Phishing web site detection using diverse machine learning algorithms

Ammara Zamir
Department of Computer Science, University of Wah, Quaid Avenue, Wah Cantt, Pakistan and Department of Computer Science, COMSATS University Islamabad – Wah Campus, Islamabad, Pakistan

Hikmat Ullah Khan and Tassawar Iqbal
Department of Computer Science, COMSATS University Islamabad, Wah Campus, Islamabad, Pakistan

Nazish Yousaf
Department of Computer Science, University of Wah, Quaid Avenue, Wah Cantt, Pakistan and Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, Islamabad, Pakistan

Farah Aslam
Department of Computer Science, University of Wah, Quaid Avenue, Wah Cantt, Pakistan

Almas Anjum
Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, Islamabad, Pakistan, and

Maryam Hamdani
Department of Computer Science, University of Wah, Quaid Avenue, Wah Cantt, Pakistan

Received 14 May 2019
Revised 8 September 2019
24 October 2019
Accepted 6 November 2019

Abstract

Purpose – This paper aims to present a framework to detect phishing websites using stacking model. Phishing is a type of fraud to access users' credentials. The attackers access users' personal and sensitive information for monetary purposes. Phishing affects diverse fields, such as e-commerce, online business, banking and digital marketing, and is ordinarily carried out by sending spam emails and developing identical websites resembling the original websites. As people surf the targeted website, the phishers hijack their personal information.

Design/methodology/approach – Features of phishing data set are analysed by using feature selection techniques including information gain, gain ratio, Relief-F and recursive feature elimination (RFE) for feature selection. Two features are proposed combining the strongest and weakest attributes. Principal component analysis with diverse machine learning algorithms including (random forest [RF], neural network [NN], bagging, support vector machine, Naïve Bayes and k-nearest neighbour) is applied on proposed and remaining features. Afterwards, two stacking models: Stacking1 (RF + NN + Bagging) and Stacking2 (kNN + RF + Bagging) are applied by combining highest scoring classifiers to improve the classification accuracy.

The Electronic Library
Vol. 38 No. 1, 2020
pp. 65-80

0264-0473
DOI 10.1108/EL-05-2019-0118

Findings – The proposed features played an important role in improving the accuracy of all the classifiers. The results show that RFE plays an important role to remove the least important feature from the data set. Furthermore, Stacking1 (RF + NN + Bagging) outperformed all other classifiers in terms of classification accuracy to detect phishing website with 97.4% accuracy.

Originality/value – This research is novel in this regard that no previous research focusses on using feed forward NN and ensemble learners for detecting phishing websites.

Keywords Classification-based techniques, Ensemble learners, Feed forward neural network, Phishing detection, Neural networks, Stacking models, Ensemble techniques, Feature selection, Malicious URLs

Paper type Research paper

1. Introduction

In recent years, social networks have become a virtual meeting place for the general public. Unfortunately, while connecting through social networks, people experience phishing attacks. Phishing is a cybercrime which risks a user's privacy, may execute malware attacks and often steals their sensitive information. Phishing is carried out by using different engineering techniques including: instant messages (Jakobsson, 2018); fraudulent emails or mimicking an online bank, auction or payment sites; and directing people to fake Web pages (Rodríguez *et al.*, 2019) that resemble a login page to a genuine site. Phishing attacks have increased drastically in 2019, according to the Anti-Phishing Working Group which detected the total number of phishing websites in 2019 as 180,768 (Anti-Phishing Working Group, APWG, 2019). Also, according to a Proofpoint survey, people who use social websites are more exposed to potential phishing threats (www.proofpoint.com/us/security-awareness/post/latest-phishing-first-2019, accessed 8 May 2019).

A website phishing attack is carried out by spoofing legal identities, such as a legitimate website. A malicious website succeeds at obtaining some user information and can lead a user to additional malicious links that consequently gain access to even more of the user's sensitive or personal information. To achieve this goal, identical websites are created which so closely resemble the original website that the duplicitous duplication cannot be detected. Phishing attacks cause great economical, intellectual property and national security damages (Vayansky and Kumar, 2018).

Phishing spoils industries including e-commerce and internet banking. Several techniques exist to save users from phishing attacks, including the heuristic approach (Babagoli *et al.*, 2019), the rule-based approach (Adewole *et al.*, 2019) and a supervised machine learning (ML) approach (Sahingoz *et al.*, 2019). Supervised ML algorithms are widely used for classification (Alzu'bi *et al.*, 2018; Hawashin *et al.*, 2019) and are more popular among all the techniques used to detect phishing websites. Kumar and Chaudhary (2017) introduced a framework based on machine learning for e-commerce-based mobile applications to detect malwares. This approach detects mobile phishing and protects information leakage (Kumar and Chaudhary, 2017). Internet banking is also effected by phishing. The rule-based approach was introduced by Moghimi and Varjani (2016) to detect phishing in internet banking using four sets of features. Support vector machine (SVM) is applied to classify the Web pages, and the proposed framework achieved 99 per cent accuracy in detecting Web pages for phishing in internet banking (Moghimi and Varjani, 2016).

This research study focusses on a supervised ML approach to detect phishing websites. The contributions of this research study are as follows:

- Application of feature selection algorithms including: gain ratio (GR), information gain (IG), Relief-F and recursive feature elimination (RFE) to analyse the importance

of highest ranking features on a data set of phishing sites containing 11,055 websites with 32 attributes.

