THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 16 Article 5

December 2021

Performance Assessment of some Phishing predictive models based on Minimal Feature corpus

Orunsolu Abdul Abiodun
Moshood Abiola Polytechnic, Abeokuta, orunsolu.abdul@mapoly.edu.ng

Sodiya A.S
Federal University of Agriculture, Abeokuta

Kareem S.O
Moshood Abiola Polytechnic, Abeokuta

Oladimeji G. B Mr.
Moshood Abiola Polytechnic, Abeokuta

I. INTRODUCTION

Phishing attacks are criminal attempts that fraudulently deceived unsuspecting online users through fake websites into divulging their sensitive personal credentials. These credentials are then used by the con artists to commit identity theft on behalf of the victims. These attacks often led to severe damages ranging from online brand damages to significant financial losses (Abdelhamid et al., 2014; Qabajeh et al. 2018; Mao et al. 2019). For instance, Stats and Trends 2017 in their security reports revealed that about $5billion were lost to phishing attacks involving more than 24,000 victims worldwide. Besides, most ransomware-based attacks are perpetuated through phishing emails (CSO Online report 2016). In a similar vein, Action Fraud Security estimated that about 2 million pounds have already been reported lost to coronavirus-related fraud in the UK as cyber attackers capitalize on the advantages of the current pandemic. For instance, as the Zoom app witnessed a huge rise in the number of users on its platform, cybercriminals immediately used passwords from previous data breaches to perpetuate what is called *zoo bombing* (Action Fraud Report, 2020). On the global level during the pandemic, cyber attackers take advantage of individuals' hunger for safe news, information and solutions to coronavirus to send phishing emails to people to lure them to reveal their sensitive information. Figure 1 shows a fake email purportedly emanating from the World Health Organization (LOC Security report, 2020). This kind of email is often used by phishers to circulate bogus coronavirus tracking sites, maps etc. which are then employed to install ransomware and malicious software.

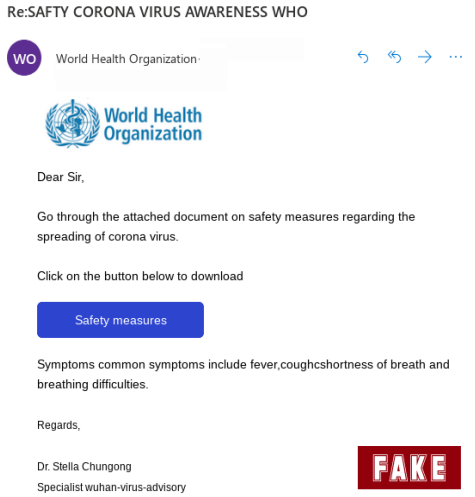

Figure 1: Phishing email purportedly from WHO.

A phishing attack involves setting up a counterfeit website that perfectly mimics the appearance of a known legitimate website. The online users are then deceitfully prompt to access the fake website through email message or links claiming important info or update from the legitimate sites. In this process, most online users get their sensitive credentials harvested by cybercriminals. The credentials harvested normally include bank account numbers, passwords or PINs, credit card numbers, security questions, security codes etc. With the harvested credentials, the

attackers can log in to the genuine websites to steal the victim's money or launch other related attacks. In most instances, vulnerability to phishing threat is due to the ease with which unsuspecting online users navigate web pages using links or URL within a body of an online message (Han et al. 2012). Moreover, there is an increased motivation for phishers as the number of mobile-connected devices accessing social media sites continues to grow. Phisher now embeds malicious links or abnormal URL shortner into e-chat (Aggarwal et al. 2012; Kumar and Kumar, 2014; Orunsolu et al. 2018).

Due to the numerous threats posed by phishing attacks, the online security community and industry have come up with several solutions called anti-phishing systems (Kumar and Kumar 2014). One of the promising anti-phishing countermeasures is the adoption of the machine learning approach in mitigating the severity of phishing attacks (Hamid and Abawajy 2014; Tan et al. 2017). Numerous anti-phishing predictive models have been developed to combat phishing attacks. These predictive models have shown significant performance results in terms of high accuracy, low false positive and false negatives and zero-day detection capability (Sonowal et al. 2017; Adebowale et al. 2018; Mao et al. 2019; Orunsolu et al. 2019). However, the performance of these predictive models is heavily dependent on the types of machine learning algorithm adopted and the type/size of heuristics in the feature set corpus (Qabajeh et al. 2018). These two factors affect the responsiveness and response time of anti-phishing solutions which can limit their application in real-life scenarios (Silva et al. 2020). The limitation is often connected with superfluous training/testing time which may result in high memory overheads, delay in detection time, expensive maintenance/update etc. Thus, responsiveness is used to measure prediction accuracy with commensurate processing time while the response time is used to ensure that the detection time for any window of vulnerability is reasonable and insignificant (Silva et al. 2020). To achieve these, it is imperative to choose an appropriate machine learning algorithm with a minimal dimensional representative feature set (Sonowal et al. 2017; Orunsolu et al. 2019).

