Feature Selection for Phishing Website Classification

Shafaizal Shabudin[1]
Information Management Department
Ministry of Works, Kuala Lumpur
Malaysia

Khairul Akram Zainal Ariffin[3]
Center for Cyber Security
Universiti Kebangsaan Malaysia
Selangor, Malaysia

Nor Samsiah Sani[2]*
Center for Artificial Intelligence Technology (CAIT)
Universiti Kebangsaan Malaysia

Mohd Aliff[4]
Instrumentation and Control Engineering
Malaysian Institute of Industrial Technology
Universiti Kuala Lumpur, Malaysia

***Abstract*—Phishing is an attempt to obtain confidential information about a user or an organization. It is an act of impersonating a credible webpage to lure users to expose sensitive data, such as username, password and credit card information. It has cost the online community and various stakeholders hundreds of millions of dollars. There is a need to detect and predict phishing, and the machine learning classification approach is a promising approach to do so. However, it may take several phases to identify and tune the effective features from the dataset before the selected classifier can be trained to identify phishing sites correctly. This paper presents the performance of two feature selection techniques known as the Feature Selection by Omitting Redundant Features (FSOR) and Feature Selection by Filtering Method (FSFM) to the 'Phishing Websites' dataset from the University of California Irvine and evaluates the performance of phishing webpage detection via three different machine learning techniques: Random Forest (RF) tree, Multilayer Perceptron (MLP) and Naive Bayes (NB). The most effective classification performance of these machine learning algorithms is further rectified based on a selected subset of features set by various feature selection methods. The observational results have shown that the optimized Random Forest (RF_{PT}) classifier with feature selection by the FSFM achieves the highest performance among all the techniques.**

Keywords*—Relevant features; phishing; web threat; classification; machine learning; feature selection

I. Introduction

Phishing is a simple yet complex mechanism that escalates threats to the security of the Internet community. With little information about the victim, the attacker can produce a believable and personalized email or webpage. It is also hard to catch the attacker, as most of them tend to hide their location and work in almost complete anonymity [1]. Even with high technology and excellent security software, users can become victims of this scheme. This is due to the huge of number of methods that can be used by the attackers to attract users into their phishing scheme. A report by Forbes has highlighted that approximately $500 million losses related to phishing attacks occur every year in the US businesses.

Phishing is defined as an attack to lure users to a fake webpage that masquerades as a legitimate website and aims to obtain disclosed personal data or credentials. The largest phishing campaign is conducted using spam emails to direct users to fake webpages [2] using impersonation techniques such as email spoofing and Domain Name System (DNS) spoofing and as well as social engineering. In addition, a phished website also tries to mimic the legitimate source by numerous methods, such as embedding some important contents imported directly from the legitimate website [3] and using similar keywords that refer to the target, including the title, images, and links [4,5].

A study by Hassan et al. raises concern on the methods used to detect and filter phishing webpages or emails successfully. Phishing can be considered as a semantic attack that easily tricks the users by crafting deceptive semantic techniques. The phrases in the phishing vector, especially through emails, are Lure, Hook, and Catch [6]. Two mechanisms are suggested to defend against this phishing vector: developing awareness programmes and deploying the detection and filtering systems. Awareness programmes are designed to educate users by implementing phishing defensive training such as that found in [7], [8] and [9]. Whereas for the deployment of technical defences against phishing, one can apply the two-factor authentication in a robust secure email [10], use disguised executable file detection [11], analyse and detect executable files transferred via emails, and add another layer of security by warning a user when abnormal data in the header source code are detected, such as in the spoofed email [12].

II. Phishing Mechanism in Cyber attack

The establishment of a cyber-attack may undergo some phases to achieve its objectives. It can take up to seven phases, such as reconnaissance, weaponization, delivery, exploitation, installation, command and control, and action on the objectives [13,14]. Thus, this attack can utilize phishing in delivery phases. It is started when the attacker learns about the target organization, either through webpages or any downloaded materials. Then, the attacker puts malicious code into a delivery vehicle, such as a fake webpage or an attachment. In the context of the fake webpage, the attacker clones the targeted official webpage with several input fields (e.g., text box, image). The attachment and link to the fake webpage can

*Corresponding Author

also be sent to users through email to attract thousands of victims. In addition, it is also possible in spreading phishing link and fake webpages with the aid of blogs, forums and so forth [15].

Before the phishing webpage is loaded to feed to the victims, the attacker will utilize technical subterfuge and also social engineering methods in the weaponization phase. In general, the attackers apply social engineering when they send bogus emails. In this kind of technique, the aim is to convince the recipients to respond with sensitive information. This information can be the name of banks, credit card companies, and e-retailers [16-17]. In technical subterfuge techniques, the attacker will implant malware into the victim's system to steal the credentials by using Trojans and keyloggers [15].

The malware can also mislead the victims to the fake webpage or proxy server. In most cases, the attacker attaches the malware or malicious link to the fraudulent email to distribute malicious software. According to the Symantec report [18], spear phishing, which is the act of targeting a specific group of people or organization, is the prime method employed by attackers in 2017. Then, when users open or click to the fraud hyperlink, malicious software is quietly installed on users' system. This malicious software will reside in users' system and collect confidential data from the system, for example, through keylogger software that captures the details of each key hit made by users. The command and control server, together with the Trojan, allows the attackers to gain remote access to users' system and collect data whenever they want.