- Proposing two new features based on the weakest and strongest features to increase the classification accuracy of classifiers.
- Comparison of various supervised learning algorithms including random forest (RF), SVM, bagging, kNN, neural network (NN) and j48 on a combination of different feature sets.
- Stacking of the highest performing algorithms to increase the classification accuracy.
- Performance evaluation measurements, such as accuracy, *f*-measure, recall, and precision, are used to assess the performance of all classifiers.

The rest of the paper is organized as follows. Section 2 reviews the previous work and techniques used to detect Web phishing. Section 3 shares the techniques applied in this study. Section 4 discusses the results, and Section 5 presents the conclusions of this study.

2. Related work

This section provides a review of the relevant earlier research work of phishing attacks in general and focusses on the classification techniques applied to detect Web phishing in particular.

2.1 Phishing attack methods

Spoofing email is a type of phishing in which phishers send an email to a user using a fake email address, thereby tricking people into opening the message (Gunawardena *et al.*, 2013; Gupta *et al.*, 2016; Mujtaba *et al.*, 2017). Phishers then manipulate users to get their private information. Email spoofing is carried out by using simple mail transfer protocol (SMTP). In this kind of attack, phishers use spoofed email which mimics a legal identity. This attack is often carried out to steal data from well-known organizations (Caputo *et al.*, 2014). Phishers also create fake accounts on social sites, such as Facebook, Twitter, Gmail and LinkedIn. People then share their personal information on these social sites without regard to privacy issues. Phishers send requests to people by portraying themselves as a legal identity (Allen *et al.*, 2012). On a website, a phishing uniform resource locator (URL) plays an important role. A set of features are provided to correctly classify URLs. Phishers can attack websites by stealing an organization's financial assets too. This problem can be reduced by blacklisting and detecting phishing URLs (Xiang *et al.*, 2011). A Trojan horse is a threat to system privacy when a user clicks on a file; the main function implements an action. Many job advertisements require the person to enter their personal information. Scammers create illegal banking sites by displaying better interest rates than other banks. People who complete the information then fall prey to phishing through a Trojan horse (Wadhwa and Arora, 2017).

2.2 Classification techniques to detect website phishing

To date many techniques have been introduced to get rid of phishing attacks and to provide safety to online users. Spoofed emails and fake URLs are difficult to detect and are unstoppable. To stop phishing, the best way is to block harmful emails and fake URLs. Mamun *et al.* (2016) proposed an approach to detect and classify harmful URLs. A proactive approach is introduced to detect malicious URLs using lexical analysis. A

feature set is proposed to effectively categorize the harmful URLs. Then, an analysis has been made on the obfuscation technique to mitigate these URLs (Mamun *et al.*, 2016).

Fette *et al.* (2007) introduced a method based on ML techniques named PLIFER. This method requires age of domain of URLs (?). Furthermore, ten features are extracted and RF is applied to detect a phishing website. This method correctly classified 96 per cent of phishing emails. Classification techniques are also used to detect phishing, using labelled data sets. Different classification techniques use features including URL-based and textual-based features. URL-based features include IP address, domain name, geographic properties as input for the classification of phishing, using ML classifiers, K-nearest, Naïve Bayes (NB) and SVM (Blum *et al.*, 2010; Khonji *et al.*, 2011). Islam and Abawajy (2013) presented a multi-tier approach for detecting spoofed emails. The email features are computed on the basis of the weight of header and body contents. A total of 21 features are extracted, and the classification algorithms of SVM, AdaBoost and Naïve Bayes are applied. The process is divided into three phases. First, each email is classified and then algorithms are applied. Afterwards, spam email is sent to a folder in the end. This framework achieved 97 per cent accuracy and reduced the false positive rate (Islam and Abawajy, 2013).

Welch *et al.* (2011) drew out features from the content and the URL to detect phishing. Ensemble classifiers are used to detect phishing in the email environment. The authors also introduced an approach including static and run-time features of a Web page (Welch *et al.*, 2011). Static Web pages are used to identify harmful Web pages. These malicious Web pages are investigated further. This approach uses run-time features at the end of the final investigation. This toolkit is effective for the detection of phishing attacks. In other research, Zhao *et al.* (2016) presented two-stage classification to detect harmful pages. Static features are low cost and hence are used to detect harmful Web pages in the first stage and then these Web pages are forwarded to the next step. Run-time features are expensive while used in the last step to identify phishing. The study explores the classification algorithm performance to detect phishing websites (Zhao *et al.*, 2016). Bhattacharjee *et al.* (2017) extracted textual features, document object model structure and component files from the main page for the detection of phishing. These features are classified through ML algorithms (Bhattacharjee *et al.*, 2017).

Kumar and Chaudhary (2017) proposed an approach which uses hyperlinks in pages of a website. Hyperlink features are used to detect a phishing website. Several ML approaches are used along with the hyperlinks feature set. ML techniques are applied on phishing and non-phishing data sets. The proposed approach is language independent and detects phishing up to 98 per cent accuracy (Jain and Gupta, 2019). Chiew *et al.* (2019) proposed a features selection model *hybrid* ensemble *feature selection* (HEFS) to detect phishing websites based on ML algorithms. To extract the primary feature set, a cumulative distribution gradient algorithm is applied. Then, a function named *data perturbation* ensemble is used to extract the second feature set. Afterwards, RF, an ensemble learner, is applied to detect phishing websites. The results show that HEFS detected phishing attributes with up to 94.6 per cent accuracy (Chiew *et al.*, 2019).

