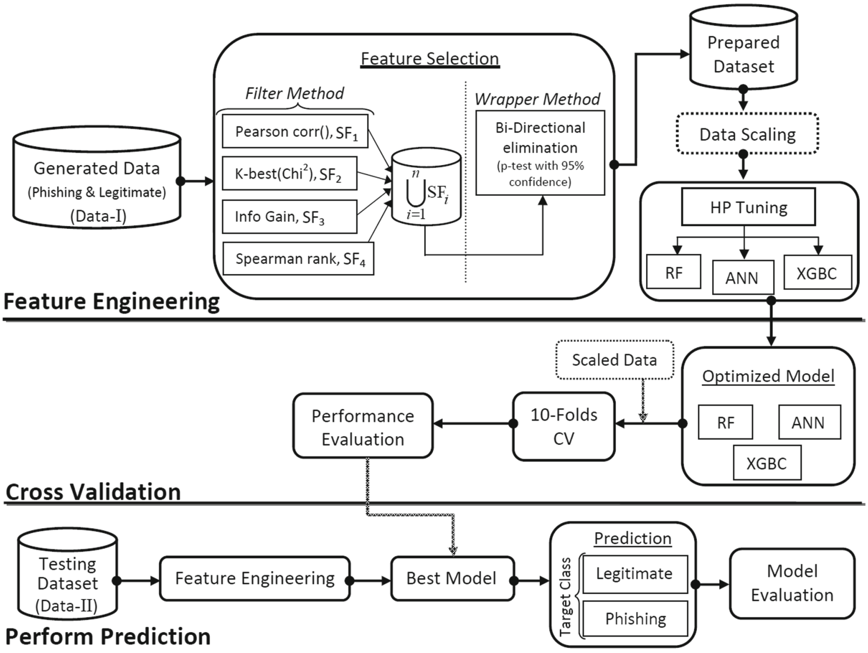

Fig. 1. Architecture of the proposed methodology

Vrbančič collected 58,645 and 88,647 website URLs for Data-I and Data-II respectively and designed a feature extraction algorithm. The algorithm extracts 111 unique features from each URL and saves them in a CSV file. The URLs were collected from Phishtank and Alexa ranking websites, and the instances were labeled (legitimate or phishing) according to the sources [16].

In the Feature Engineering layer, we propose a novel feature selection approach that showed the pathway to derive the best features from the datasets. The feature selection approach, in this study, used a hybrid framework. In this process to select the most important features from Data-I, the raw features are refined through two methods namely the filter method and the wrapper method. In the filter method, the study applied four different techniques such as Pearson correlation (PC), Chi-square (Chi^2), Information Gain (IG), and Spearman correlation rank (SR), and combined the important features with set union operation ($\cup$). Then the feature subset of the filter method is fed to a wrapper method known as Bi-directional elimination (BDE) to select the best features. In the wrapper method, the features are selected with the statistical model which runs a p-value test with 95% confidence intervals, i.e., the probability to accept the null hypothesis is only 5%. Afterward, we get the prepared dataset containing the best features, which is then scaled and fed to the ML models such as Random Forest Classifier, XGBoost Classifier, and Artificial Neural Network for hyperparameter (HP) tuning. Grid Search is applied for tuning the HP of the models, and then the optimized models are passed to the next layer 'Cross Validation'.