Although there are numerous counter phishing researches carried out in the past, phishing is still a severe problem, not only because of the rapid growth in the number of these websites but also because the attackers are becoming better in being able to counter the countermeasures. This research's motivation is to form a flexible and effective technique that employs machine learning algorithms and tools to detect phishing websites. Predicting phishing websites is very useful when using the classification technique. The results can define phishing website indicators and characteristics together with their relations. Comparing between different classifications techniques with various pre-processing methods is also an objective to discover the best combination for the best prediction performance.

Machine learning has made dramatic improvements and is a core sub-area of artificial intelligence. It also enables computers to discover themselves without being explicitly programmed. A set of machine learning algorithms can be used to obtain meaningful insights into the data that help make effective detection on phishing websites. However, it is still very far from reaching human performance. The machine still needs human assistance to predefine the algorithms on initialization.

This paper highlights the phishing webpage detection mechanism based on machine learning classification techniques. The rest of the paper is organized in the following manner: Section 3 presents the phishing website research methodology, Section 4 presents the utilization of machine learning classification techniques, and Section 5 presents the experimental results gained after the implementation of the classification data mining methods in the phishing training datasets.

III. Methodology

Machine learning is one of the most exciting recent technologies. Machine learning had been positioned to address the shortages of human cognition as well as information processing, specifically in handling large data, their relations and the following analysis [19-23]. In general, machine learning studies the research and algorithms construction that can learn from, and derive predictions about, data [24,25]. Therefore, the machine learning approach is selected to predict whether a website, according to a dataset with some extracted features, is legitimate or phishing. Some extracted features acquire the same influence level on classifier accuracy to predict phishing sites and are considered as redundant. Optimization classification performance was conducted in determining the most effective features among all the features extracted [24]. Various feature selection methods were applied to reduce the features that are not relevant and group the reduced features as a new subset. Finally, the experiments required in analysing the extent to which the established machine learning techniques are effective in determining the most effective subset of features were also carried out.

A. Classification Techniques for Predictions

1) Random forest tree: The Random forest (RF) model was proposed in 2001 by Breiman based on the bagging approach. It is nonparametric statistical and an ensemble classification prediction model [26]. The model builds the forest at random, and the huge number of trees in the forest that is forming a combined forecasting model. The model prediction accuracy is improved through the summary of many classification trees. The random nature of two aspects is represented by the outstanding characteristic of the RF model. Firstly, the training samples are the original samples' resampling bootstrap, and the training samples are randomized. Secondly, in the process of building every tree, the input variables which are the best grouping variables at present which serve as the optimal variables of a stochastic candidate input variable subset for all variables with the variables randomized. This technique is an ensemble of decision trees that aims at constructing a multitude of decision trees within the training data and generating the class as an output. Table I illustrates the pseudo code of the algorithm.

TABLE I. Random forest pseudo code

1. For simple Tree T
2. For each node
3. Select m a random predictor variable
4. If the objective function achieved (m = 1)
5. Split the node
6. End if
7. End for
8. Repeat for all nodes

2) Multilayer perceptron: Multilayer Perceptron (MLP) is an artificial neural network model which could be employed for data classification [27]. Artificial neural network terminology is the way human brain neurons function and also interact simultaneously for recognition, reasoning, as well as recovery of damage [28]. It is also called a multi-layer feed forward neural network. This algorithm learns by finding the most suitable synaptic weight in classifying patterns in the training dataset. Neurons in the network are being connected with one another through a link called synaptic. Multilayer perceptron is an artificial neural network structure which is also a nonparametric estimator that can be employed for classifying and detecting intrusions. Table II illustrates the pseudo code of the algorithm.

3) Naive bayes: Naive Bayes (NB) is a classification technique that makes use of the Bayes theory which is based on probability and statistical knowledge [29]. This technique was founded by Thomas Bayes in the 18th century. Each instance $x = \{x_1, x_2, .., x_d\}$ of data set x is assumed to belong to exactly one class. Decision-making with regards to the Bayes theorem is relating to the inference probabilities which gather knowledge pertaining to prior events by predicting events using the rule base. The Naive Bayes classification consists of independent input variables which assume that the presence of an articular feature of a class does not have any relation to the presence of other features. Table III illustrates the pseudo code of the Naive Bayes algorithm.

B. Data Description

The data set came from the University of California Irvine (UCI) repository of machine learning databases under the name 'Phishing Websites' [30]. The dataset consists of 11,055 instances with 6,157 samples labelled as legitimate and 4,898 samples labelled as phishing. The choice of this dataset is due to its richness in the extracted features from various categories, which will be described in the next subsection. This dataset can be considered as equally distributed because the margins between the two classes were small.

C. Features Selection and Pre-Processing

Feature selection is a process to improve classification accuracy by removing irrelevant and redundant features from the original dataset [31]. Feature selection, also known as attributes selection, is used to reduce the dimensionality of the dataset, increase the learning accuracy, and improve result comprehensibility. In this study, two ranking methods, Feature Selection by Omitting Redundant Features (FSOR) and Feature Selection by Filtering Method (FSFM), are evaluated. A total of 30 extracted features from the phishing webpage dataset was identified, as shown in Table IV.

In feature selection, the following methods are implied to remove the ineffective features. The purpose of these methods is to increase the classification performance.

