Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning

Alyssa Anne Ubing[1], Syukrina Kamilia Binti Jasmi[2], Azween Abdullah[3], NZ Jhanjhi[4], Mahadevan Supramaniam[5]
School of Computing and IT, Taylors University, Subang Jaya, Malaysia[1,2,3,4]
Research & Innovation Management Centre, SEGI University, Malaysia[5]

***Abstract*—This research focuses on evaluating whether a website is legitimate or phishing. Our research contributes to improving the accuracy of phishing website detection. Hence, a feature selection algorithm is employed and integrated with an ensemble learning methodology, which is based on majority voting, and compared with different classification models including Random forest, Logistic Regression, Prediction model etc. Our research demonstrates that current phishing detection technologies have an accuracy rate between 70% and 92.52%. The experimental results prove that the accuracy rate of our proposed model can yield up to 95%, which is higher than the current technologies for phishing website detection. Moreover, the learning models used during the experiment indicate that our proposed model has a promising accuracy rate.**

Keywords—Phishing; feature selection; classification models; random forest; prediction model; logistic regression

I. Introduction

In this technological era, the Internet has made its way to become an inevitable part of our lives. It leads to many convenient experiences in our lives regarding communication, entertainment, education, shopping and so on. As we progress into online life, criminals view the Internet as an opportunity to transfer their physical crimes into a virtual environment. The Internet not only provides convenience in various aspects but also has its downsides, for example, the anonymity that the Internet provides to its users. Presently, many types of crimes have been conducted online. Hence, the main focus of our research is phishing. Phishing is a type of cybercrime [1] where the targets are lured or tricked into giving up sensitive information, such as Social Security Number personal identifiable information and passwords. This obtainment of such information is done fraudulently. Given that phishing is a very broad topic, we have decided that this research should specifically focus on phishing websites.

According to [2], performing a general phishing attack has four steps. First, the phisher creates and set up a fake website that will look exactly like a legitimate website. Second, he or she would send the uniform resource locator (URL) link of the website to their targeted victims by pretending to be a legitimate company or organisation. Third, he or she will attempt to convince the victim to visit the constructed fake website. Fourth, gullible victims will click on the link of the fake website and input the required useful information into it. Finally, by using the personal information of the victim, the phisher will use the information in performing fraud activities. However, phishing attacks [3] are not performed professionally to avoid suspicions from users or victims.

Phishing becomes a threat to many individuals, particularly those who are not aware of the threats in the Internet. Based on a report produced by FBI [4], a minimum damage of $2.3 billion had been caused by phishing scams between the period of October 2013 and February 2016. Commonly, users do not observe the URL of a website. Sometimes, phishing scams engaged through phishing websites can be easily deterred by observing whether a URL belongs to a phishing or legitimate website. In the case where a website is suspected as phish, a user can direct him- or herself out from the virtual environment and away from the criminal’s grasp.

However, current technologies are not fully capable to detect phishing websites, for example, browser security indicators. A survey on ‘Why Phishing Works’ [5] reported that 23% of its respondents relied only on the webpage content to determine its legitimacy. In addition, many users cannot differentiate between a padlock icon in the browser and a padlock icon as a favicon or in page contents. Completely relying on Hypertext Transfer Protocol Secure (HTTPS) [6] is not advisable also because malware can install the public key of a phisher’s certificate authority (CA). This may be used to fool the trusted root CA list of a computer.

Owing to the limitations of existing technologies in detecting a phishing website, expecting the users to observe and have the ability to determine whether a URL is phishing or legitimate would be unrealistic, inefficient and inaccurate. Therefore, in addressing these challenges, an automated approach must be considered for phishing website detection. Currently, [7] one of the problems encountered in such developments is accuracy.

This research paper presents the accuracy improvement with the help of an employed feature selection algorithm, as well as a prediction model by using ensemble learning where majority of the results influence the final prediction. The conclusion will discuss the major results of all the models used in the ensemble. We have also documented the accuracy comparison among individual learning models that were tested through the Azure Machine Learning Studio for benchmarking purposes.