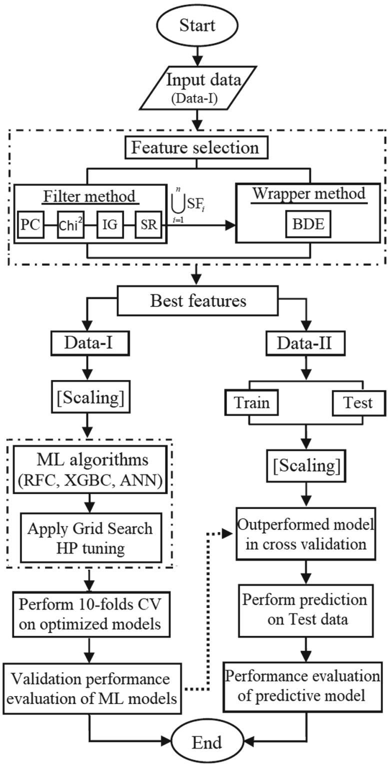

Fig. 2. Architecture of the proposed methodology

In the 'Cross Validation' phase, we performed 10-folds CV on the three optimized models and evaluated their performance. Besides, we made a comparative assessment to find out the outperformed model which is used to perform prediction on Data-II in the succeeding layer. Finally, in the Prediction phase, we split the Data-II into training and testing data then performed prediction on the testing data with the best model. We estimated the model's evaluation metrics to assess its performance in detecting phishing and legitimate websites. The flow of research methodology is shown in Fig. 2.

The experiment was conducted on Jupyter notebook environment using Python programming language. The proposed feature selection algorithm (in Algorithm 1) and ML models are implemented with different python machine learning libraries.

4 Implementation, Results Analysis and Discussion

We carried out the experiment according to the proposed methodology. In this section, we discussed and highlighted the outcomes i.e., statistical modeling and numerical simulation of this experiment.

4.1 Implementation

Dataset Description. The phishing websites dataset used in this experiment is gathered from Mendeley [17]. The dataset was found by Grega Vrbančič, and it has two variants namely, dataset_small (Data-I) and dataset_full (Data-II). He used a feature extraction algorithm to get a list of features from the input (URLs). In general, the features of the datasets can be grouped into 6 classes, such as URL properties, domain properties, URL directory properties, URL file properties, URL parameter properties, and URL resolving data and external metrics. To estimate the value of features, the website URL strings are divided into four sub-strings (domain, directory, file, parameter), besides other external services are considered. Although the number of observations in Data-I and Data-II is 58,645 (phishing-30,647, legitimate-27,998) and 88,647 (phishing-30,647, legitimate-58,000) respectively, both the datasets have an identical and equal number of features (112). The target variable defined as 'phishing', concludes whether an observation of website URL falls in legitimate or phishing class [16] (Fig. 3).

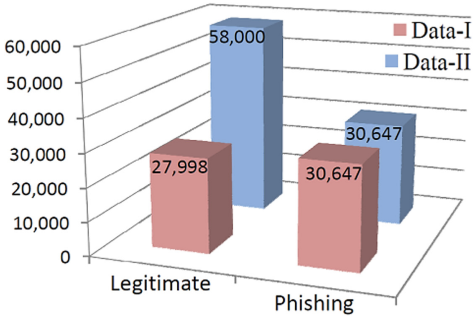

Fig. 3. Graphical representation of Vrbančič's dataset

Feature Engineering. Data-I and Data-II contain 111 features (excluding target variable) each which are one of the highest numbers among the available phishing datasets. However, all these features are not equally useful, and as the number of features increases, the dimension of the data-set also increases. Consequently, we proposed an empirical feature selection approach to select the best features. It's a hybrid approach that applies two methods namely, the filter method and the wrapper method. Filtering the most important feature subsets from the dataset is the first step of this hybrid approach, and in the final step, a wrapper method is applied to select the best features from the feature subset. However, the proposed empirical hybrid feature selection algorithm is noted in Algorithm 1.

Algorithm 1 Empirical Hybrid Feature Selection

```
Input: Dataset (O,F)                                      // O= observations, F= features
Output: Best Features
  Initialization:
  N← FilterMethods(Pearson,$Chi^2$,Info Gain,Spearman)
  for each N do                                                          // Filter method
      $FCV_i$ = CalculateFilterScore(Dataset)
  end for
  $T_N$← InitializeCutOff($T_k$)                     // $T_k$= cut-off range of filter methods (N)
  for each F do
      if $FCV_i \in T_N$ then                                       // Primary feature subset
          $SF_j = \bigcup_{i-1}^{F} FCV_i$                           // Set union operation
      end if
  end for                                                   // BFF = Best filtered feature
      $BFF = \bigcup_{i-1}^{N} SF_j$                                // Function perturbation
  best_feature = []                                                          // Empty list
  StepwiseElimination(BFF,$SL_{in}$=0.05,$SL_{out}$=0.05)                // Wrapper method
  for each feature in BFF do
      p_value = CalculatePvalue(BFF)
      if $p_{min} < SL_{in}$ then                                     // Forward Selection
        add to best_feature
        for each best_feature do
            if $p_{max} >= SL_{out}$ then                            // Backward Selection
              remove from best_feature
            else break
        end for
      else break
  end for
  return best_feature
```