In this work, we proposed an approach to examining the different state of art predictive model using reduced phishing feature corpus to resolve the uncertainties that result from performance issues (responsiveness) and other inconsistencies (response time, computational overhead etc.) in the feature set corpus. The primary element of this approach is the composition of the feature set. It considers various factors that have been examined in the literature for the most representative features set (Varshney et al 2016; Fadheel et al. 2017). Specifically, this approach leverage the feature frequency analysis technique for selecting the resultant feature set (Orunsolu et al. 2019). This method

provides the advantage of using features that are regularly more exploited in phishing attacks while reducing the redundant features i.e. low relevance features. For instance, the URL-based features are found to be more regularly exploited than other features in most phishing attacks (Silva et al. 2020; Zouina and Outtaj (2017); Varshney et al. 2016). Besides, our choice of ML algorithms included in the performance measurement is informed by their existing results in extant literature (Basnet et al. 2007; Fadheel et al. 2017; Chin et al. 2018; Orunsolu et al 2019). The contribution of this paper is to improve the deployment of predictive models through slight tuning of feature set with significant performance accuracy. The paper also presents the advantage of improving the discoverability of choice of feature set corpus.

The rest of the paper is organized as follows: Section II presents a literature review on anti-phishing solutions based on non-machine learning approaches and classification algorithms. The reduced feature set algorithm is examined and presented in Section III using some features. In Section IV, the application and results of the different predictive model on the proposed feature set are presented. Section V contains some relevant discussions to our findings in the light of other anti-phishing studies while Conclusions and future works are presented in Section VI.

II. LITERATURE REVIEW

Phishing scams are on the rise today as cyber attackers easily find loopholes to fit into the current situation to launch attacks. For instance, during tax breaks, phishers can design bogus websites asking individuals to file their tax claims. The earliest phishing attacks dated back to more than two decades ago. These attacks started with the bogus emails targeting AOL users and since then, the attacks have transpired to other services on the Internet using sophisticated methods to deceive even experienced online users (Mohammed et al. 2017; Dhamija et al. 2006; Orunsolu et al. 2018). As the phishers continue to circumvent some existing countermeasures, the motivations for online services become threatened. Face with this severe situation, the security communities, software vendors and research institutions responded with several approaches called anti-phishing techniques. For this study, these techniques are classified as (1) non-machine learning approach and (2) machine learning approach.

A. Non- Machine Learning approaches

These approaches are designed to mitigate phishing without the application of classification algorithms. These approaches often include user security training, list-based methods, game-approach, use-case scenarios etc. For instance, Orunsolu et al. 2018 investigated a use-case study that

revealed the socio-demographic perception which influences the users' understanding of security tips information. The study indicated that gender, academic qualification and user's computer knowledge significantly influenced the ability to recognize phishing messages. The study does not consider spear email and phishing websites/logo-based phishing attacks which may limit the generalization of the research study. Similarly, Mohammed et al. 2015 showed that about 53% of individuals were still vulnerable to phishing attacks even after being primed with security tips. However, the study does not provide information about factors that still allow such susceptibility in the altitude of individuals within the study population. In a more recent approach, Silva et al. 2020 proposed a user study that evaluates a set of 12 static features observed in the current phishing attacks. The approach found that some features are more regularly found in a phishing attack with the possibility of greater exploitation from phisher thereby indicating the need for further examination of such features. However, the study does not consider all categories of phishing attacks such as search-engine based, logo-based phishing etc.

In another development, Oest et al. 2020 proposed a framework to improve the performance of the blacklist approach in continuously identifying phishing websites. The approach showed a remarkable performance in proactively protecting users from modern phishing attacks. However, maintaining a blacklist may be a difficult issue due to the everyday explosion in the numbers of newer URLs on the internet. Similarly, Orunsolu et al. 2020 investigated a lightweight approach called PhishCalcluator. This approach used URL legitimacy with a weighting factor to detect phishing. The performance of the approach provides remarkable results in the fight against phishing attacks. However, the use of a small dataset in the evaluation process limit the application of the approach in a critical online scenario Prakash et al. 2010 investigated one of the earliest studies on the blacklist approach. The authors proactively designed a matching framework for new phishing URLs using variations from the original ones. However, the approach provides for superfluous computations of child URLs which may not apply to real-phishing attacks. Jain and Gupta 2016 proposed an auto-updated whitelist approach to prevent client-side phishing attacks. The approach use URL and DNS information for mitigating phishing attacks. The approach achieved an accuracy rate of 86.02%. Varshney et al. 2016 proposed a search-engine strategy called a phishing detector to mitigate phishing attack using domain name and title. The approach achieved an accuracy rate of 99.5%. Generally, these approaches have advantages of simplicity, low computational requirement, efficient resource

management and high adoption e.g. Blacklist on Safe Google browsing. However, these approaches suffer from the poor generalization of new phishing attacks, high false alarms, lower accuracy in certain instances, low real-time protection mechanisms (Qabejah et al. 2018; Adebowale et al. 2018)

B. Machine Learning approaches

Machine learning-based anti-phishing solutions are countermeasures that are enhanced through classification algorithms to detect or predict phishing activities using certain features usually called *fingerprints.* This class of anti-phishing solution remains popular because of its advantages of minimizing false positives and the ability to generalize phishing detection using known instances. This is possible as the ML algorithm can produce a powerful predictive model once the initial feature sets have been chosen.