II. Related Work

Recently, proposals on many anti-phishing techniques are presented to reduce phishing attacks through prevention and detection. These studies focus on the structure and components of a URL, feature selection method, ensemble learning and existing phishing detection technologies.

A. *Structure and Component of a URL*

A URL [8] is commonly known as the website address. It is composed of many different parts [9], as illustrated in Fig. 1.

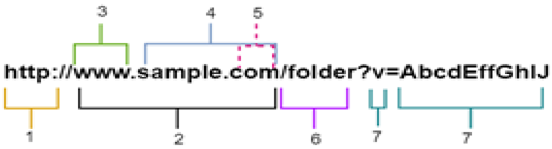

Fig. 1. Structure and Components of a URL.

In the figure, the area labelled with ‘1’ is the Hypertext Transfer Protocol (HTTP). The HTTP represents the protocol used to fetch resources and contents that are requested. The area labelled with ‘2’ is the hostname. The hostname can be further divided into three parts, namely, subdomain (labelled with ‘3’), domain (labelled with ‘4’) and top-level domain (labelled with ‘5’) which is also known as the web address suffix. The area labelled with ‘6’ shows the path that can be typically referred to as a directory on the webserver. Finally, the area labelled with ‘7’ holds the parameter (v) and value (AbcdEffGhlJ). The symbol ‘?’ before the parameter initialises the parameters inside the URL.

B. *Feature Selection*

Feature selection [10] plays a significant role during data analysis. The feature selection method aids in improving the accuracy of the prediction model in such that it reduces the number of features to only those that are critical in influencing the prediction. Specifically, this method helps in cleaning the initial dataset features by retaining only relevant and useful features. Thus, the feature selection [11] algorithm will disregard the features that do not have a high rank in feature importance. However, information loss has no critical effect if the data underwent the feature selection.

C. *Ensemble Learning*

The concept of ensemble learning is an ensemble of algorithms that use more than one learning models. The models [12] used to create an ensemble has its predictions combined to obtain the final prediction.

Ensemble [13] methods are useful and have three primary advantages. The application of this method can be used for a statistical reason, which is relevant to the lack of sufficient data used to represent the data distribution. Owing to the lack of such data, the hypotheses that provide a similar training accuracy can be used as one of the learning algorithms for the ensemble. Thus, these methods can help in risk reduction when a wrong model is selected by aggregating the available candidate models. In addition, the ensemble method can be used for computational purposes. Moreover, many learning algorithms, such as decision tree or neural network (NN) that work by executing a local search, are available. These methods will provide optimal solutions from a local perspective. The ensemble method can showcase its advantage in such scenarios because it can run multiple local searches in a parallel manner at different starting points. Finally, it can be used in representation purposes. Although the representation of the actual function cannot be implemented by a single hypothesis, it can be approximated by the combined hypotheses. This concept is similar to signal processing.

Several ensemble methods [14] are currently available worldwide (e.g. bagging, boosting and stacking).

1) Bagging: Bagging is known as one of the earliest ensemble learning algorithms. This algorithm has a superior performance and is also one of the simplest to implement. Bootstrapped copies of training data cause bagging diversity. This method is helpful when the data are insufficient or have limited size. To ensure that sufficient training samples are available, large portions of the samples are placed into each sample subset, allowing individual training subsets to have identical instances. To ensure that data diversity is maintained, an unstable base learner should be used to produce variations of decision boundaries.

2) Boosting: Boosting develops different types of base learners by sequentially reweighting the instances of the training dataset. Each instance that has been wrongly classified by the previous base learner will receive a larger weight in the subsequent training round. Boosting repeatedly applies a base learner to modified versions of a dataset. Each boosting iteration fits the weighted training data to a base learner. The error and weight computation of accurately predicted instance is reduced, whereas those that were wrongly predicted have increased weights.

3) Stacking: Stacking is a high-level base learner that mainly combines lower-level base learners to improve the predictive accuracy. It is tasked to learn a meta-level base learner to combine the predictions of all the base-level base learners. Then, these base learners are generated by applying various types of learning algorithms to a dataset. Stacking collects the output of each base learner into a new dataset. Stacking repeats and the dataset for each instance represents every base learner’s prediction, as well as the correct classification of the dataset. Base learners must be formed from a batch of training data that do not have the instance included within it; this step is similar to cross-validation. The newly created data should be used for a learning problem, whereas a learning algorithm should be applied to address this problem.