Feature Selection. The filter method, in this study, used four feature selection techniques such as Pearson correlation, Chi-square, Information Gain, and Spearman correlation rank. The techniques estimate the importance score of individual features considering their frequency or correlation with other features. Each technique provides a feature subset (SF) that has the most important features in it. At the end of this method, all the feature subsets are combined with a set union operation—this process is known as 'function perturbation' [8]. Besides, from a technique to select the most correlated features, a cut-off range (Tk) is set. Each technique has an identical cut-off, determined by analyzing and ranking the feature importance scores. A cut-off range [8, 10] defines an optimal extent to select features regarding their importance score. For instance, in Table 1, the cut-off range of Spearman correlation 0.084 to 0.686 demonstrates, only the features having an importance score in this threshold point will be selected with this technique. In this study, cut-off of distinct techniques is gauged manually between the high and low feature importance score i.e., poorly correlated features are excluded. The output of the filter method is a cumulative feature subset (BFF), which is fed to the wrapper method. Subsequently, applying function perturbation ensemble on the feature subsets of four filter techniques i.e., (SF1 ∪ SF2 ∪ SF3 ∪ SF4), we obtained 89 most important features (excluding target feature) from Data-I.

Table 1. Cut-off ranges of feature selection methods

Feature selection method		Cut-off range (Tk)	Selected feature
Filter	Pearson correlation (SF1)	0.087 to 0.627	68
	Chi-square (SF2)	0.00015 to 0.688	61
	Information gain (SF3)	0.0001 to 0.336	59
	Spearman correlation (SF4)	0.084 to 0.686	67
	BFF = (SF1 ∪ SF2 ∪ SF3 ∪ SF4)	–	89
Wrapper	Stepwise elimination	p-test (95% confidence)	**51**

In the wrapper method, the bi-directional elimination (aka, stepwise elimination) technique is applied to finally get the best features set. Bi-directional elimination is a combination of the forward and backward elimination techniques, which aims to find out the features' best correlation. In this technique, a statistical model is built where a p-value test is carried out to select features with peak confidence intervals. Although the computational time required to perform the wrapper method is considerably high compared to the filter method, it selects features with maximum prediction accuracy [18]. Hence, from the filtered features, the best 51 features were selected with the wrapper method conforming 95% confidence intervals. Interestingly, the features from 'URL resolving data and external metrics' are found out to be more important compared to the other groups (in Table 2).

The output feature set of the wrapper method is used as the final features of Data-I and Data-II to perform CV and prediction. As we mentioned, features of the datasets are grouped into 6 classes, in Table 2 the number of features selected from each group is listed respectively. The shape of Data-I and Data-II before feature selection was (58645,112) and (88647,112) respectively. But after applying the proposed feature selection method, the shape reduced significantly to (58645,52) and (88647,52).

Table 2. Number of features selected from each group with feature selection methods

Feature selection	Features in each group of Data-I considering URL properties						Total feature
	URL	Domain	Directory	File	Parameter	External data	
Base data	20	21	18	18	20	14	111
Filter	15	7	17	17	20	13	89
Wrapper	7	6	10	10	5	13	**51**

Afterward, the datasets are scaled with the Min-Max feature scaling method, since data in the real world does not available in a fixed range. Thus, scaling will control the bias of the features having higher weight. Besides, the scaled data allow each feature to pay a uniform contribution in optimizing the target function. In the Min-Max feature scaling method, the data are scaled in a range of 0 to 1.