- Feature Selection by Omitting Redundant Feature (FSOR)

FSOR is applied by following an assumption that the features with the same degree of accuracy and influence are redundant, therefore they should be removed from the dataset. The FSOR process is implemented by using the Relief Ranking Filter to rank all extracted features before the desired features are chosen. Kira and Rendell introduced the Relief Algorithm in 1992 [32]. For an attribute to be classified useful, the attribute should be able to differentiate instances from various classes and yield the same value for instances in the same class [33]. The Relief Algorithm randomly samples an instance from the training data, and later locates a nearest sample that is from the same class termed as the nearest hit, and one other from a different class termed as the nearest miss. The feature values of the nearest neighbours are being employed in updating the relevant weights of features. Then, the feature weights are ranked, features with weights exceeding a specific threshold are chosen when forming the effective feature subset.

TABLE II. MULTILAYER PERCEPTRON PSEUDO CODE

1. For iteration = 1 to t
2. For e = 1 to *n* (all examples)
3. *x* = input for example *e*
4. *y* = output for example *e*
5. *w* = weights
6. *a* = activation function
7. *d* = derivative of activation function
8. For each *i* input neuron, compute $y_i = x_i$
9. For each *j* hidden neuron, compute $yj = \Sigma_i\, a\, (w_{ji} \cdot output_i)$
10. For each *k* hidden neuron, compute $yk = \Sigma_i\, d\, (w_{ji} \cdot output_i)$
11. output = $\{output_k\}$
12. Repeat

TABLE III. NAÏVE BAYES PSEUDO CODE

Input: Dataset *D*
For each Feature *f*
Compute the assumptions of *f* values based on class label 1
End for
For each Feature *f*
Compute the assumption of *f* values based on class label 2
End for
Prediction class = Maximum (assumption label 1, assumption label 2)
Repeat for all features

TABLE IV. EXTRACTED FEATURES

ID Feature	Feature Name
1	Using the IP Address
2	URL-Length
3	Shortening-Service
4	having-At-Symbol
5	double-slash-redirecting
6	Prefix-Suffix
7	having-Sub-Domain
8	SSLfinal-State
9	Domain-registration-length
10	Favicon
11	port
12	HTTPS-token
13	Request-URL
14	URL-of-Anchor
15	Links-in-tags
16	SFH
17	Submitting-to-email
18	Abnormal_URL
19	Redirect
20	On-mouseover
21	RightClick
22	popUpWindow
23	Iframe
24	Age-of-domain
25	DNSRecord
26	Web-traffic
27	Page-Rank
28	Google-Index
29	Links-pointing-to-page
30	Statistical-report

The fundamental concept of Relief Ranking Filter lies in drawing instances at random, later computing their nearest neighbors, and also adjusting a feature weighting vector in order to provide more weight to features that differentiates the instance from neighbors of different classes. In particular, the Relief Ranking Filter attempts in locating a good estimate for the probability that follows be assigned as the weight for every feature f as depicted in (1).

$$w_f = p_d\left(\frac{x}{c_d}\right) - p_s\left(\frac{x}{c_s}\right), \quad (1)$$

where w is the weight for every feature *f*, P_d is probability different value of feature *x* of different classes c_d and P_s is probability different value of feature *x* of different the same class c_s. This method yields good performance in numerous domains [33].

- Feature Selection by Filtering Method (FSFM)

Feature selection is the identification and elimination process of irrelevant and redundant information as much as possible. Fewer attributes is desirable because it dwindles the complexity of the model and enables faster and effective operation of the learning algorithms. In the process of assigning a scoring for every feature, a statistical measure is applied by the filter feature selection methods [34]. The ranking of features is based on the score and it is chosen either to be removed or kept from the dataset. The techniques are usually univariate and take the feature into consideration independently, or with regard to the dependent variable. The FSFM process is implemented by using Information Gain (IG). IG is a crucial measure that is used for ranking and it measures the extent to which the features are mixed up [35]. Also, IG is employed in measuring the relevance of attribute K in class L. As the mutual information value between classes K and attribute L gets higher, the relevance between classes K and attribute L gets higher, as shown in (2).

$$\mathrm{IG}(\mathrm{L},\mathrm{K}) = \mathrm{H}(\mathrm{L}) - \mathrm{H}(\mathrm{L} \mid \mathrm{K}), \quad (2)$$

where $H(L) = -\sum_{cEL} P(L)\log P(L)$, the entropy of the class $\mathrm{H}(\mathrm{L} \mid \mathrm{K})$, and is the conditional entropy of class given attribute, $H(L|K) = -\sum_{cEL} P(L|K)\log P(L|K)$. Since Phishing Websites dataset has balanced class, the probability of class for both positive and negative is 0.5. Consequently, the entropy of classes *P(L)* is 1. Later, the information obtained could be formulated as in (3).

$$\mathrm{G}(\mathrm{L},\mathrm{K}) = 1 - \mathrm{H}(\mathrm{L} \mid \mathrm{K}), \quad (3)$$

The minimum value of $G(L,K)$ happens if only if $H(L \mid K) = 1$ which indicates that attribute *K* and *L* classes have no relation to one another at all. In contrast, there is a tendency to select attribute *K* that usually appears in one class *L* either positive or negative. In other words, a set of attributes that appear only one in one class are classified as the best features This indicates that the maximum $\mathrm{IG}(\mathrm{L} \mid \mathrm{K})$ is attained when $P(K)$ is equivalent to $P(K|L_1)$ resulting in $P(L_1|K)$ and $H(L_1|K)$ being equivalent to 0.5. When $P(K) = P(K|L_2)$, then the value of $P(K|L_2)$ results in $P(L_1|K) = 0$ and $H(L_1|K) = 0$. The value of $\mathrm{IG}(\mathrm{L} \mid \mathrm{K})$ is varied from 0 to 0.5.