D. *Existing Technology for Phishing Detection*

Browser extensions such as Spoofguard and Netcraft, are used to detect phishing websites [15], with an accuracy of up to 85%. Moreover, automatic real-time phishing detectors (e.g. PhishAri) [16] are available. PhishAri has an accuracy of 92.52%. It is an easy-to-use Chrome browser extension and detects phishing through features such as shortened URL. Meanwhile, DeltaPhish [17] can detect phishing webpages in compromised legitimate websites; its accuracy rate remains higher than 70%. According to an experiment [18], these technologies have an accuracy rate of up to 84% by using six anomaly based features.

III. Proposed Model

The proposed solution model (Fig. 2) improves the accuracy by employing a feature selection algorithm. By filtering into 30 features of the initial dataset, the algorithm selects those that are critical in influencing the outcome of the prediction. Therefore, by having a few features, irrelevant features do not influence the accuracy of the model and its prediction. Furthermore, the prediction model is trained through ensemble learning where multiple learning models are used. By using multiple models when conducting predictions, the outcomes are not bias to only one model. Hence, we demonstrate that the results from all the models are used and counted to determine the majority of votes. For example, if the majority of the models indicate that a website is phishing, then, the final prediction of the ensemble shows that the website is indeed phishing.

A. *Phishing Website Dataset (30 Features)*

We have retrieved a set of phishing website datasets from the UCI Machine Learning Repository. The dataset used has 30 features with result column. The features include ID, having_IP_Address,URL_Length,Shortening_Service,having_At_Symbol,double_slash_redirecting,Prefix_Suffix,having_Sub_Domain,SSLfinal_state,Domain_registration_length,Favicon,port,HTTPS_token,Request_URL,URL_of_Anchor,Links_in_tags, SFH, Submitting_to_email, Abnormal_URL, Redirect, on_mouseover, Right Click, pop Up Window, iFrame, age_of_domain,DNSRecord,web_traffic,Page_Rank,Google_Index, Links_pointing_to_page and Statistical_report.

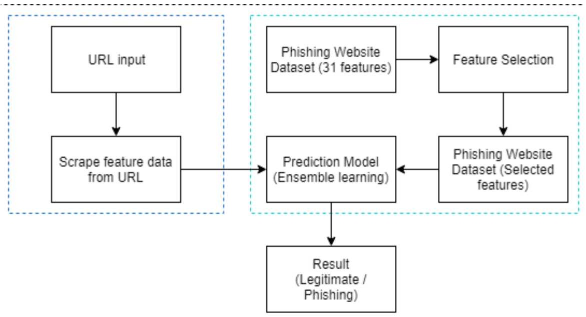
URL input
Scrape feature data from URL
Phishing Website Dataset (31 features)
Feature Selection
Prediction Model (Ensemble learning)
Phishing Website Dataset (Selected features)
Result (Legitimate / Phishing)

Fig. 2. Flowchart of the Proposed Model.

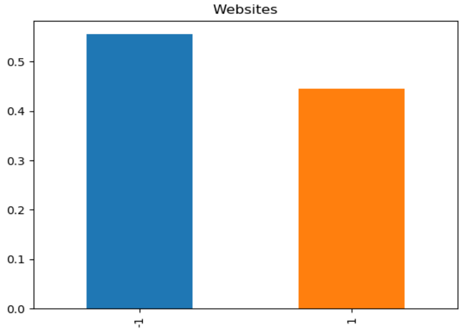

Fig. 3. Bar Graph of the Dataset used for Website Phishing. the Dataset Contains 55% Phishing and 45% Legitimate Websites.