Hyperparameter Optimization. In this study, we used three machine learning models i.e., Random Forest classifier, XGBoost Classifier, and Artificial Neural Networks to detect phishing websites from the datasets. Machine learning algorithms with default parameters are less likely to perform their best than algorithms with tuned parameters. Therefore, the Grid Search hyperparameter tuning technique is applied to tune the parameters of the three models. This technique used Data-I (train-80%, test-20%) to perform parameter tuning and provides optimized learning models as output. These optimized models are employed to perform cross-validation in layer-2 of the experiment. The optimized hyperparameters of the models are listed in Table 3.

Table 3. Optimized hyperparameters of the ML models

Model	Tuned parameter selected with Grid Search CV
ANN	Optimizer = Adam, learning_rate = 0.0012, epochs = 95, batch_size = 64
RFC	n_estimators = 800, criterion = 'entropy', max_depth = 75, min_samples_leaf = 1, min_samples_split = 2
XGBC	n_estimators = 1000, learning_rate = 0.1, max_depth = 5, min_child_weight = 4, subsample = 0.7, colsample_bytree = 0.8

4.2 Experimental Results Analysis and Discussions

Cross Validation. The study performed 10-folds CV using three optimized models (RFC, XGBC, ANN) on Data-I and estimated their performance evaluation metrics. The study also calculated the Mean Square Error (MSE) rate, and area under curve (AUC) score of each fold in cross-validation. Besides, the mean Receiver Operating Curve (ROC) of each model is estimated. The study made a comparative assessment considering the cross-validation performance of the models, shown in Table 4.

Table 4. Performance comparison of ML models on Data-I in CV

Fold	ANN			RFC			XGBC		
	ACC	*F1*	*MSE*	*ACC*	*F1*	*MSE*	*ACC*	*F1*	*MSE*
1	94.04	94.31	0.0449	96.01	95.98	0.0398	**96.26**	96.24	0.0374
2	93.90	94.25	0.0456	**95.72**	95.71	0.0428	95.62	95.61	0.0438

(*continued*)

Table 4. (*continued*)

Fold	ANN			RFC			XGBC		
	ACC	*F1*	*MSE*	*ACC*	*F1*	*MSE*	*ACC*	*F1*	*MSE*
3	93.89	94.18	0.0455	95.83	95.82	0.0416	**96.30**	96.29	0.0370
4	93.99	94.32	0.0449	95.87	95.86	0.0412	**96.23**	96.22	0.0377
5	94.09	94.35	0.0441	**95.95**	95.95	0.0404	95.89	95.87	0.0411
6	94.12	94.40	0.0437	95.85	95.83	0.0414	**95.92**	95.91	0.0408
7	94.15	94.43	0.0438	**96.00**	96.00	0.0399	95.72	95.71	0.0428
8	93.74	94.08	0.0455	96.16	96.14	0.0384	**96.54**	96.52	0.0346
9	93.92	94.24	0.0450	95.80	95.80	0.0419	**96.11**	96.10	0.0389
10	94.16	94.40	0.0439	95.94	95.93	0.0405	**96.25**	96.22	0.0375
Avg.	94.00	94.27	0.0447	95.91	95.90	0.0408	**96.08**	96.07	0.0392

In cross-validation, the XGBC model surpassed ANN and RFC in all possible performance evaluation metrics considered in this experiment—accuracy (ACC), for example, is 96.08%, followed by F1-score of 96.07% and the MSE of 0.039. Besides, the RFC model showed a notable performance with an accuracy of 95.91% which is close to XGBC. ANN model, on the other hand, provided a decent outcome securing an accuracy of 94% and F1-score of 94.27%. The average AUC score of XGBC, RFC and ANN is 0.9855, 0.9921, and 0.9923 respectively. Considering the true-positive and false-positive rates, ROC curves of the classifier models are illustrated in Fig. 4.

Perform Prediction. The study performed prediction on Data-II with the best model XGBC. In this circumstance, Data-II is randomly split into 'train data' (80%) and 'test data' (20%). The 'train data' is used to train the predictive model XGBC, and then it performed prediction on 'test data'. Similarly, we used the selected top 51 features for Data-II. The model XGBC concludes whether an unknown instance is legitimate or phishing. Based on that, the study evaluated the model's performance on test data by calculating the evaluation metrics, root mean square error (RMSE), kappa score, error rate, AUC score, and also estimated the ROC curve and precision-recall (PR) curve.