Several works have reported several classification algorithms to demonstrate the effectiveness of this approach. For example, Han et al. (2012) investigated a whitelist approach using the Naïve Bayes algorithm to capture login information to predict the status of a loading page. The scheme produced a significant phishing detection model. However, their technique is susceptible to new login problem and pharming attacks. In other related works, Orunsolu et al. (2019) proposed a predictive model for phishing detection using frequency analysis of existing feature corpus to design a more discriminative feature class. The system used an aggregate of 15-dimensional feature set trained using Naïve Bayes and Support Vector Machine. The system achieved a remarkable performance with 99.96% accuracy with low false positive. In another application of the SVM model, Mao et al. (2019) investigated an anti-phishing system based SVM machine learning approach using the visual analysis method. The scheme considered webpage layouts using property vector extraction, property vector generation and comparison vector generation. The technique produced a significant accuracy of more than 93.0%. Zouina and Outtaj (2017) studied URL features using the SVM model to obtain a lightweight phishing detection system. Their method considered six features extracted from the domain address of a querying page. Using the evaluation dataset from PhishTank and Alexa, the system produced an accuracy rate of 95.80%.

Using the ensemble machine learning approach, Hamid et al. (2011) analyzed various machine learning models like Bayesian Net, AdaBoost, Decision Tree and Random Forest. In their evaluation, phishing dataset consisting of two separate partitions are used for training and testing purposes. The results indicated that Random Forest produced the highest accuracy of 93%. Similarly, Hota et al. (2018) investigated an approach where features are removed and replaced from the original feature set randomly until a certain accuracy

threshold is achieved. This method is called the Remove-Replace Feature selection technique (RRFST). The approach achieved an accuracy of 99.27% with an ensemble of C4.5 and CART. In earlier related work, Mohamed et al. 2014 examined the problem of phishing detection using several rule induction algorithms. The authors evaluated their approach with a dataset tested on C4.5, CBA, RIPPER and PRISM. Similarly, Khadi and Shinde (2014) investigated the problem of an email phishing detection system by combining a RIPPER ML algorithm with fuzzy logic on several features from *fingerprints.* The approach produced a prediction rate of 85.4%. Recently, Li et al., 2019 considered a stacking approach with 20 features extracted from the URL and HTML. The extracted features were subjected to training using an ensemble model of Gradient Boosting Decision Tree, XGBoost and LightGBM. The approach which was evaluated using a large dataset achieved a remarkable accuracy of 98.60% accuracy and a 1.54% false alarm rate. In a similar vein, Adebowale et al. 2018 investigated an integrated approach consisting of 35-dimensional features set using an Adaptive Neuro-Fuzzy Inference System. The authors' integrated features consist of text, images and frames selected using Chi-Square Statistics and Information Gain technique. The authors evaluated the scheme with a predictive model consisting of SVM, K-NN and ANFIS. This system achieved 98.3% accuracy.

Chin et al. (2018) presented an approach called PhishLimiter that used deep packet inspection (DPI) and a software-defined networking method to identify phishing activities in email and web-based communication. Their scheme adopted an Artificial Neural Network model with an accuracy of 98%. Similarly, Seymour and Tully (2018) considered a new ML-based on NN called Long Short Term Memory Artificial NN to combat the problem of spear-phishing on online social networks. The model presented word vectors after the training process consisting of different post messages. The approach provided experimental results that indicated that the proposed system was superior to other manual classification approaches. In one of the earlier schemes to NN, Mohammad et al. (2014b) developed a Neural Network-based anti-phishing model that improves the learned predictive model based on the system's previous training experiences. The authors posited the use of a self-structuring Neural Network classification approach to cope with the changing nature of phishing *fingerprints*. The authors considered about thirty features to investigate the accuracy of their model. The evaluation process involved more than 10000 instances with remarkable accuracy.