Table V shows the ranking of the extracted features after applying the FSFM and FSOR method. The features number is different from the result of full extracted features because the sequences were renumbered after the removal of redundant features. Eleven features have been selected as the best accuracy for each classifier. In this method, the feature with a weight value of less than 0.05 is considered to be ineffective. There are 22 attributes that have been selected, which are presented by ID Features of 11, 7, 18, 6, 5, 12, 13, 21, 1, 19, 2, 16, 3, 17, 4, 9, 14, 22, 10, 20, 8 and 15. With the reduction of the number of features, the processing time can be reduced and the performance can also increase, especially when operating on a lower specification computer.

TABLE V. ATTRIBUTES RANKING BY USING RELIEF RANKER WITH SELECTED FEATURES THROUGH FSOR

Rank	Weight	ID Feature	Feature Name
1	0.45	11	URL_of_ Achor
2	0.39	7	SSLfinal_State
3	0.23	18	web_traffic
4	0.12	6	having_Sub_Domain
5	0.11	5	Prefix_Suffix
6	0.11	12	Links_in_tags
7	0.08	13	SFH
8	0.06	21	Links_pointing_to_page
9	0.05	1	Having_IP_Address
10	0.05	19	Page_Rank
11	0.05	2	URL_Length
12	0.04	16	Age_of_domain
13	0.04	3	Shorting_Service
14	0.03	17	DNSRecord
15	0.03	4	Having_At_Symbol
16	0.03	9	Port
17	0.03	14	On_mouseover
18	0.02	22	Statistical_report
19	0.02	10	Request_URL
20	0.02	20	Google_Index
21	0.02	8	Domain_registration_Length
22	0.01	15	RightClick

IV. ANALYSIS AND EVALUATION

The experiment on the phishing webpage dataset is applied on three common machine learning algorithms to create the classification models to detect phishing URLs. The dataset is classified into three classes as legitimate, suspicious and phishing with respective labels of '1', '0' and '-1'. The three selected classifiers are Random Forest Tree, Multilayer Perceptron and Naive Bayes. The 10-fold cross validation testing is employed in evaluating the classifiers.

A. Evaluation without Feature Selection

We select several learning techniques to benchmark the phishing website classification performance. These are Random Forest, Multilayer Perceptron and Naive Bayes, and all are supervised learning techniques. A key characteristic of supervised machine learning techniques is their selection of the appropriate technique with appropriate features. Table VI depicts the classification results of three selected classifiers by using all the extracted features from the dataset. It can be observed from the table, the values of overall accuracy, Random Forest tree and Multilayer Perceptron classifiers are closest to each other. The Naive Bayes classifier gives the lowest accuracy. The Random Forest tree classier exceeds the two other classifiers in terms of overall accuracy as it attains an accuracy of 96.98% with 15 seconds processing time. Next, the Multilayer Perceptron classifier achieves an accuracy of 96.32% with 945 seconds, while the Naive Bayes classifier achieves an accuracy of 92.94% with 1 second processing time. The Random forest (RF) model Numbered lists can be added as follows:

B. Evaluation with Omitting Redundant Features (FSOR)

The most effective subset of features is chosen by eliminating the ineffective ones and the corresponding performance for every classifier. As seen in Table VII, nine features, which are ID Feature 3, 5, 10, 12, 17, 18, 19, 22 and 23, have the same accuracy from the classification with three classifiers. Based on the results, only ID Feature 3 is selected to represent the other redundant features with an assumption that all features with the same accuracy are redundant and have the same degree of influence. A total of 22 features are selected from the balance features after removing the redundant features. This process reduced features by approximately 27% from the total extracted features.

Table VIII shows the classification accuracy based on features selection by FSOR. As seen from the results, the accuracy with Random Forest, Multilayer Perceptron and Naive Bayes classifiers achieved accuracies of 97.08%, 96.51% and 92.98%, respectively. The overall accuracy is improved on average by 0.2% from the accuracy of using all extracted features. In conclusion, one feature from the redundant feature group was enough to represent this group of features, and the processing time also improved by 40%

C. Evaluation with Omitting Redundant Features (FSOR)

Table IX shows the classification accuracy based on features selection by FSFM. As shown in Table IX, the results show an improvement in processing time, but the accuracy for all classifiers have decreased a little bit. This indicates that the coloration between features, excluding redundant features, is still high even when the weight is small. However, from an overall point of view, this is considered as a good overall performance, as it can provide a significant improvement on processing time with more than 95% accuracy. This classification model can be used to speed up the process with a lower specification computer by losing some accuracy.