The XGBC model performed prediction prominently good on Data-II. The model secured significant accuracy of 97.29%, F1-score of 97.01. Besides, the model show-ed a kappa score of 94.01%, around 97% of precision and recall score, and a high AUC score of 0.996. It also gained a favorable RMSE rate of 0.1645.

In Fig. 5, the performance assessment metrics of the predictive model XGBC are illustrated with a bar graph. Besides, the confusion matrix gives the information that only 480 observations out of 17,730 testing samples are misclassified where the rest of the samples were truly classified. The true-positive rate is about 0.981, and the false-positive rate of about 0.041. The ROC curve and PR curve of the XGBC classifier model is illustrated in Fig. 6. The ROC curve is based on the true-positive and false-positive rates, where the PR curve is derived from precision and recall rates.

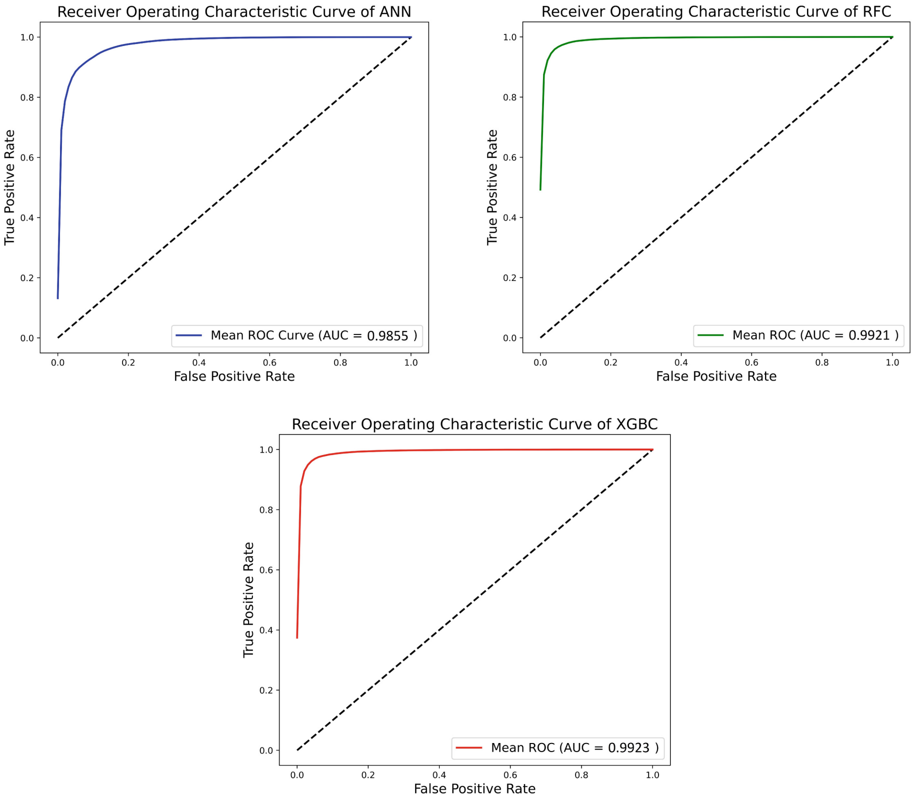

Fig. 4. Mean ROC curve of the ANN, RFC and XGBC model in CV

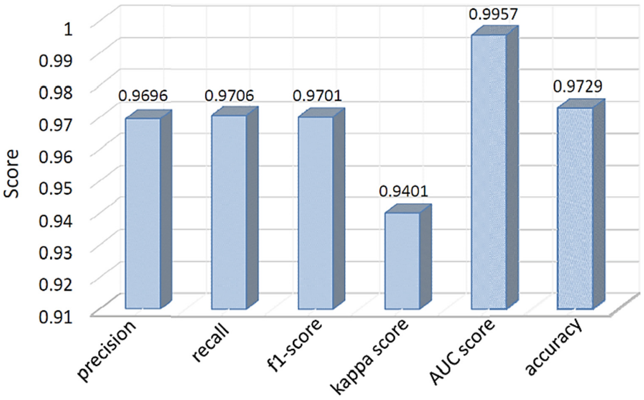

Fig. 5. Prediction performance of XGBC model on Data-II

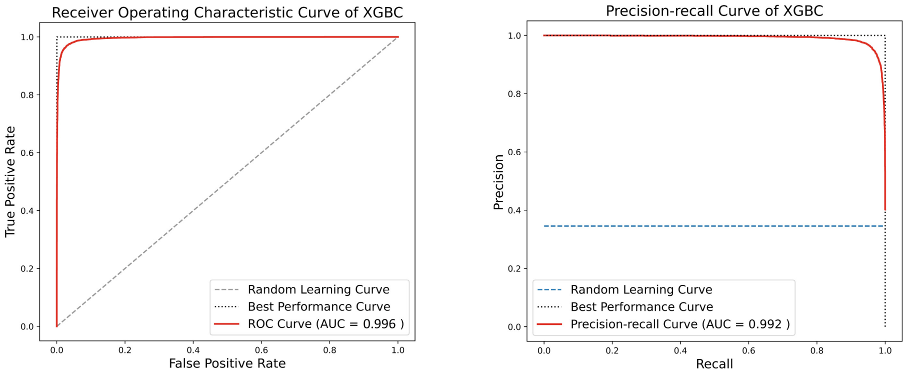

Fig. 6. ROC and PR curve of the predictive model XGBC

With this proposed study, we have achieved a considerable performance in detecting phishing and legitimate websites. As we used the latest phishing datasets for this experiment, only a few pieces of research have been conducted previously on these data. The highlighted outcome of this study is that we used only about 46% of the original features of the data. Cross-validation