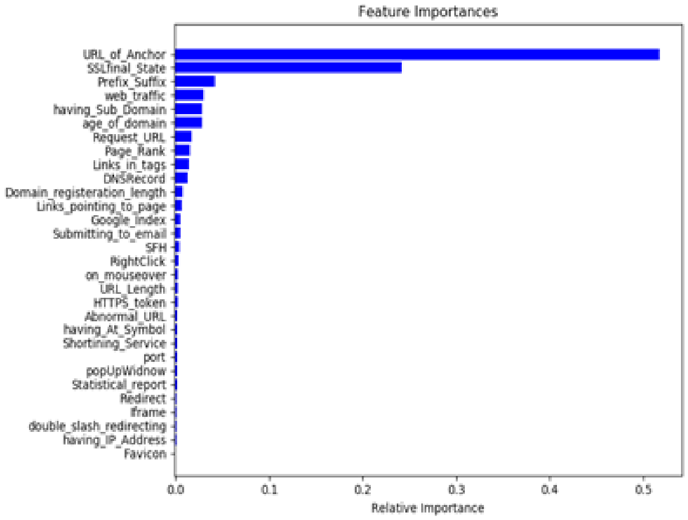

Fig. 4. Dataset Features Ranked based on Feature Importance.

However, not all of these features would be critical in influencing the prediction whether a website is legitimate or phishing. Therefore, to improve detection accuracy and efficiency, the initial dataset is passed through the feature selection model. Figure 3 shows the statistical representation of the dataset classification (1–legitimate; negative 1–phishing).

B. *Feature Selection*

Feature selection model processes the initial dataset and obtains the array value of the selected features. Before conducting the feature selection, we must first drop the result as well as the ID column because these data should not be included. The used feature selection algorithm is based on random forest regressor (RFG). The RFG has a built-in feature selection library that can identify the specified amount of critical features that are necessary according to feature importance. Figure 4 illustrates the features based on its relative importance. In this research, we have utilised nine features based on the feature importance algorithm where the model returns the nine features in the form of array values based on the Comma-separated values (CSV) that it had read.

C. *Prediction Model (Ensemble Learning)*

The prediction model will read the newly created CSV file that only holds the result data and the selected features that have been identified using the feature selection algorithm. We set the SEED of 8888 where the test and train sizes of our model are 0.2 and 0.8, respectively. The concept of ensemble learning is when two or more models are used to achieve the final prediction of data. In this project, we have combined a number of models, namely, Gaussian naive Bayes, support vector machine, K-nearest neighbour, logistic regression, multilayer perceptron NN, gradient boosting and random forest classifiers. Finally, each of these models is individually scored based on their predictions. The predictions made will be compared with the test data. Thereafter, all the predictions from each model are listed. Thus, each model has its own list of results. The list will be compared and is then compared with the test data list to obtain the accuracy score of the combined models against the accurate result. Prediction is conducted in a manner that majority of the model's prediction is employed. For example, if five out of seven models predicts that the website is legitimate, then, the result will show that the website is indeed legitimate. Here, the majority of votes apply, and an accuracy rate of up to 95.5% can be achieved. This rate is relatively high when compared with the results gathered from the experiment performed in the Microsoft Azure Machine Learning Studio.

D. URL Input

Herein, we will use the URL as input to identify if a website is legitimate or phishing. Then, the URL that has been inputted will go through our code, and it will return a CSV file that contains the scraped feature data for the specified URL. Additional details of the scraping feature data will be discussed in the following section.

E. Scrape Feature Data from a URL

As mentioned in the feature selection section, nine features are classified as critical ones, which are the main features that will be used to identify if a website is legitimate or phishing. These selected nine features are URL_of_Anchor, SSLfinal_State,Prefix_Suffix,Web_traffic,having_Sub_Domain,age_of_domain,Request_URL,Page_RankandLinks_in_tags. Therefore, we have programmed our system to the scrape feature data based on these features. In the development stage, our system is programmed according for each feature requirement and then it will return the result of 1 or −1 or 0. After all the nine features have been scraped, the result will be generated into a CSV file. Subsequently, the new CSV file will be fed to the prediction model to evaluate whether a website is legitimate or phishing.

F. Result

The model will predict whether a row of data is legitimate or phishing. After performing the prediction, the results will be printed accordingly.