D. Random Forest Parameterization

A key characteristic of supervised machine learning techniques lies in the selection of appropriate techniques with appropriate features and parameters35. From the observations during the feature selection step in Sections B and C, the findings showed that the most effective classification method is Random Forest. To improve the performance of the best classifier (i.e., Random Forest), a parameter tuning experiment was carried out. The experiment was conducted in order to identify the most suitable parameterization set of the Random Forest model to be employed, as the model has several alternatives and options that would define the method's success. The classifier is tuned using different tuning parameters to produce high accuracy results. The optimized RF with best parameters setup is indicated as $RF_{PT.}$

TABLE VI. CLASSIFICATION RESULT OF THREE SELECTED CLASSIFIERS

Classifier	Processing Time	Accuracy
Random Forest	15 second	96.98%
Multilayer Perceptron	945 seconds	96.32%
Naive Bayes	1 second	92.94%

TABLE VII. CLASSIFICTION RESULT OF THREE SELECTED CLASSIFIERS FOR EXTRACTED FEATURES

ID	Feature Name	Classifier		
		RF	MLP	NB
1	Using the IP Address	56.23%	55.74%	56.23%
2	URL-Length	55.97%	55.97%	55.97%
3	Shortening-Service	55.69%	55.69%	55.69%
4	having-At-Symbol	55.65%	55.83%	55.43%
5	double-slash-redirecting	55.69%	55.69%	55.69%
6	Prefix-Suffix	57.56%	57.06%	57.56%
7	having-Sub-Domain	66.47%	66.11%	66.47%
8	SSLfinal-State	88.89%	88.89%	88.89%
9	Domain-registration-length	62.48%	62.48%	62.48%
10	Favicon	55.69%	55.69%	55.69%
11	port	55.69%	55.42%	55.69%
12	HTTPS-token	55.69%	55.69%	55.69%
13	Request-URL	63.43%	63.43%	63.43%
14	URL-of-Anchor	84.73%	84.73%	84.73%
15	Links-in-tags	63.09%	63.09%	63.09%
16	SFH	55.75%	55.79%	56.02%
17	Submitting-to-email	55.69%	55.69%	55.69%
18	Abnormal_URL	55.69%	55.69%	55.69%
19	Redirect	55.69%	55.69%	55.69%
20	On-mouseover	55.41%	55.41%	55.37%
21	RightClick	55.69%	55.44%	55.69%
22	popUpWindow	55.69%	55.69%	55.69%
23	Iframe	55.69%	55.69%	55.69%
24	Age-of-domain	56.37%	55.95%	56.37%
25	DNSRecord	55.08%	55.63%	55.14%
26	Web-traffic	69.79%	69.79%	69.79%
27	Page-Rank	55.69%	54.94%	55.69%
28	Google-Index	58.54%	58.24%	58.54%
29	Links-pointing-to-page	55.69%	55.35%	55.69%
30	Statistical-report	56.85%	56.60%	56.85%

TABLE VIII. CLASSIFICATION RESULTS FOR FEATURE SELECTION BY OMITTING REDUNDANT FEATURES (FSOR)

Classifier	Selected Features	Processing Time	Accuracy
RF	1,2,3,4,6,7,8,9,11,13,14,15,16,20,21,24,25,26,27,28,29,30	10 seconds	97.08%
MLP		600 seconds	96.51%
NB		1 second	92.98%

TABLE IX. CLASSIFICATION RESULTS FOR FEATURE SELECTION BY FILTERING METHOD (FSFM)

Classifier	Selected Features	Processing Time	Accuracy
RF	14,8,26,7,6,15,16,29,1,27,2	6 seconds	95.19%
MLP		360 seconds	95.01%
NB		1 second	92.43%

Based on the Random Forest program developed which is followed by various studies on Random Forest parameterizations, the three key parameters required by the Random Forest were identified: (a) the maximum depth of the tree (maxDepth); (b) the desired batch size for batch prediction (batchSize); and (c) the number of iterations (numIterations).

A set of initial default parameter values was the first to be defined, which was consisting of a 100 batchSize, a 0 maximum depth of the tree (maxDepth) and 100 iterations (numIterations). Individual parameters investigated were altered while keeping the other default parameters mentioned above intact. The specifications of the parameter values that were tested are as follows: (a) the maxDepth was carried out in a potential range between 1 and 50; (b) a number of different batchSize were tested, ranging from 10 to 100 in steps of 10; and (c) with regards to the numIterations parameter, a few different values were being tested beginning from the smallest value of 100 to the largest value of 200. This process as applied on extracted features selected by FSOR and FSFM. One at a time, every parameter was changed to record the parameters' performance variation systematically. This ensured that the effect of parameter variation was quantified individually in an accurate manner. The parameters were performed, and then the results attained are discussed.

Fig. 1 shows the default parameter value '0' for maxDepth achieving 97.08% accuracy for FSOR and 95.19% accuracy for FSFM. Value '1' for maxDepth achieved the lowest accuracy of 90.64% and 90.77% for FSOR and FSFM, respectively. Value '1' for maxDepth can be considered as an initial point to tune the performance by using the maxDepth parameter and the maxDepth default value as a benchmark. The accuracy increases significantly with the increment of the maxDepth value at the beginning but then starts to become static for both feature groups. Accuracy for FSOR and FSFM features become static at maxDepth values of 14 and 12, respectively. Parameter value 13 for maxDepth achieved the highest accuracy of 97.12% and it showed that the larger maxDepth number will not necessarily produce better results.