is performed on Data-I using three optimized machine learning models namely, Random Forest Classifier, XGBoost Classifier, and Artificial Neural Network. The XGBoost classifier model outperformed in CV, and the model is applied to perform prediction on Data-II. Now, let's compare the performance metrics of this proposed study with the existing research works.

Table 5. Performance comparison between the previous and proposed research studies

Dataset	Research reference	Best method	Features	Performance
Data-I	Grega Vrbančič et al. [11]	DNN, optimized with firefly algorithm	111	Accuracy 90.17%, F1-score 90.11%
	Pankaj Bhowmik et al. [proposed method]	XGBoost, with hybrid feature selection	**51**	Accuracy 96.08%, F1-score 96.07%
Data-II	Grega Vrbančič et al. [11]	DNN, optimized with firefly algorithm	111	Accuracy 94.39%, F1-score 93.83%
	Mohammed Al-Sarem et al. [2]	Stacking ensemble, GA-based optimization	111	Accuracy 97.39%
	Pankaj Bhowmik et al. [proposed method]	XGBoost, with hybrid feature selection	**51**	Accuracy 97.29%, F1-score 97.01%

Discussion. On Data-II, the stacking model by Al-Sarem et al. performed slightly better compared to our model since their model used all the 111 features of the dataset, but our proposed model used only 51 of the original features. Besides, the stacking model ensemble the strength of the four different ML models—on the other hand, we developed an individual model. Overall, from the comparative analysis in Table 5, we can see that the proposed model XGBC outperformed using only 45.95% features.