For this study, the following ML algorithms have been identified to investigate the performance of our minimal feature set due to their high adoption, popularity in phishing

problems, remarkable performance and computational efficiency (Qabajeh et al. 2019; Pham et al. 2014; Orunsolu et al. 2019; Pham et al. 2018).

i. **Naïve Bayes Classifier**: This is a simple prediction and classification algorithm which use the joint probabilities of certain features to estimate the conditional independence assumption of other unknown attributes. This classifier is more practical because it does not require a very large training set and can easily handle missing attribute values. It has been researched in many anti-phishing systems with significant performance accuracy. For instance, Han et al. 2012 used the NB algorithm on login user interface information of whitelisted websites to achieve an efficient anti-phishing system. Besides, Orunsolu et al. 2019 used NB on certain heuristics from the URL, Webpage properties and webpage behaviour to design an efficient anti-phishing predictive model.

ii. **Random Tree:** This is another classifier that has been widely used in phishing detection (Mao et al. 2019; Garera et al. 2007). It consists of an ensemble machine learning method used for classification, regression and other data mining tasks. The approach operates basically by constructing a multitude of decision trees at the training time and produces the output as a class that is the mode of the classes or mean prediction of the individual's trees.

iii. Support Vector Machine: This is one of the most popular classifiers in designing a machine-learning-based phishing detection model (Orunsolu et al. 2019; Hota et al 2018). The SVM model is often generated by obtaining a set of annotated training samples, each as belonging to one or the other of two categories which then assigns new examples to one or another category. The model is therefore referred to as a non-probabilistic binary classifier. For instance, Zouina and Quttaj (2017) examined an SVM predictive model using URL features with remarkable performance results.

iv. Artificial Neural Network: This classification algorithm is often composed of the input layer, one or more hidden layers and the output layer (Kanchan et al. 2017). The input layer is used to compute the weights of the feature instances with the hidden layer assisting in the model/learning construction procedure while the prediction is generated by the output layer. This classification model generates the best possible result without redefining the output criteria.

v. Decision Tree: This is a classification algorithm whose goal is to create a machine learning model that correctly predicts the value of a target sample based on some input samples. Decision Trees consists of basically two main types namely the classification tree and regression tree. In the phishing detection system, the term Classification and Regression Tree (CART) analysis have been used to describe most research in this area. Notable examples of decision tree algorithms include Iterative

Dichotomiser 3, C4.5, Conditional Inference Trees, Chi-square automatic interaction detection etc. For instance, Li et al. 2019 investigated an anti-phishing approach where a Decision Tree was used on features from URL and HTML. The approach indicated the superior performance of this classifier in phishing detection.

III. MINIMAL FEATURE GENERATION ALGORITHM

Features are *fingerprints* that provide recognition for any instances of a class. In phishing problem, features are used to define the legitimacy or otherwise of any website, email or URLs. Although several features have been proposed in the extant literature, the task of generating the most representative feature set remains a big task in any anti-phishing studies. While some works (Zouina et al. 2017; Mao et al. 2019; Hota et al. 2018), considered a single class of feature in their studies, others considered integrated features involving two or more categories (Adebowale et al. 2019; Orunsolu et al. 2019; Li et al. 2019). In either case, efforts are geared toward obtaining a feature set classifier with greater performance accuracy and reasonable resource requirement. It is therefore imperative to continue evaluating the performance of different classifiers on several features in other to keep the anti-phishing model efficient and relevant. Thus, feature generation algorithms are used to create new features using a scientific approach from existing features to construct a predictive model. This is because the generation of relevant features remains central to the performance of data mining and machine learning algorithms. For instance, Gupta et al. 2016 and Toolan et al. 2010 provided the ranking categories for different features used in phishing and spam detection. This ranking provides an insight into low relevance features and high relevance features. The low relevance features are features that are less exploited in phishing attacks. This may due to the cost of implementation from the phishers' side or ease of deployment. On the other hand, high relevance features are features that are more regularly exploited in phishing attacks. These features often call for further investigation as phishers' usually mimic them in a most sophisticated manner to launch new attacks (Silva et al. 2020). Based on this premise, we identified a minimal feature set using the concept of frequency analysis of existing features to investigate the performance of a certain remarkable class of ML algorithms from the extant literature to increase the coverage of anti-phishing solutions. This agrees with Zhu et al. 2020 which claimed that an excessive number of features resulted in over-fitting.

In this study, the phishing dataset includes 13 features extracted from 10,000 instances as captured in a WEKA application. The dataset is obtained from the UCI phishing repository. The dataset is then

normalized and the feature generation algorithm is subsequently invoked (Algorithm 1). Algorithm 1 is adopted with little modification from Orunsolu et al. 2019. The feature set consists of 85% URL-based category and 15% non-URL category. This is due to the popularity of URL-based features in most anti-phishing studies i.e. high relevance features (Sahingoz et al. 2019; Qabajeh et al. 2018; Orunsolu et al. 2019; Adebowale et al. 2018; Silva et al. 2020). The URL feature category remains the most adopted in anti-phishing design because of its simplicity, remarkable accuracy and negligible response time (Zouina and Quttaj (2017); Orunsolu et al. 2019; Toolan and Carthy (2010)). The other features (i.e. non-URL) were chosen randomly without any regard to their underlying contributive significance. The purpose of this is to examine the contributive effect of these features on the URL features. That is, the objective is to determine how different feature category (i.e. high relevance feature vs low relevance feature) can limit the performance of a minimal feature set.