The second parameter to be tuned is numIterations. The initial value for numIterations is 100. Then, it will test with values of 101 to 110, 120, 130, 140, 150, 160, 170 and 200. The result shows that for accuracy, there is a fluctuation at the beginning of the test for omitting redundant features until reaching 110 before it starts to decrease. In comparison, the accuracy for filtered features shows less fluctuation as the value changes. Fig. 2 shows numIterations for omitting redundant features achieving the highest accuracy of 97.13% at 105 and 95.19% at 140. It highlights that the filtered features achieve multiple points of highest accuracy, but choosing the lowest number of iterations is the best practise in order to obtain better prediction performance.

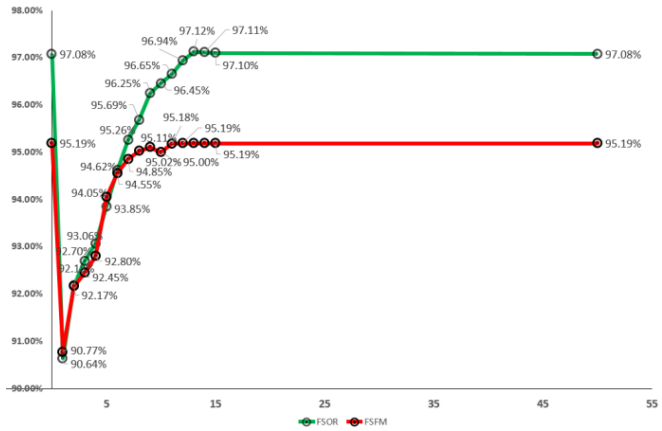

Fig. 1. Accuracy of MaxDepth from Parameter Tuning based on FSOR and FSFM Features.

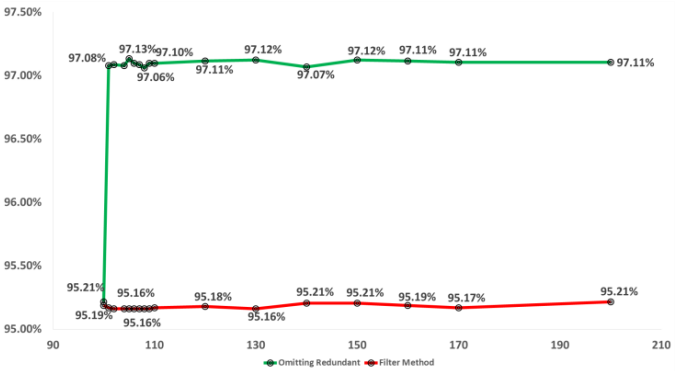

Fig. 2. Accuracy of NumIterations from Parameter Tuning based on Omitting Redundant (FSOR) and Filtering Method (FSFM) Features.

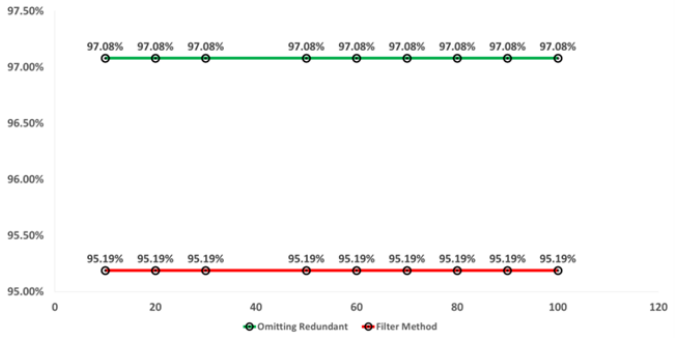

Fig. 3. Accuracy for BatchSize from Parameter Tuning based on Omitting Redundant and Filtering Method Features.

The final parameter to configure in this performance tuning was batchSize. The result shows that changing the parameter values for batchSize will not change the accuracy for both feature groups, as shown in Fig. 3. Therefore, in this study, the parameter value will remain at 10 for batchSize with an accuracy of 97.08% and 95.19% for omitting redundant and filtering method features, respectively.

Unlike filtering method features, using both th e best parameter results for numIterations and maxDepth only leads to a lower accuracy for omitting redundant features. It achieves a slightly lower accuracy of 97.11% compared to the accuracy achieved by using the default parameter value. The filtering method features achieve the highest accuracy of 95.21% by using both best results from numIterations and maxDepth. Several tests were executed by mixing and matching the results of numIterations and maxDepth for omitting redundant features, and from the observation, these combinations will achieve a higher accuracy of 97.18% by combining 105 numIterations and 14 maxDepth.

Parameter tuning is an important part of the data pre-processing, as it improves classification accuracy. In this case, the accuracy increases by 0.10% from 97.08% to 97.18% for omitting redundant features selection and increased by 0.02% from 95.19% to 95.21% for filtering method feature selections when tuning the parameter numIterations and maxDepth for the Random Forest tree algorithm. On the basis of these parameterization experiments, the following RF parameters were chosen to be applied for the subsequent experiment. The finalized parameters are batchSize of 10, maxDepth of 14, and 105 numIterations.

V. Discussion

This paper proposed an improvised classification performance based on the pre-processing and parameter tuning. The pre-processing stage involves two feature selection methods, which are Feature Selection by Omitting Redundant Features and Feature Selection by Filter Method. The empirical results for feature selection in Table X show that Feature Selection by Omitting Redundant Features achieves the highest accuracy of 97.18%, while the Feature Selection by Filtering Method displays the lowest accuracy result, which is 95.21% for the RF_{PT} classifier. However, processing time is increasing alongside the classification performance. The RF Classifier with 22 features from the dataset presents an increment in performance for both accuracy and processing time, as shown in Table X.