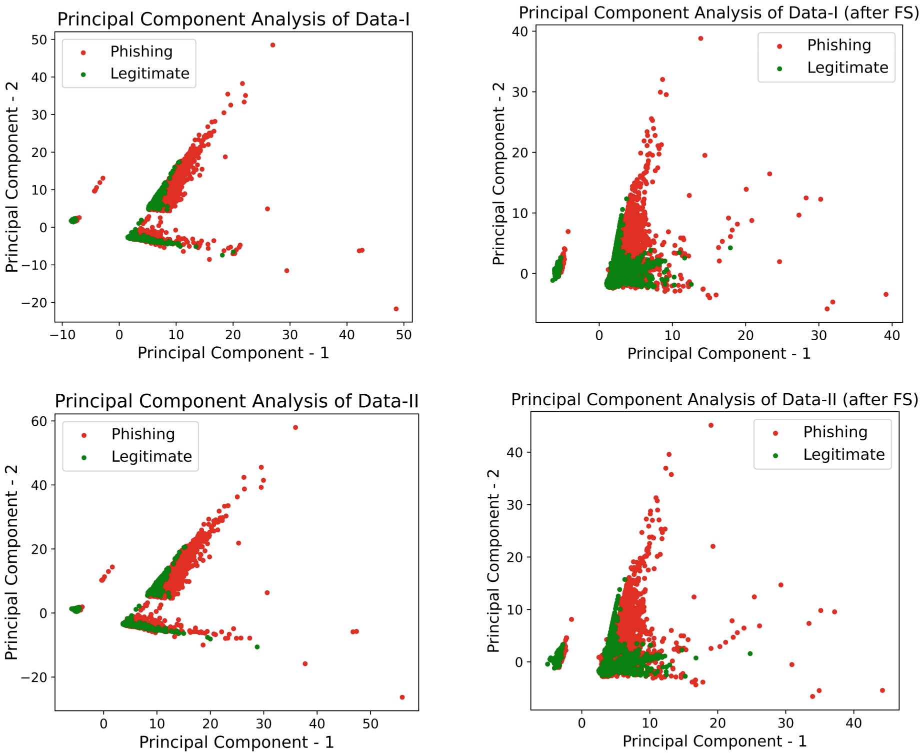

Fig. 7. PCA of the datasets before (left) and after (right) applying hybrid feature selection

Responding to **RQ-1**, the study performed principal component analysis (PCA) on the datasets (Data-I, Data-II) before and after feature selection (FS), illustrated in Fig. 7. The findings of PCA showed that hybrid feature selection reduced the overlapping of phishing and legitimate class cases in the datasets. Hence, it has facilitated the learning algorithms in decision-making and fit decision boundaries more accurately in the feature space. Besides, since the dimension of datasets is reduced, the complexity and degree of computations of models are also minimized. In response to **RQ-2**, considering the feature size of Data-I and Data-II, the proposed framework comparatively outplayed the existing studies. The outcomes of this study revealed that the hybrid framework had supported the ML models in improving the overall performance.

In this study, we endeavored to improve the outcomes of ML models utilizing a minimal number of features (only the best features) from the dataset. However, the well-organized empirical framework facilitated this experiment to achieve a notable performance—essentially, the proposed hybrid feature selection approach and the ML models with optimized hyperparameters are the fundamental factors.

5 Conclusion and Future Works

In this article, we introduced an empirical hybrid feature selection approach to leverage the performance of machine learning models in phishing websites detection. We used the two latest URL-based phishing datasets in the study. The proposed methodology of this experiment has three layers, including Feature Engineering, Cross Validation, and Perform Prediction. In the Feature Engineering layer, applying the hybrid feature selection method we derived only 51 features (out of 111) from the dataset. In this method, the raw features are refined through the filter method and the wrapper method accordingly. The hybrid approach used five distinct (4 filters and 1 wrapper) feature selection techniques in total. Following that, we optimized the proposed models (ANN, RFC and XGBC) with Grid Search based hyperparameter tuning. During the Cross Validation layer, we performed 10-folds cross-validation using the optimized models on Data-I. The result showed XGBC came up with the maximum prediction accuracy of 96.08%, F1-score of 96.07% and with MSE rate of 0.392. Finally, we performed prediction on Data-II using the best model XGBC. The model showed a significant performance with securing the accuracy of 97.29%, F1-score of 97.01%, kappa score of 94.02%, and RMSE of 0.1645 on the test data. Although the number of features of the datasets is cut down to about 46%, the proposed method outperformed the previous studies [2, 11].

Considering the proposed hybrid feature selection method, since the dimension of the datasets is reduced, the models performed notably well. Besides, their complexity and degree of computations are also minimized. We will endeavor to apply the proposed framework on different available phishing datasets and experiment with state-of-art deep neural networks in the future. In Table 1, the ad-hoc mounting of the cut-off ranges of feature selection techniques revealed promising remarks on the model's performance. Besides, we will resume the study to design an automatic assignment method of the best cut-off ranges for the feature selection techniques.