IV. Experimental Setup

In this section, we will provide a summary of the performed experiment for this research. We start with a set of 177 features of which 38 are content-based and the rest are URL-based. Content-based features are mostly derived from the technical (HTML) contents of webpages e.g., counting external and internal links. Counting IFRAME tags, and checking whether IFRAME tag's source URLs are present in blacklists and search engines, checking for password field and testing how the form data is transmitted to the servers (whether Transport Layer Security is used and whether "GET" or "POST" method is used to transmit form data with password field), etc. URL -based features include lexical properties of URLs such as counting number of ".", "–", "_", etc. in various parts of URLs, checking whether IP address is used and what type of notation is used to represent the IP address in place of a domain name . This experiment was set up to evaluate the accuracy of individual learning model when it is fed into the Phishing Website Dataset prior to feature selection.

A. Accuracy Comparison among Individual Learning Models

In this experiment, we completely relied on the Microsoft Azure Machine Learning Studio, a tool that supports collaboration and allows drag and drop, which can be used for testing. In addition, this tool can be used to establish and deploy predictive analytics solution.

The used phishing dataset contains 30 features and 5126 records. The split data module's property for the fraction of row was maintained at 0.8, whereas the random seed was set to 8888. These properties were statically set during the entire experiment.

The experiment was performed in accordance with the guideline from Microsoft. First, we dragged and dropped the modules into our experimental platform. Second, we connected the modules. This will be our runnable experiment in the Machine Learning Studio. After the structure had been set up, the experiment was saved. Third, the phishing dataset was uploaded onto the platform and was then dragged and dropped into the experimental platform. Finally, the dataset was connected to a module called 'split data'. The split data module was used to divide the dataset into two different sets used for training and testing. Thereafter, we searched and chose the classifiers that we will use for accuracy comparison. The chosen classifier and split data module must be connected to the train model module. The train model module allowed for the training to occur. For the purpose of this research, the training model was set to a classification model to determine whether a website is legitimate or phishing. In this case, the results that are expected to be returned are 1 or −1. To ensure that the model knows what it must predict, we set the 'selected column' on the train model module to the dataset column that we want it to predict. In this experiment, we have fixed the column as 'Result'. Once training is completed, the subsequent module to be added is the 'Score Model'. By using a trained classification model, the score model should generate predictions based on the given data. Finally, the 'Evaluate Model' module was used to determine the accuracy of each of the trained model. The metric result is dependent on the classifier models that were used during the experiment. This module also produces graphs that show the accuracy of each classifiers used during the experiment. Tables 1 and 2 document the results of the different classifiers used. The gathered result also includes total number of true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), precision, recall, accuracy and F1 score.

The 2 × 2 confusion matrix table (Table 1) lists the rate of TP, TN, FP and FN. A confusion matrix is a type of contingency table, which is also known as error matrix. The table can be constructed if both the predicted and true values for a sample set are known. The TP rate indicates the proportion of correct predictions, also called as recall. An FP rate shows the proportion of negative cases that have been predicted as positive. A TN rate represents the proportion of negative cases that have been correctly predicted, whereas the FN rate shows the proportion of positive cases that have been wrongfully predicted as negative.

Recall is simply defined as the percentage measurement of the actual phishing websites that have been correctly classified. The percentage of cases that have been correctly classified is known as precision. F1 score is the weighted average of precision and recall. As presented in Table 2, recall, precision, recall and F1 score are documented for each of the models. The accuracy rate of each model has also been obtained, as shown in Fig. 5.

TABLE I. CONFUSION MATRIX COMPARISON BETWEEN MODELS

Classification	True Positive	False Negative	False Positive	True Negative
Two-Class Averaged Perceptron	437	36	40	512
Two-Class Bayes Point Machine	438	35	42	510
Two-Class Boosted Decision Tree	452	21	7	545
Two-Class Decision Forest	449	24	8	544
Two-Class Decision Jungle	452	22	28	524
Two-Class Locally Deep Support Vector	451	22	13	539
Two-Class Logistic Regression	437	36	36	516
Two-Class Neural Network	451	22	17	535
Two-Class Support Vector Machine	432	41	41	511