The algorithm consists of an initial large feature set corpus, *DB*, where the frequency analysis assessment method is employed. In some cases, the DB may consist of both a phishing database and a legitimate database. This would provide a better judgement for accessing a particular feature in both databases. For example, preliminary analysis in Orunsolu et al. (2019) indicated that the use of "-" is common to both phishing and legitimate websites. As such, such a feature cannot provide marked differences for predicting a querying URL. The frequency analysis method is based on equation (1). A Frequency Information (FI) is defined based on the principle of exclusivity as a threshold for the selection of any feature (equation 2).

$$FI = f_i / \sum DB \qquad (1)$$

$$0 < FI \ll 1 \qquad (2)$$

The value 0 means no occurrence within the *DB* and the value 1 means the feature is found in all occurrences within *DB*. If the value of a feature exceeds the exclusion limit, the feature is enrolled into the new feature list, *x*. This procedure continues until the entire *DB* is exhausted. The new list, *x*, is then ranked and the highest relevant features are selected. The final minimal feature list, *m*, is constructed according to equation 3. The equation provides the statistical information about the composition of *m* where more than two-third are URL-based and less than one-third is non-URL-based.

$$m = \sum \frac{.85.f_{url}}{x} + \frac{.15 f_{\_url}}{x} \qquad (3)$$

Table 1 presents the meaning of the notations used in the description of algorithm 1 and Table 2 contains the

selected features and their short description.

Table 1. List of notations and their description

Notations	Description
FI	Frequency information
f_i	An instance of a feature
f_{*i}	The feature set of highly relevant features
θ	The exclusion limit for frequency analysis
DB	Database of confirmed phishing fingerprints
n	Number of features in DB
x	New feature list of high relevant features
f_{url}	Instances of URL list in *x*
$f_{\_url}$	Instances of the non-URL list in *x*
m	The final minimal feature list

Algorithm 1: Frequency Feature Assessment Algorithm

Input: Database of feature set corpus, *DB*; predefined exclusion limit value, θ; Frequency Information, *FI*

Output: Minimal Feature set corpus, *m*

Begin

1. For $i = 1\ to\ n\ do\ begin$
2. $\forall\ fi\ \in DB\ do$
3. Calculate $FI\ of\ f_i$
4. $x \leftarrow new\ Feature\ list$
5. { **IF** ($f_i > \theta$) **Then**
6. Insert $f_i\ into\ x$
7. **Else** reject f_i }
8. Next *i*
9. **Continue**
10. Rank $f_i \in x$
11. Select high relevant $f_{*i} \in x$
12. $m \leftarrow select\ top\ f_{*i} \in x\ as\ minimal\ feature\ set$

End

It should be observed that certain features such as keyword extraction, ‘’-“ in the URL path, non-ASCII characters were omitted in our minimal feature set corpus. This is because our investigation revealed that some of these features are related to some features already captured in our feature set. For instance, keyword extraction is related to F5 as it indicates whether prefix or suffix are related to the contents of a page. Also, the “–‘' in the URL path is usually related to F2 as the omitted features are often used in URL elongation.

Table 2. Selected Feature set

S/N	Feature name	Description
1	Number of Dots	This feature elongates a domain name address by adding irrelevant prefix or suffix to genuine URL
2	URL Length	Phishers use a long domain name to disguise fake website
3	@ Symbol	This is used by phishers to redirect to the phishing domain
4	No HTTPS	Most phishing website is hosted on a non-HTTPS domain by phishers due to its non-expensive nature
5	Domain in path	Phishers make use of the domain name in the links to hide the identity of malicious link in the address
6	Https in Hostname	Fraudsters make use of subdomain to let a malicious link look legitimate
7	Path Length	Phishers add the domain mane of a genuine site within the path length of a URL to deceive users
8	IP address	This involves the use of IP address to obscure a server's identity by phishers
9	Popup Window	Phishers used pop-window to circumvent data validation during the authentication process
10	Submitting to Email	This involves phishers using servers that are different from the loading page to obtain users credentials
11	Missing Title	Phishers often host their domain name on a compromised domain whose domain keywords do not relate to its brand.
12	IFrame redirection	Phishers use an Html tag that displays additional pages invisible without a frame border
13	Return URL Length	Phishers use URL that does not return to a particular whois server by obfuscating web address using unrelated information in the URL path

IV. PERFORMANCE ASSESSMENTS AND RESULTS

In this section, the performance assessments of some selected predictive models on minimal phishing heuristics are examined using standard comparison metrics. The dataset used for the evaluation consists of 10000 phishing instances that were imported into a WEKA application. A Java library called JSoup HTML parser was adopted in extracting the feature set from the experimental dataset instances. The library is equipped with API to manipulate data from URL or HTML using DOM, Jquery and CSS techniques. On the other hand, the WEKA application provides an environment where the extracted

features are trained and tested with different classification algorithms. A typical WEKA preprocesses interface for the proposed model indicates the extracted features, size of the evaluation dataset and other defaults settings in the WEKA application. These features can be reverted in WEKA to show the contribution of each of a group of selected features.