Nagaraj *et al.* (2018) proposed an ensemble ML model for classification of phishing websites. The technique was made twofold, first by applying RF classifier and then by integrating the obtained results with a feedforward NN. K-fold cross validation has been

applied to validate the performance of the ensemble. The results showed an accuracy of 93.41 per cent with the RF_NN model on a publically available data set (Nagaraj *et al.*, 2018).

In other research, a stacking model has been proposed by Li *et al.* (2019) using HTML and URL features for detection of phishing webpages. Gradient boosted decision tree, light gradient boosting machine (LightGBM), and XGradientBoost have been combined to devise a stacking model. The proposed approach has been applied on two publically available data sets and provided 97.3 per cent accuracy (Li *et al.*, 2019). Sahingoz *et al.* (2019) implemented a real-time anti-phishing system for detection of phishing URLs. The proposed approach uses seven classification algorithms [decision tree, K-star, AdaBoost, kNN (n = 3), SMO, RF, and Naïve Bayes] and different types of natural language processing-based features. RF has been used with NLP-based features. The proposed technique outperformed the previous works with 97.98 per cent accuracy on the data set constructed by the authors themselves which comprised 73,575 URLs (Sahingoz *et al.*, 2019).

3. Methodology

This section discusses the proposed framework, applied ML algorithms, feature selection techniques, selected data set and performance evaluation measures used to carry out the experiments. Figure 1 shows the proposed model of the experimental setup. In the first step, a phishing website data set is selected. Then top attributes are analysed by feature selection algorithms, such as IG, GR, Relief-F and RFE. After analysing the importance of features, the weakest features are combined in a new feature N1, and the strongest features are combined in N2. All features are transformed into normalized form. Afterwards, features are fed to ML classifiers with principal components analysis (PCA). Stacking is performed by combining the highest performing algorithms. Features are trained using ten-fold cross validation. Performance evaluation measures are used to evaluate the performance of classifiers for phishing website detection. The proposed framework is shown in Figure 1.

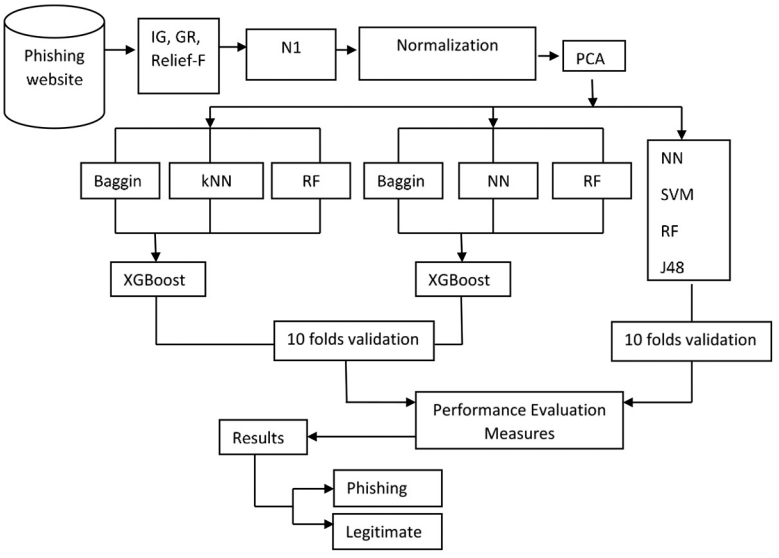

Figure 1.
The proposed framework for phishing detection

Algorithm 1: Algorithm for stacking and classification

```
Start
Input X                ▷ X = website dataset
Output Y               ▷ Y = predicted class label (phishing or legitimate)
Rx=FeatureReduction(X)
Rx=Normalize(Rx)
Rx=PrincipalComponentAnalysis(Rx)
For(i = 1; i<=10; i++)
    Predict1 = Classify(Rx)
        Predict1= ComputeGradient(Predict1)//for classification removestep 9
    If (Predict1 < 0)
        Y=−1
    else
        Y = 1
        break
    end if
End for
Return Y
Stop
```

3.1 Applied machine learning techniques

This section discusses the algorithms applied on phishing websites data set.

3.1.1 Support vector machine. SVM is a widely used algorithm offered by supervised machine learning methods. This algorithm is used in engineering and science applications. The basic purpose of SVM is to build models which further generate a linear classifier. SVM performs non-linear classification effectively. SVM executes non-linear classification by using kernel trick to map inputs into high-feature space (Alzu'bi *et al.*, 2019).

3.1.2 Naïve Bayes. Naïve Bayes is a simple probabilistic classifier based on conditional probability. This classification algorithm uses basic Bayesian theorem and propagates a firm independence that is present between features. Occurrence of features along with their inter-relations is calculated in corpus consideration. It uses one class label. In NB, correlation of all neglected attributes is considered to be independent (Abooraig *et al.*, 2018).

3.1.3 Random Forest. RF is a recognized learning algorithm owing to its broad usage in literature to achieve classification and regression successfully with higher performance (Abooraig *et al.*, 2018). The algorithms execute in a way that each predictor of the tree depends on random values, whereas decision of classification originates by taking a vote from all decision trees. By the ensemble method, a number of trees RF achieve enhanced accuracy for classification.

3.1.4 K-nearest neighbour. kNN is known as a classic supervised learning classifier that works on the basis of analogy-based classification. This classifier addresses classification and regression issues in an efficient manner. On providing test samples to kNN, it searches in n dimensional space to find K-closest neighbour by measuring the distance. The K-closest neighbour is used to make decisions of classification.