Furthermore, in this study, a paired corrected T-test was performed. The statistical test is used to identify whether the performance of the two features selection method is statistically significantly different or one that is better than the other[22]. The T-test was conducted to compare the performance between two feature selection techniques (i.e., FSOR and FSFM) on three classifiers (i.e., RF_{PT}, MLP, NB). In this test, the accuracy results of all feature selection methods (i.e., FSOR and FSFM) on three classifiers (i.e., RF_{PT}, MLP, NB) are collected, and their significance of difference is tested using the T-test. The results show that FSOR is the best performer when using Random Forest as a classifier, and the result is statistically significant at the 0.05 level. Additionally, the T-test shows that there are statistically significant differences between the performances of the three classifiers (i.e., RF_{PT}, MLP, NB), which is significant at the 0.05 level. In a nutshell, these results indicate the presence of significant differences between the FSOR and FSFM methods when applied on the Random Forest classifier. Hence, the performance of the Random Forest (i.e., RF_{PT}) method can be said to be better than that of the other classifiers (i.e., MLP, NB).

TABLE X. STATISTICAL TESTS FOR CLASSIFICATION

Feature Selection Methods	Indicators	Classifier		
		RF_{PT}	MLP	NB
Feature Selection by Omitting Redundant (FSOR)	Accuracy	97.18%	96.51%	92.98%
	Processing Time	12 seconds	600 seconds	1 second
	Number of Features	22		
Feature Selection by Filter Method (FSFM)	Accuracy	95.21%	95.01%	92.43%
	Processing Time	8 seconds	360 seconds	10 seconds
	Number of Features	9		

VI. CONCLUSION

This study provides a comparison of performance between two feature selection methods (i.e., Feature Selection by Omitting Redundant and Feature Selection by Filter Method) in classifying phishing websites. The performance of each feature method was compared based on the classification accuracy of three classifier methods (Random Forest Tree, Multilayer Perceptron and Naive Bayes). Before comparing the performances, a few pre-processing techniques like data cleaning, feature selection, and parameter tuning were conducted. Statistical relevance of the experimental results was determined by the paired T-test. The results demonstrate that the FSOR method is statistically significant and outperforms the other method when using Random Forest classifiers. Hence, we can conclude that phishing website classification with 23 features (i.e., Using the IP Address, URL-Length, Shortening-Service, having-At-Symbol, double-slash-redirecting, Prefix-Suffix, having-Sub-Domain, SSLfinal-State, Domain-registration length, port, HTTPS-token, Request-URL, URL-of-Anchor, Links-in-tag, SFH, on-mouseover, RightClick, age-of-domain, DNSRecord, web-traffic, Page-Rank, Google-Index, Links-pointing-to-page, Statistical-report) will perform better if Random Forest is used instead of Naive Bayes and Multilayer Perceptron.

Future work can be conducted so that they serve as a comparison with the other latest machine learning algorithms, obtaining a higher accuracy but with less complexity. Classification performance can also be carried out using a larger dataset to confirm the effectiveness of processing time.

ACKNOWLEDGMENT

The authors would like to thank Universiti Kebangsaan Malaysia (UKM) and Ministry of Education, Malaysia (MOE) under the Research University Grant (project code: GUP-2019-060 and FRGS/1/2018/ICT02/UKM/02/6) for funding and supporting this research.

REFERENCES

[1] I. Vayansky, and S. Kumar, “Phishing – challenges and solutions,” Computer Fraud & Security, pp. 15-20, 2018.

[2] Phishing Activity Trends Report – 1st Quarter 2018. Available online: https://docs.apwg.org/reports/apwg_trends_report_q1_2018.pdf (accessed on: 1 February 2019).

[3] Y. Pan, and X. Ding, “Anomaly-based web phishing page detection”, In Proc. Of the 22nd ACSAC, IEEE, Miami, FL, USA, pp. 381-392, 2006.

[4] S. Marchal, G. Armano, T. Grondahl, K. Saari, N. Singh, and N. Asokan, “Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application,” IEEE Transaction on Computers, vol. 66, no. 10, pp. 1717-1733, 2017.

[5] A. K. Jain, and B. B. Gupta, “Comparative analysis of features based machine learning approaches for phishing detection,” In Proc. INDIACom, IEEE, New Delhi, India, 2016.

[6] H. Y. A. Abutair, and A. Belghith, “Using Case-Based Reasoning for Phishing Detection,” Procedia Computer Science, vol. 109, 281-288, 2017.

[7] N. A. Bakar, M. Mohd, and R. Sulaiman, “Information leakage preventive training,” In Proc. Of 6th ICEEI, IEEE, Langkawi, Malaysia, 2018.

[8] A. Carella, M. Kotsoev, and T. M. Truta, “Impact of security awareness training on phishing click-through rates,” In IEEE Proc. Big Data, IEEE, Boston, MA, USA, 2017.

[9] T. Steyn, H. Kruger, and L. Drevin, “Identity theft - empirical evidence from a phishing exercise” In New Approaches for Security, Privacy and Trust in Complex Environments; Venter, H.; Eloff, M.; Labuschagne, L.; Eloff, J.; von Sohns, R. Springer: Boston, MA, USA, vol. 232, pp. 193-203, 2007.