References

1. Somesha, M., Pais, A.R., Rao, R.S., Rathour, V.S.: Efficient deep learning techniques for the detection of phishing websites. Sādhanā **45**(1), 1–18 (2020). https://doi.org/10.1007/s12046-020-01392-4
2. Al-Sarem, M., et al.: An optimized stacking ensemble model for phishing websites detection. Electronics **10**(11), 1285 (2021). https://doi.org/10.3390/electronics10111285
3. Kalaharsha, P., Mehtre, B.M.: Detecting Phishing Sites – An Overview. arXiv:2103.12739v2 (2021)
4. Sarma, D., et al.: Comparative analysis of machine learning algorithms for phishing website detection. In: Smys, S., Balas, V.E., Kamel, K.A.., Lafata, P. (eds.) Inventive Computation and Information Technologies. LNNS, vol. 173. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-4305-4
5. da Silva, C.M.R., Feitosa, E.L., Garcia, V.C.: Heuristic-based strategy for phishing prediction: a survey of URL-based approach. Comput. Secur. **88**, 101613 (2020)
6. Zuraiq, A.A., Alkasassbeh, M.: Review: phishing detection Approaches. In: 2nd International Conference on new Trends in Computing Sciences (ICTCS), pp. 1–6 (2019)

7. Alsariera, Y.A., Adeyemo, V.E., Balogun, A.O., Alazzawi, A.K.: AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access **8**, 142532–142542 (2020). https://doi.org/10.1109/ACCESS.2020.3013699
8. Chiew, K.L., et al.: A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Inf. Sci. **484**, 153–166 (2019). https://doi.org/10.1016/j.ins.2019.01.064
9. Ali, W.: Phishing website detection based on supervised machine learning with wrapper features selection. Int. J. Adv. Comput. Sci. Appl. **8**(9) (2017). https://doi.org/10.14569/IJACSA.2017.080910
10. Pes, B., Dessì, N., Angioni, M.: Exploiting the ensemble paradigm for stable feature selection: a case study on high-dimensional genomic data. Inform. Fus. **35**, 132–147 (2017). https://doi.org/10.1016/j.inffus.2016.10.001
11. Vrbančič, G., Fister, I., Jr., Podgorelec, V.: Parameter setting for deep neural networks using swarm intelligence on phishing websites classification. Int. J. Artif. Intell. Tools **28**(06), 1960008 (2019). https://doi.org/10.1142/S021821301960008X
12. Adeyemo, V.E., Balogun, A.O., Mojeed, H.A., Akande, N.O., Adewole, K.S.: Ensemble-based logistic model trees for website phishing detection. In: Anbar, M., Abdullah, N., Manickam, S. (eds.) ACeS 2020. CCIS, vol. 1347, pp. 627–641. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-6835-4_41
13. Vaitkevicius, P., Marcinkevicius, V.: Comparison of classification algorithms for detection of phishing websites. Informatica **31**(1), 143–160 (2020). https://doi.org/10.15388/20-INFOR404
14. Korkmaz, M., Sahingoz, O.K., Diri, B.: 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 (2020)
15. Hannousse, A., Yahiouche, S.: Towards benchmark datasets for machine learning based website phishing detection: an experimental study. Eng. Appl. Artif. Intell. **104**, 104347 (2021). https://doi.org/10.1016/j.engappai.2021.104347
16. Vrbančič, G., Fister, I., Jr., Podgorelec, V.: Datasets for phishing websites detection. Data Brief **33**, 106438 (2020). https://doi.org/10.1016/j.dib.2020.106438
17. Vrbančič, G.: Phishing websites dataset. Mendeley Data. **V1**,(2020). https://doi.org/10.17632/72ptz43s9v.1
18. Mochammad, S., et al.: Stable hybrid feature selection method for compressor fault diagnosis. IEEE Access **9**, 97415–97429 (2021). https://doi.org/10.1109/ACCESS.2021.3092884