3.1.5 Bagging. Bagging (Bootstrap aggregation) is one of the most prevailing ensemble meta-algorithms. This algorithm increases the accuracy of ML algorithms. It is a general procedure that is applied in static classification and regression problems to reduce variance and over fitting.

3.1.6 Neural network. NN is a technique of non-linear classification that uses back propagation for training purposes. NN contains interconnected nodes. Each node is known as a *perceptron* which feeds the signals to activation function. This technique works by labelling the required class for each sample to get compared by the NN output. The activation function is calculated as shown in equation (1).

$$x(w_i) = tanh(w_i) \tag{1}$$

Here *tanh* represents the optimization function for each sample of w.

3.2 Feature reduction methods

The following feature reduction algorithms are used to reduce features from the phishing data set.

3.2.1 Information gain. IG is basically mutual information gained by calculating probability for feature selection. It also decreases the bias in multivariate features (Hawashin *et al.*, 2013). Following is the formula used for IG shown in equation (4).

$$h(class) = -\sum_{class_i \in class}^{n} p(class_i) log\, p(class_i) \tag{2}$$

$$\mathrm{h(class|feature)} = -\sum_{\mathrm{fk_i \in feature}}^{n} \mathrm{p(fk_i)} \sum_{\mathrm{class_i \in class}}^{n} \mathrm{p(class_i \mid fk_i) log(p(class_i|fk_i))} \tag{3}$$

$$\mathrm{Infomation\,Gain = h(class) - h(class|feature)} \tag{4}$$

In the above equations, $\boldsymbol{p}(\boldsymbol{class_i})$ represents $class_i$ probability, $p(fk_i)$ represents the probability of a certain feature i, $\boldsymbol{p}(\boldsymbol{class_i \mid fk_i})$ represents a specific class i probability at a certain feature i.

3.2.2. Gain ratio. An extension of IG is GR. GR is used to overcome the biasness of IG. Using split information, GR applies normalization to IG, as shown in equations 5 and 6.

$$Splitinfo_A\,(D) = \sum_{i=1}^{m} \frac{|D_i|}{|D|} X\, log_2 \frac{|D_i|}{|D|} \tag{5}$$

$$GainRatio = \frac{Gain(A)}{SplitInfo(A)} \tag{6}$$

3.2.3 Relief-F. Relief-F is a widely used feature selection technique and reliable algorithm for probability estimation. It is also used for the purpose of binary classification. The major advantage of using Relief-F is less time consumption in its execution. In Relief-F, heuristics are not considered and hence no dependence on heuristics is found. Relief-F can be applied to

binary and continuous data. It executes by separating important features from neighbour classes using equation (7).

$$Wf = P(\text{differentvalueofclosestfeaturesofdifferentclasses}) - P(\text{differentvalueofclosestinstancefromsomeclass}) \quad (7)$$

3.2.4 Principal components analysis. PCA is a technique that statistically transforms the data set coordinates into completely novel coordinates known as *principal components*. This technique is most conventionally and widely used in the fields of biosciences, image processing (Alzu'bi *et al.*, 2011) and control systems. It works by excluding the minor components in a great number to obtain principal components in a small number on a low and linear dimensional subspace.

3.2.5 Recursive feature elimination. RFE is a feature reduction technique used to eliminate the weakest features from a data set. RFE removes the least important features recursively and ranks the features according to their importance. The removal of the least important features results in decreasing the error rate (Rado *et al.*, 2019).

3.3 Data set

The phishing websites data set is freely available for research purposes (www.kaggle.com/akashkr/phishing-website-dataset#dataset.csv, accessed 8 January 2019). The data set comprises 11,055 websites, and 32 attributes have already been used in earlier research work, such as Ibrahim and Hadi (2017), Shirsat (2018) and Tyagi *et al.* (2018). The data set features are shown in Table I.

3.4 Performance evaluation measures (PEM)

For predicting classifier accuracy, several PEM are used on the phishing website data set. These measures are accuracy, recall and precision which are defined in equations 8, 9 and 10, respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{TP + TN + FP + FN} \quad (8)$$

Table I.
Features of the phishing data set

F1	Having_IP_Address	F17	SFH
F2	URL_length	F18	Submitting_to_email
F3	Shortening Service	F21	Redirecting
F4	Having_at_Symbol	F22	RightClick
F5	Double_slash_redirecting	F23	Pop_up_window
F6	Prefix_sufix	F24	Iframe
F7	Having_sub_domain	F25	Age_of_domain
F8	SSL_final_state	F26	DNSRecord
F9	Domain_registration_length	F27	Web_traffic
F10	Favicon	F28	Page_rank
F11	Port	F29	Google_index
F12	Http_token	F30	Link_Point_to_Page
F13	Request_URL	F31	Statistical_report
F14	URL_of_anchor	F32	Result
F16	Links_in_tag		

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

TP represents all true positive outcomes which the model has predicted correctly as a positive class, *TN* represents all true negative outcomes which the model has predicted correctly as a negative class, *FP* represents all false positive outcomes which the model has predicted incorrectly as a positive class, and *FN* represents all false negative outcomes which the model has predicted incorrectly as a negative class.

4. Results and discussions

This section discusses the feature analysis, data analysis and experimental results carried out on the selected data set of phishing detection.

4.1 Feature analysis

Several experiments have been carried out to analyse the individual features and to observe their correlation with result. The feature selection methods IG, GR, Relief-F and RFE are used. SSLfinal_State, URL_of_Anchor, Web_traffic and Prefix suffix are the most correlating attributes with results. Their values are shown in Table II: among these four attributes, Web traffic and URL_of_Anchor are correlated with each other. While other attributes, PopUpWindow, on_mouse_over, Iframe, favicon, port and submitting_to_email are the least correlating features with results shown in Table III.