The evaluation metrics consists of True Positive (TP) rate, False Positive (FP) rate, Precision, Recall, F1-score and Receivers Operating Curve (ROC). The TP is the rate of correctly predicted phishing instances out of the total phishing instances. On the other hand, the FP is the rate of misclassified phishing instances out of the aggregate phishing instances. The Precision is the ratio of the correctly detected phishing instances to the total number of phishing instances in the evaluation process. The Recall is a measure that determines the number of phishing instances identified correctly as existing phishing instances. F1-score is the measure that determines the harmonic mean of Precision and Recall. The ROC is used to determine the change in FP to the variation in TP. These metrics are very significant in determining the effectiveness of machine learning algorithms. Specifically, the TP and FP evaluate the performance assessment of machine learning classifiers while the remaining metrics assess the efficiency of machine learning classifiers.

The experimental dataset instances were separated into training and testing data using 10-fold cross-validation techniques. Validation techniques often come in different folds based on the settings on the WEKA default interface. This technique ensures the correctness of querying the dataset on some selected features in a testing scenario. Usually, a cross-validation technique is a predictive model that evaluates the performance of a machine learning model on new instances based on a specific portion of the dataset. Thus, the 10-fold cross-validation randomly split the test dataset into ten equal samples where a single stratum then validates the training of the other remaining strata. This process is necessary to generalize the performance of the predictive model to independent data corpus while providing error performance verification for the machine learning model (Orunsolu et al. 2019).

Figure 2 presented the visualization effects (VE) of different features used in the proposed system. The VE clearly has shown that the URL features have more discriminative predictive power than the non-URL features. Specifically, the HTTPS in hostname separated the data instances into two points while the other features produced significantly different colour patterns of the experimental data instances. This function can be extended to construct the confusion matrix and Receivers Operating Curve model of the approach.

A confusion matrix is a table that describes the performance of the classification scheme while ROC estimates the predictive accuracy of the proposed model.

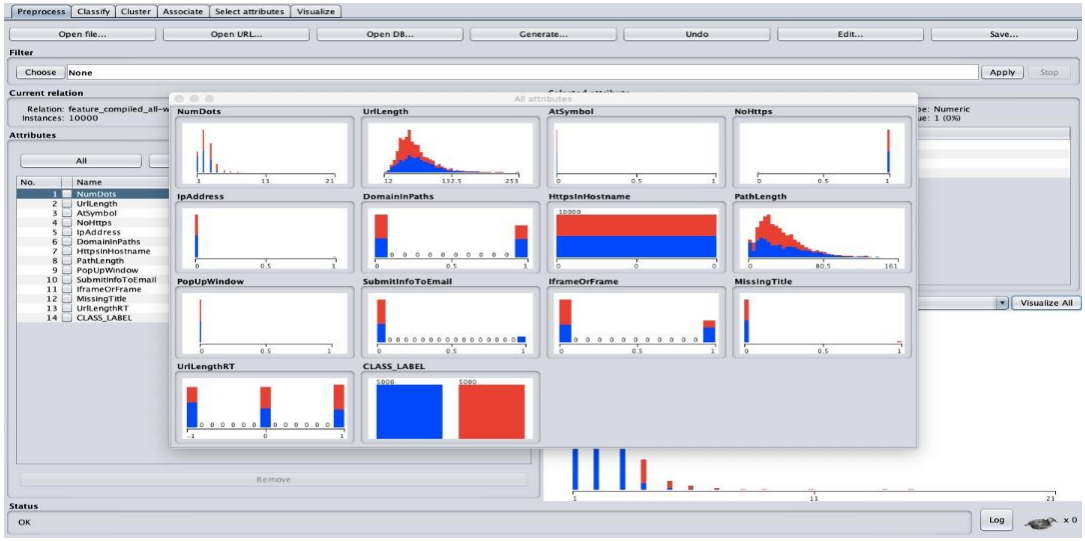

Figure 2. Feature Visualization in WEKA

Table 3 presented the experimental results for the different classifiers used in evaluating our phishing *fingerprints*. The classifiers in this experiment are Naïve Bayes, Support Vector Machine (SVM), Artificial Neutral Network (ANN), Random Tree (RT) and Decision Tree (DT). The results indicated that Random Tree outperforms other classifiers with significant accuracy of 96.1% and a ROC value of 98.7%. These results were next by the Decision Tree classifier with an accuracy of 78.2% and a ROC of 85.7%. The Multilayer perceptron model (ANN) performed next to DT with 74.6% accuracy and 82.4% ROC value. The SVM classifier produced an accuracy of 72.9% and a ROC value of 72.9%. The NB classifier was the least with 69.9% predictive accuracy and a ROC of 77.9%.

Classifier	TP	FP	Precision	Recall	F1-Score	ROC
RT	**96.1**	**0.39**	**96.1**	**96.1**	**96.1**	**99.7**
DT	**78.2**	**2.18**	**78.6**	**78.2**	**78.1**	**85.7**
ANN	**74.6**	**2.50**	**75.6**	**74.2**	**73.9**	**82.4**
SVM	**72.9**	**2.71**	**74.2**	**72.9**	**72.8**	**72.9**
NB	**69.9**	**3.01**	**70.2**	**69.9**	**69.8**	**77.8**

These results indicated that even the least performed classifier hover a well-above average (i.e. 50% prediction rate) in experimental results. Also, the range of the ROC values (i.e. 98-77%) is indicative of a good predictive accuracy of the selected classifiers and features.