TABLE II. CONFUSION METRIC COMPARISON AMONG LEARNING MODELS

Classification	Accuracy	Precision	Recall	F1 Score
Two-Class Averaged Perceptron	0.926	0.916	0.924	0.920
Two-Class Bayes Point Machine	0.925	0.912	0.926	0.919
Two-Class Boosted Decision Tree	0.973	0.985	0.956	0.970
Two-Class Decision Forest	0.969	0.982	0.949	0.966
Two-Class Decision Jungle	0.952	0.942	0.956	0.949
Two-Class Locally Deep Support Vector	0.966	0.972	0.953	0.963
Two-Class Logistic Regression	0.930	0.924	0.924	0.924
Two-Class Neural Network	0.962	0.964	0.953	0.959
Two-Class Support Vector Machine	0.920	0.913	0.913	0.913

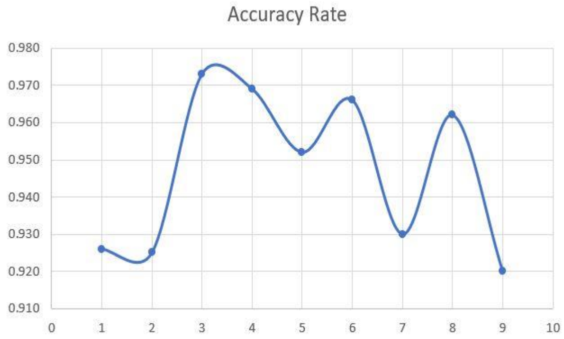

Fig. 5. Accuracy Rate of Learning Models based on Confusion Metrics Obtained from the Microsoft Azure Machine Learning Studio.

TABLE III. CONFUSION MATRIX FOR THE PROPOSED MODEL

Classification	True Positive	False Negative	False Positive	True Negative
Ensemble Learning	560	18	29	419

TABLE IV. CONFUSION MATRIX FOR THE PROPOSED MODEL

Classification	Accuracy	Precision	Recall	F1 Score
Ensemble Learning	0.954	0.935	0.959	0.947

B. *Accuracy Rate based on the Proposed Model*

In this experiment, we use the same phishing datasets that have been used in Experiment A. However, the datasets only contain nine critical features that have been chosen using our feature selection algorithm. Moreover, the number of records (5126) remains, as well as the random seed, was set to 8888. These properties were statically set during the entire experiment.

Our proposed model was coded in Python language, and the compiler we used to perform this experiment is called PyCharm. The 'Scikit-learn' library of Python language does support a confusion matrix. Therefore, we utilise the library to obtain our confusion matrix for the proposed model.

First, the CSV file that contains nine feature datasets was imported into code. Second, the data was divided into train and test datasets accordingly. Finally, the data passed through our proposed model (i.e. ensemble learning model) to train our system. After completing the training, the predicted result was obtained. Therefore, we have both the actual and predicted results. Thus, by utilising the 'Scikit-learn' library and feeding in our actual and predicted results, we can obtain the confusion matrix of our proposed model, as shown in Tables 3 and 4.

The confusion matrix shown in Tables 3 and 4 has the same format as Tables 1 and 2. The overall confusion matrix tables include the rate of TP, TN, FP, FN, accuracy, precision, recall and F1 score. The result of Experiments A and B are discussed in the Findings section.

C. *Findings*

On the basis of the experimental results regarding the readings gathered from the confusion matrix of both the ensemble learning and individually tested learning models from the Microsoft Azure Machine Learning Studio, we conclude that the performance of our proposed model is better than the performance of most of the individual learning model. The proposed model does not perform better than other ensemble learning libraries, such as decision tree, boosted decision tree, locally deep support vector and NN. However, it is better than the decision jungle that can be found in the ensemble learning library. The one possible reason in which the decision tree-like models exceeded the proposed model is overfitting. Overfitting occurs when there is high train accuracy but low validation or test accuracy. The pattern that is being trained by the model may be distorted owing to the noise being fed into the training data. Given that noise is stochastic, the training data fitted with noise reduces training error. However, it does not help in reducing the validation or test error, resulting in validation and test error increase. The attributes or features that are irrelevant to the prediction can result in overfitting the training data. Because the results of the Microsoft Azure Machine Learning Studio are based on individual learning models that have been fed into the dataset that have not underwent feature selection, irrelevant features may have contributed to the noise, causing such models to produce an overfitting result.