4.2 Data analysis

Various feature selection techniques were applied over the data set to find the importance of features. It was observed that the following attributes were the most important features according to their IG, GR and Relief-*F* values.

Features	Correlation
SSL_final_state	0.71
URL_of_Anchor	0.69
Web_traffic	0.35
Prefix_suffix	0.35

Table II. Highest correlating features with results

Features	Correlation
Favicon	0.0002
Pop_Up_Window	3.6×10^{-5}
On_Mouse_Over	0.04
Iframe	0.003
Submitting_to_Email	0.01

Table III. Least correlating features

- Server form handler (SFH)
- PopUpWindow
- SSLfinal_State
- Request_URL
- URL_of_Anchor

RFE is also applied to analyse the importance of the feature set. The rank of all features was found to be same, that is, 1 except for the features; port, Iframe, right_click and on_mouse_over.

These attributes have the same values range from $\{-1, 0, 1\}$. Here 1 implies legitimate, 0 implies suspicious, and -1 implies phishy. When a user submits information on a Web page, this information is transferred to the authentication server from where the Web page is loaded. However, a phisher redirects the link to another URL link, which is different from the right URL. If a site is asking users to submit their personal information using a popup window, then there is a possibility that the site is phishy. Phishers use fake HTTP protocols to deceive users. If objects are loading from a different URL, rather than the requested URL, the site is probably phishy. A Web page where links direct to a different domain other than the typed URL is phishy.

4.2.1 Attributes visualization. Figure 2 presents the comparison of SFH and the result attribute. The SFH attribute has more phishy values. The graph is getting wider and smaller where values are legitimate, whereas the graph is getting thinner and higher where values are getting more phishy.

The PopUpWindows is the second highest ranked attribute among the other attributes. The PopUpWindow attribute comprises all phishy values shown in Figure 3. The visualization shows that PopUpWindows are mostly harmful in websites to steal user credentials while using any website.

SSLfinal_State comprises more legitimate values than phishy values shown in Figure 4. The graph consisting of more legitimate values is higher and thinner than that of phishy values.