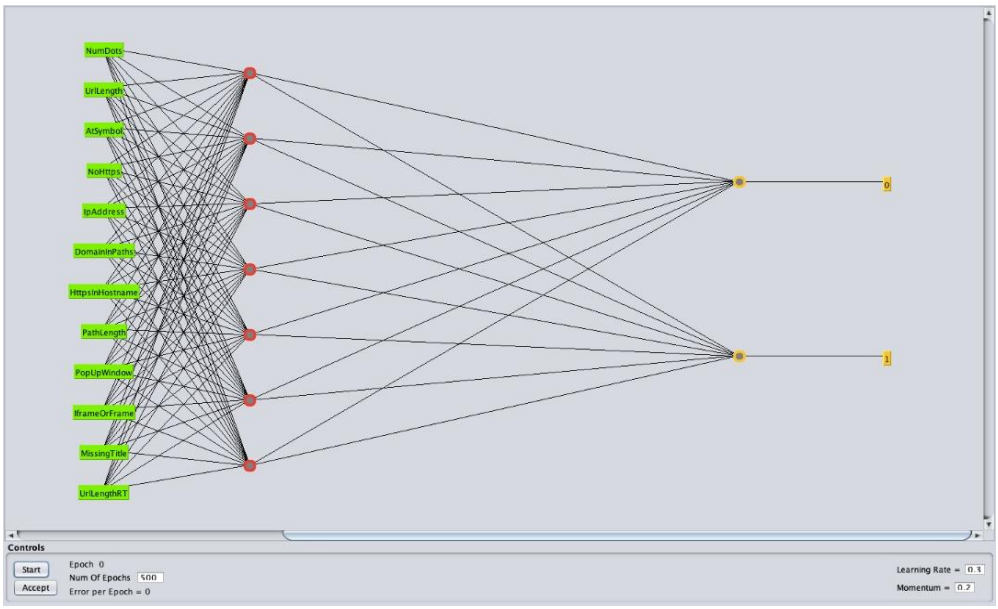

Figure 3. Visualization of ANN predictive model

Similarly, the low FP of RT models is a promising feature that indicates that the model has the potential application into critical web transaction for determining the status of a loading website. Thus, the predictive models based on the reduced phishing feature sets can produce a good generalization model for building efficient classifiers.

Figure 3 presented the Multilayer Perceptron of the ANN predictive model concerning feature input and its binary output value.

V. DISCUSSION

The research findings in this paper provide insights into the performance of different classifiers when exposed to reduced feature set technique. The results indicated that the Random Tree classifier outperformed other classifiers. The results of RT are better when compared with similar work by Galera et al. 2007 in which a framework for detecting and measuring phishing attacks was designed and analyzed. The authors used several URL heuristics to model a logistic regression classifier which produced a false positive rate of 0.7%. These results limit the application of their approach in critical web transactions in which sensitive financial data/online brand identity is involved. In more recent work, Karabatak and Mustafa (2018) investigated some heuristics to some specific classifiers to assess their performance comparison. In their work, the authors considered a reduced dataset with 27 features extracted using the Feature Selection algorithm from the extant literature. This is in sharp contrast with our work in which the extraction is based on frequency assessment. This implies that our feature selection algorithm gives better insight into the stability of each feature from the domain where it is selected by creating a frequency list as a weighting factor for their inclusion in the discriminative feature list. This provides the proposed system with a more minimal list i.e. 13 features when compared with 27 features used in Karabatak et al. 2018.

VI. CONCLUSIONS AND FUTURE WORKS

In this work, the performance evaluation of different classification models is considered for a smaller feature set. The features are selected from extant literature with particular consideration for URL features due to their sterling performance in existing works where they have been applied. These features are then trained and tested using 10000 phishing instances on five different classifiers. The experimental procedure was implemented using JSoup Parser and the WEKA application. The scheme uses the JSoup Parser to extract the selected features from the loading experiment instances. At the same time, WEKA provides the running environment for the preprocessing and performance evaluation for the different classifiers adopted in this work. The approach uses the cross-validation experiment to generalize and verify error performance associated with the different classifiers. Specifically, the scheme employed a 10-fold cross-validation experiment. The experimental results indicated that Random Tree outperforms other classifiers with remarkable accuracy and low false positive. These results showed that this approach presents a more accurate predictive model for mitigating phishing attacks.

In the future, we intend to incorporate incremental feature performance assessment for each classifier to determine which feature influence phishing detection significantly. This approach will assist the anti-phishing scheme to include more discriminating features in the composition of classifiers. Also, we hope to measure the response of different classifiers to this approach to determine their sensitivity to these features.