[10] A. Yasin, and A. Abuhasan, “Enhancing anti-phishing by a robust multi-level authentication technique,” IAJIT, vol. 15, pp. 990-999, 2018.

[11] I, Ghafir, V. Prenosil, M. Hammoudeh, F. J. Aparicio-Navarro, K. Rabie, and A. Jabban, “Disguised executable files in spear-phishing emails: Detecting the point of entry in advanced persistent threat,” In Proc. ICFNDS’18, ACM, Amman, Jordan, 2018.

[12] B. Opazo, D. Whitteker, C. C. Shing, “Email trouble: Secrets of spoofing, the dangers of social engineering, and how we can help,” 13th International Conference on Natural Computation, ICNC-FSKD, IEEE, Guilin, China, pp. 2812-2817, 2017.

[13] W. Harrop, and A. Matteson, “Cyber Resilience: A Review of Critical National Infrastructure and Cyber-Security Protection Measures Applied in the UK and USA,” In Current and Emerging Trends in Cyber Operations: Policy, Strategy and Practice; George Washington University, USA, Springer, 2015.

[14] A. Waleed, "Phishing website detection based on supervised machine learning with wrapper features selection," International Journal of Advanced Computer Science and Applications, vol. 8, pp. 72-78, 2017.

[15] K. Firdous, B. Al-Otaibi, A. Al-Qadi, and N. Al-Dossari, “Hybrid client side phishing websites detection approach,” International Journal of Advanced Computer Science and Applications,vol. 5, pp. 132-140, 2014.

[16] R. Sihwail, K. Omar, K. A. Z. Ariffin, “A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis,” IJASEIT, vol. 8, pp. 1663-1671, 2018.

[17] B. Opazo, D. Whitteker, S. J. Wang, T. Herath, R. Chen, A. Vishwanath, and H. R. Rao, “Research article phishing susceptibility: An investigation into the processing of a targeted spear phishing email, ”IEEE Trans. Prof. Commun., vol. 55, pp. 345-362, 2012.

[18] J. D. Holliday, N. Sani, and P. Willett, “Calculation of substructural analysis weights using a genetic algorithm,” J. Chem. Inf. Model, vol. 55, pp. 214-221, 2015.

[19] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, “Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection,” IEEE Access, vol. 6, pp. 33789-33795, 2018.

[20] J. D. Holliday, N. Sani, and P. Willett, “Ligand-based virtual screening using a genetic algorithm with data fusion,” Match-Commun. Math. Co., vol. 80, pp. 623-638, 2018.

[21] N. Sani, I. Shlash, M. Hassan, A. Hadi, and M. Aliff, “Enhancing Malaysia rainfall prediction using classification techniques” J. Appl. Environ. Biol. Sci, vol. 7, pp. 20-29, 2017.

[22] N. S. Sani, M. A. Rahman, A. A. Bakar, S. Sahran, and H. M. Sarim, “Machine learning approach for bottom 40 percent households (B40) poverty classification,” IJASEIT, vol. 8, pp. 1698-1705, 2018.

[23] A. Chelli, and M A. Pätzold, "Machine Learning Approach for Fall Detection and Daily Living Activity Recognition," IEEE Access, vol. 7, pp. 38670-38687, 2019.

[24] A. L'heureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz, "Machine learning with big data: Challenges and approaches," IEEE Access, vol. 5, pp. 7776-7797, 2017.

[25] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," IEEE Access, vol. 6, pp. 14277-14284, 2018.

[26] L. Breiman, "Random Forests," Mach. Learn, vol. 45, pp. 5-32, 2001.

[27] A. Majida, and H. Alasadi, "High Accuracy Arabic Handwritten Characters Recognition Using Error Back Propagation Artificial Neural Networks," International Journal of Advanced Computer Science and Applications, vol. 6, pp. 145-152, 2015.

[28] G. Carleo, and M. Troyer, "Solving the quantum many-body problem with artificial neural networks" Science, vol. 355, pp. 602-606, 2017.

[29] L. Li, Y. Zhang, W. Chen, S. K. Bose, M. Zukerman, and G. Shen, "Naïve Bayes classifier-assisted least loaded routing for circuit-switched networks," IEEE Access, vol. 7, pp. 11854-11867, 2019.

[30] The UC Irvine Machine Learning Repository. Available online: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites.

[31] N. S. Sani, I. I. S. Shamsuddin, S. Sahran, A. H. A. Rahman, and E. N. Muzaffar, "Redefining selection of features and classification algorithms for room occupancy detection ," IJASEIT, vol. 8, pp. 1486-1493, 2018.

[32] K. Kira, and L. A. Rendell, "Practical Approach to Feature Selection," Proc. Ninth Intl. Conf. on Machine Learning (ICML), pp. 249-256, 1992.

[33] M. Robnik-Šikonja, and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," Mach. Learn., vol. 53, pp. 23-69, 2003.

[34] M. R. Gray, "Entropy and Information Theory," Springer Science and Business Media: Stanford, CA, USA, 2011.

[35] A. I. Pratiwi, and K. Adiwijaya, "On the feature selection and classification based on information gain for document sentiment analysis," Appl. Comput. Intell. Soft Comput., vol. 2018, 2018, 1407817-1–1407817-5, 2018.