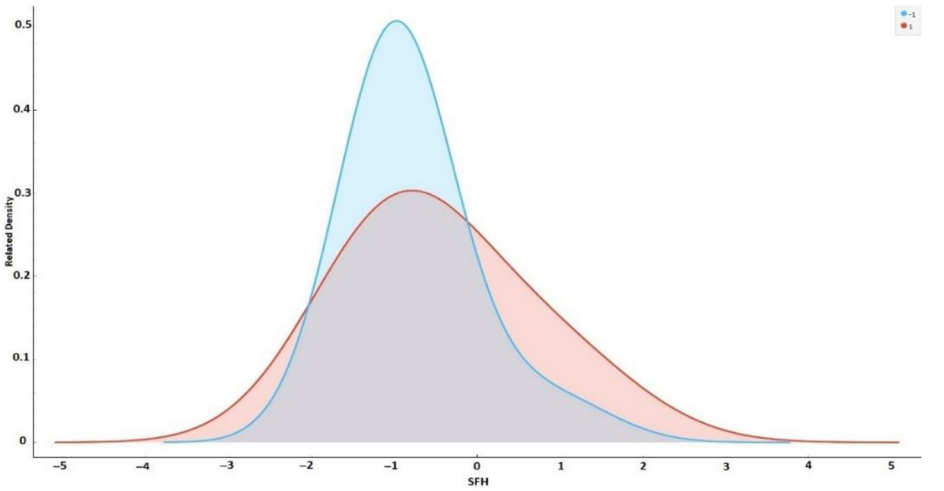

Figure 2.
SFH and the result feature visualization

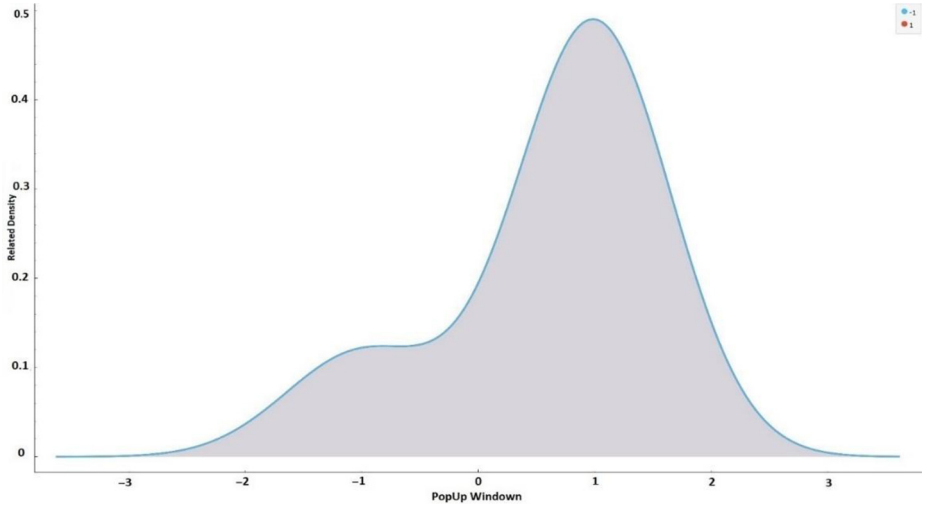

Figure 3.
PopupWindows and the result visualization

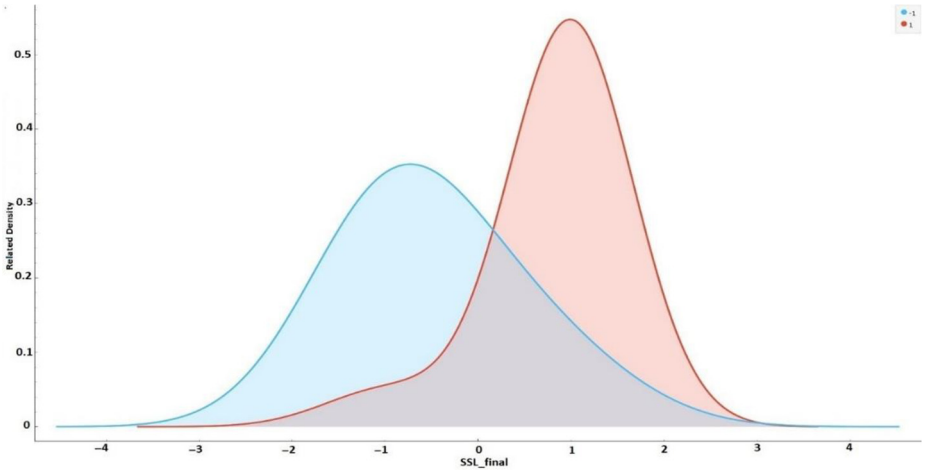

Figure 4.
SSLfinal_State and the result visualization

4.3 Classification results and discussion

Here the results of the experiments, parameter settings, results of two stacking models and proposed features are discussed.

After performing a detailed analysis on the feature set, two features are proposed, one is the combination of the weakest features and the other is the combination of the strongest features, as shown in Table IV. To evaluate the performance of the classifiers, all of the features of the data set along with the two proposed features are normalized. After normalization, PCA is applied along with diverse ML algorithms with settings of tenfold cross validation. Besides this, two stacking models are also applied on the selected feature set.

Results of diverse algorithms without PCA and any new technique are shown in Table V. The data set is fed to diverse classifiers without normalization. Results are computed using tenfold cross validation settings. RF outperformed all other algorithms in terms of classification accuracy, as shown in Table V.

Furthermore, the results are evaluated on the proposed features with a combination of other features of the data set. The weakest features including F11, F20, F22 and F24 are removed. Algorithms are applied with PCA on the remaining features with N1 and N2. Besides this, two stacking models Stacking1 (RF + NN + bagging) and Stacking2 (kNN + RF + bagging) are also applied to evaluate the performance of classifiers. Stacking1 (NN + RF + bagging) outperformed all other classifiers in terms of classification accuracy with 97.4 per cent shown in Table VI.

Table IV.
Proposed features

N1: Combined weakest features	N2: Combined strongest features
N1	*N2*
Port	
Favicon	
Submitting_to_Email	Web_traffic
Pop_up_Window	URL_of_anchor
On_mouse_over	
Iframe	

Table V.
Results of the classifier without PCA

Algorithms	Accuracy	Precision	Recall	*F*-measure
All features				
NB	0.7267	0.737	0.651	0.69
k-NN	0.942	0.925	0.939	0.93
SVM	0.931	0.917	0.946	0.93
RF	*0.969*	*0.969*	*0.955*	*0.96*
Bagging	0.951	0.938	0.948	0.94
NN	0.958	0.967	0.958	0.96

Table VI.
Results of classifier of original features with N1 and N2 with PCA

Removed features	Algorithms	Accuracy	Precision	Recall	*F*-measure
11, 24, 20, 22	NB	0.746	0.988	0.638	0.775
	k-NN	0.953	0.939	0.954	0.946
	SVM	0.947	0.925	0.954	0.939
	RF	0.973	0.961	0.976	0.968
	Bagging	0.969	0.959	0.971	0.965
	NN	0.972	0.963	0.975	0.969
	Stacking1 (NN+RF+bagging)	*0.974*	*0.960*	*0.981*	*0.970*
	Stacking2 (knn+RF+bagging)	*0.972*	*0.955*	*0.981*	*0.968*

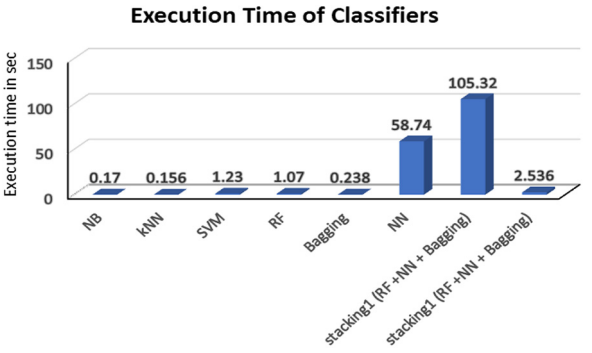

Figure 5.
Execution time for classifiers

NB	priors=None, var_smootding=1e-09
KNN	n=3
SVM	C=1.0, kernel='rbf', degree=3, gamma='auto_deprecated', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=−1, decision_function_shape='ovr', random_state=None
RF	(n_estimators='10, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None)
Bagging	(base_estimator=None, n_estimators=10, max_samples=1.0, max_features=1.0, bootstrap=True, bootstrap_features=False, oob_score=False, warm_start=False, n_jobs=None, random_state=None, verbose=0)
NN	NN=NNClassifiers(max_iter=400, hidden_layer_sizes=(300,300,300), activation='tanh',alpha=0.001,learning_rate='adaptive') cv=KFold(n_splits=10, random_state=1, shuffle=False)
XGBoost (stacking)	(loss='deviance', learning_rate=0.01, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, min_impurity_split=None, init=None, random_state=None, max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False, presort='auto', validation_fraction=0.1, n_iter_no_change=None, tol=0.0001) sclf=StackingClassifier(classifiers=[model11, model22,model33], use_probas=True,meta_classifier=models)