V. Conclusion

Certain classifiers that are more prone to overfitting than others are present, thus yielding higher accuracy rate if they are based on the dataset that they have been trained upon. This result can be observed in the experiment performed through Azure, specifically trees. To address the overfitting problem while focusing on increasing the prediction accuracy, the proposed solution model uses feature selection and ensemble learning where multiple learning models are combined to produce a prediction. By using multiple models, the prediction is not bias towards one model and is instead based on majority of predictions such that all predictions from each model influences the final ensemble prediction.

VI. Future Work

The authors believe that the phishing attacks are increasing day by day based on the literature review, though ample solutions are available. However, it is a bit challenge to educate\trained the users besides of detecting phishing attacks.

References

[1] "Phishing | What Is Phishing?" Phishing.org, 2018. [Online]. Available: http://www.phishing.org/what-is-phishing.[Accessed: 15-Oct-2018]

[2] A. Khan and R. Sharma, “A Survey Paper on Detection of Phishing Website by URL Technique,” vol. 6, pp. 33–37, 2018.

[3] R. Sakunthala and S. Shankar, “EAI Endorsed Transactions Review of Various Methods for Phishing Detection,” vol. 5, no. 20, pp. 3–11, 2018.

[4] J. McCabe, “FBI Warns of Dramatic Increase in Business E-Mail Scams”.

[5] R. Dhamija, J. D. Tygar, and M. Hearst, “Why phishing works,” Proc. SIGCHI Conf. Hum. Factors Comput. Syst. - CHI ’06, no. April, pp. 581, 2006.

[6] J. Shi and S. Saleem, “1 Introduction,” pp. 1–14, 1995.

[7] G. Jourdan, G. V Bochmann, R. Couturier, and I. Onut, “Tracking Phishing Attacks Over Time,” pp. 667–676.

[8] "Anatomy of a URL", Web Design Links, 2018. [Online].Available:https://doepud.co.uk/blog/anatomy-of-a-url. [Accessed: 15- Oct- 2018].

[9] Sistrix, “What is the difference between a URL, Domain, Subdomain, Hostname etc.?”.

[10] P. Sharma, “The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes)”.

[11] L. Rokach, “Ensemble-based classifiers,” Artif. Intell. Rev., vol. 33, no. 1–2, pp. 1–39, 2010.

[12] R. Polikar, “Ensemble learning,” Ensemble Mach. Learn. Methods Appl., pp. 1–34, 2012.

[13] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, “Ensemble deep learning for regression and time series forecasting,” IEEE SSCI 2014 - 2014 IEEE Symp. Ser. Comput. Intell. - CIEL 2014 2014 IEEE Symp. Comput. Intell. Ensemble Learn. Proc., no. July 2015, 2014.

[14] G. Wang, J. Hao, J. Ma, and H. Jiang, “A comparative assessment of ensemble learning for credit scoring,” Expert Syst. Appl., vol. 38, no. 1, pp. 223–230, 2011.

[15] D. M. Krishnan and V. Subramaniyaswamy, “Phishing website detection system based on enhanced itree classifier,” ARPN J. Eng. Appl. Sci., vol. 10, no. 14, pp. 5688–5699, 2015.

[16] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, “PhishAri: Automatic realtime phishing detection on twitter,” eCrime Res. Summit, eCrime, no. January 2012.

[17] F. R. Igino Corona, Battista Biggio, Matteo Contini, Luca Piras, Roberto Corda, Mauro Mereu, Guido Mureddu, Davide Ariu, “DeltaPhish: Detecting Phishing Webpages in Compromised Websites.” [Online]. Available: https://arxiv.org/abs/1707.00317.

[18] R. M.Mohammad, F. Thabtah, and L. McCluskey, “An Assessment of Features Related to Phishing Websites using an Automated Technique.” pp. 492–497, 2012.