REFERENCES

[1] Action Fraud Security Report 2020

[2] Adebowale M., Lwin K., Sanchez E and Hossain M. (2018). Intelligent Web-Phishing Detection and Protection Scheme using integrated Features of Images, Frames and Text. Expert System with Applications.

[3] CSO Online report on phishing activities. Accessed 2016 (www.csoonline.com/articles)

[4] Chiew L., Chang H., Sze N and Tiong K. (2015.) Utilization of website logo for phishing detection. Computer and Security Journal.

[5] Garera S., Provos N., Chew M., and Rubin A. (2007). A Framework for Detection and Measurement of Phishing Attacks. In Proc. of WORM 07 ACM. USA

[6] Gowtham R and Krishnamurthi I. (2014). PhishTackle-a web services architecture for anti-phishing. Cluster Comput.

[7] Han W, Cao Y, Bertino E and Yong J. (2012).Using automated individual white-list to protect web

digital identities. Expert Systems with Applications.

[8] Hamid A and Abawajy, J. 2014. An approach to profiling phishing activities. Journal of computer and security. Elsevier Press

[9] Hota H.S, Shrivas A.K and Hota R. (2018). An Ensemble Model for Detecting Phishing Attack with Proposed Remove-Replace Feature Selection Technique. International Conference on Computational Intelligence and Data Science. Procedia Computer Science. Vo. 123, pp. 900-907

[10] Jain A and Gupta B. (2017). Two-level authentication approach to protect from phishing attacks in real-time. J. Ambient Intell Human Comp. DOI 10.1007/s12652-017-0616-z

[11] Jain AK, Gupta BB (2016) A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP J Inf Secur 2016:1–11

[12] Kanchan H, Laxmi A, S.K. Muttoo (2017) Detecting redirection spam using multilayer perceptron neural network. Soft Comput. 21 (13) 3803–3814.

[13] Khadi, S. Shinde, Detection of phishing websites using data mining techniques, Int. J. Eng. Res. Technol. 2 (12) (2014).

[14] LOC Security report, 2020

[15] Mao J, Bian J., Tian W., Zhu S., Wei T., Li. A. and Liang Z. (2019). Phishing Page detection via classifier from page layout feature. EURASIP Journal of Wireless Communication and Networking. Vol 43,

[16] Mohammad R and Thabtah L and McCluskey. (2014). Tutorial and critical analysis of phishing websites methods. Comp Sci. Rev. J

[17] Mohammad R, F. Thabtah, L. McCluskey, Predicting phishing websites based on self-structuring neural network (2014), J. Neural Comput. Appl. (ISSN: 0941-0643) 25 (2) 443–458. Springer.

[18] Oest A. Safaei Y. and Zhang P. (2020). PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists. 29th Usenix Security Symposium

[19] Orunsolu A., Afolabi O, Sodiya A and Akinwale A. (2018). A Users Awareness Study and Influence of Socio-Demography Perception of Anti-Phishing Security Tips. Acta Informatica Pragensia.

[20] Orunsolu A. Sodiya S. and Akinwale A. (2019). A Predictive Model for Phishing Detection. Journal of King Saud University-Computer and Information Sciences.

[21] Orunsolu A, Sodiya A and Kareem S. (2020). LinkCalculator- An Efficient Link-Based Phishing Detection Tool. Acta Informatica Malaysia

[22] Pham C, Nguyen L. Tan N, Huh N and Hong S. (2018). Phishing-Aware: A Neuro-Fuzzy Approach for Anti-Phishing on Fog Networks.

IEEE Transactions on Network and Service Management

[23] Prakash P., Kumar M., Kompella R and Gupta M (2010). PhishNet: predictive blacklisting to detect phishing attacks Proceedings of 29the Conference on Information Communications.

[24] Phishtank dataset (2018). http://www.phishtank.com.

[25] Qabajeh I., Thabtah F. and Chiclana F. (2018). A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Computer Science Review.

[26] Tan C., Chiew L and Sze N. (2017). Phishing Webpage Detection Using Weighted URL Tokens for Identity Keywords Retrieval. Lecture Notes in Electrical Engineering. Vol. 398.

[27] Stats and Trends Security report 2017

[28] Seymour J, P. Tully, Generative Models for Spear Phishing Posts on SocialMedia. Technical report, 2018.

[29] Silva C, Feitosa E and Garcia V. (2020). Heuristic-based strategy for phishing prediction: A survey of URL-based approach. Computers and Security Journal. Elsevier.

[30] Varshney G, Misra M. and Atrey P. (2016). A survey and classification of web phishing detection Schemes. Security Comm. Networks

[31] Varshney G, Misra M., and Atrey K. (2016). A phish detector using lightweight search features. Comput Secur;62:213–28.

[32] Zhu E, Ju Y, Chen Z, Liu F and Fang X. (2020). DTOF-ANN: An Artificial Neural Network phishing detection model based on Decision Tree and Optimal Features. Applied Soft Computing 95 10.1016/j.asoc.2020.106505