Table VII.
Parameter settings for applied ML algorithms

Techniques	Highest achieved accuracy (%)
PWP using RF classifiers (Ibrahim and Hadi (2017)	95.2
Intelligent detection using RF (Subasi *et al.* (2017)	97.3
Proposed method using stacking	97.4
Proposed method using RF	97.3

Note: PWP = phishing web sites prediction

Table VIII.
Comparison with existing techniques

Stacking1 (NN + RF + bagging) consumes more time (i.e. up to 105.32 s) to classify but has the best classification rate among all the classifiers. NB has less execution time with less classification accuracy. Execution time of all classifiers is shown in Figure 5. The results of the proposed method are compared with existing techniques. The results are shown in Table VIII. Results are computed on a core i7 8th generation with 8 Gb RAM and 1 TB HDD. Parameter settings for optimizing all classifiers are shown in Table VII.

4.4 Comparison with other techniques

The proposed method is compared with existing techniques and shown in Table VIII. The proposed method achieved the best accuracy on the selected data set by using Stacking1 (NN + RF + bagging). Previous approaches use RF to achieve the highest classification accuracy while the proposed technique includes RF in combination with bagging and NN, the highest performing classifiers, to improve the classification results.

5. Conclusion

This research study presents a comparison of supervised learning approaches and stacking model to detect phishing websites. The main contribution of this research is to improve the classification accuracy through proposed features with PCA and stacking of highest performing classifiers. Stacking1 (RF + NN + bagging) outperformed all other classifiers with proposed features N1 and N2 and achieved 97.4 per cent accuracy.

The research is performed on the phishing websites data set. The data set contains 32 pre-processed attributes with 11,055 web hits. Usually these features are extracted through artificially learned models. The research is dependent on the quality and reliability of the extracted features. In future, the proposed technique can be combined with various feature extraction models to test its usability in a real-time scenario.

References

Abooraig, R., Al-Zu'bi, S., Kanan, T., Hawashin, B., Al Ayoub, M. and Hmeidi, I. (2018), "Automatic categorization of Arabic articles based on their political orientation", *Digital Investigation*, Vol. 25, pp. 24-41.

Adewole, K.S., Akintola, A.G., Salihu, S.A., Faruk, N. and Jimoh, R.G. (2019), "Hybrid rule-based model for phishing URLs detection", *International Conference for Emerging Technologies in Computing*, *Springer*, pp. 119-135.

Allen, J., Gomez, L., Green, M., Ricciardi, P., Sanabria, C. and Kim, S. (2012), " Social network security issues: Social engineering and phishing attacks", *Proceedings of Student-Faculty Research Day, CSIS*, *Pace University*.

Alzu'bi, S., Islam, N. and Abbod, M. (2011), "Enhanced hidden Markov models for accelerating medical volumes segmentation", *IEEE GCC Conference and Exhibition (GCC '11)*, *IEEE*, pp. 287-290.

Alzu'bi, S., Hawashin, B., Eibes, M. and Al-Ayyoub, M. (2018), "A novel recommender system based on apriori algorithm for requirements engineering", *Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS '18), IEEE*, pp. 323-327.

Alzu'bi, S., Hawashin, B., Mujahed, M., Jararweh, Y. and Gupta, B.B. (2019), "An efficient employment of internet of multimedia things in smart and future agriculture", *Multimedia Tools and Applications*, pp. 1-25.

Anti-Phishing Working Group (APWG (2019), "Phishing activity trends report, 1st quarter 2019", available at: https://apwg.org/trendsreports/ (accessed 15 May 2019).

Babagoli, M., Aghababa, M.P. and Solouk, V. (2019), "Heuristic nonlinear regression strategy for detecting phishing websites", *Soft Computing*, Vol. 23 No. 12, pp. 4315-4327.

Bhattacharjee, S.D., Talukder, A., Al-Shaer, E. and Doshi, P. (2017), "Prioritized active learning for malicious URL detection using weighted text-based features", *IEEE International Conference on Intelligence and Security Informatics (ISI '17)*, *IEEE*, pp. 107-112.

Blum, A., Wardman, B., Solorio, T. and Warner, G. (2010), "Lexical feature based phishing URL detection using online learning", *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, ACM*, pp. 54-60.

Caputo, D.D., Pfleeger, S.L., Freeman, J.D. and Johnson, M.E. (2014), "Going spear phishing: exploring embedded training and awareness", *IEEE Security and Privacy*, Vol. 12, pp. 28-38.

Chiew, K.L., Tan, C.L., Wong, K., Yong, K.S. and Tiong, W.K. (2019), "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system", *Information Sciences*, Vol. 484, pp. 153-166.

Fette, I., Sadeh, N. and Tomasic, A. (2007), "Learning to detect phishing emails", *Proceedings of the 16th International Conference on World Wide Web*, *ACM*, pp. 649-656.

Gunawardena, S.-H., Kulkarni, D. and Gnanasekaraiyer, B. (2013), "A steganography-based framework to prevent active attacks during user authentication", *8th International Conference on Computer Science and Education (ICCSE '13)*, *IEEE*, pp. 383-388.

Gupta, S., Singhal, A. and Kapoor, A. (2016), "A literature survey on social engineering attacks: phishing attack", *International Conference on Computing, Communication and Automation (ICCCA '16)*, *IEEE*, pp. 537-540.

Hawashin, B., Alzu'bi, S., Kanan, T. and Mansour, A. (2019), "An efficient semantic recommender method for Arabic text", *The Electronic Library*, Vol. 37 No. 2, pp. 263-280.

Hawashin, B., Mansour, A. and Aljawarneh, S. (2013), "An efficient feature selection method for Arabic text classification", *International Journal of Computer Applications*, Vol. 83 No. 17.

Ibrahim, D.R. and Hadi, A.H. (2017), "Phishing websites prediction using classification techniques", *International Conference on New Trends in Computing Sciences (ICTCS '17)*, *IEEE*, pp. 133-137.

Islam, R. and Abawajy, J. (2013), "A multi-tier phishing detection and filtering approach", *Journal of Network and Computer Applications*, Vol. 36 No. 1, pp. 324-335.

Jain, A.K. and Gupta, B. (2019), "A machine learning based approach for phishing detection using hyperlinks information", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10 No. 5, pp. 2015-2028.

Jakobsson, M. (2018), "Two-factor in authentication: the rise in SMS phishing attacks", *Computer Fraud and Security*, Vol. 2018 No. 6, pp. 6-8.

Khonji, M., Jones, A. and Iraqi, Y. (2011), "A novel phishing classification based on URL features", *GCC Conference and Exhibition (GCC '11)*, *IEEE*, pp. 221-224.

Kumar, N. and Chaudhary, P. (2017), "Mobile phishing detection using naive Bayesian algorithm", *International Journal of Computer Science and Network Security*, Vol. 17 No. 7, pp. 142-147.

Li, Y., Yang, Z., Chen, X., Yuan, H. and Liu, W. (2019), "A stacking model using URL and HTML features for phishing webpage detection", *Future Generation Computer Systems*, Vol. 94, pp. 27-39.

Mamun, M.S.I., Rathore, M.A., Lashkari, A.H., Stakhanova, N. and Ghorbani, A.A. (2016), "Detecting malicious URLs using lexical analysis", *International Conference on Network and System Security*, *Springer*, pp. 467-482.

Moghimi, M. and Varjani, A.Y. (2016), "New rule-based phishing detection method", *Expert Systems with Applications*, Vol. 53, pp. 231-242.

Mujtaba, G., Shuib, L., Raj, R.G., Majeed, N. and Al-Garadi, M.A. (2017), "Email classification research trends: review and open issues", *IEEE Access*, Vol. 5, pp. 9044-9064.

Nagaraj, K., Bhattacharjee, B., Sridhar, A. and Sharvani, G. (2018), "Detection of phishing websites using a novel twofold ensemble model", *Journal of Systems and Information Technology*, Vol. 20 No. 3, pp. 321-357.

Rado, O., Ali, N., Sani, H.M., Idris, A. and Neagu, D. (2019), "Performance analysis of feature selection methods for classification of healthcare datasets", *Proceedings of the Computing Conference on Intelligent Computing, Springer*, pp. 929-938.

Rodríguez, J.E.R., García, V.H.M. and Castillo, N.P. (2019), "Webpages classification with phishing content using naive Bayes algorithm", *International Conference on Knowledge Management in Organizations*, *Springer*, pp. 249-258.

Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019), "Machine learning based phishing detection from URLs", *Expert Systems with Applications*, Vol. 117, pp. 345-357.

Shirsat, S.D. (2018), "Demonstrating different phishing attacks using fuzzy logic", *Second International Conference on Inventive Communication and Computational Technologies (ICICCT '18)*, *IEEE*, pp. 57-61.

Subasi, A., Molah, E., Almkallawi, F. and Chaudhery, T.J. (2017), "Intelligent phishing website detection using random Forest classifier", *International Conference on Electrical and Computing Technologies and Applications (ICECTA '17)*, *IEEE*, pp. 1-5.

Tyagi, I., Shad, J., Sharma, S., Gaur, S. and Kaur, G. (2018), "A novel machine learning approach to detect phishing websites", *5th International Conference on Signal Processing and Integrated Networks (SPIN '18)*, *IEEE*, pp. 425-430.

Vayansky, I. and Kumar, S. (2018), "Phishing: challenges and solutions", *Computer Fraud and Security*, Vol. 2018, pp. 15-20.

Wadhwa, A. and Arora, N. (2017), "A review on cyber crime: major threats and solutions", *International Journal of Advanced Research in Computer Science*, Vol. 8

Welch, I., Gao, X. and Komisarczuk, P. (2011), "Two-stage classification model to detect malicious web pages", *IEEE International Conference on Advanced Information Networking and Applications (AINA '11)*, *IEEE*, pp. 113-120.

Xiang, G., Hong, J., Rose, C.P. and Cranor, L. (2011), "Cantina+: a feature-rich machine learning framework for detecting phishing web sites", *ACM Transactions on Information and System Security*, Vol. 14 No. 2, p. 21.

Zhao, R., John, S., Karas, S., Bussell, C., Roberts, J., Six, D., Gavett, B. and Yue, C. (2016), "The highly insidious extreme phishing attacks", *25th International Conference on Computer Communication and Networks (ICCCN '16)*, *IEEE*, pp. 1-10.

Corresponding author
Hikmat Ullah Khan can be contacted at: hikmat.ullah@ciitwah.edu.pk