Iraqi Journal of Science, 2020, Vol. 61, No. 12, pp: 3387-3396
DOI: 10.24996/ijs.2020.61.12.27

ISSN: 0067-2904

Hybrid vs Ensemble Classification Models for Phishing Websites

Folorunso, S. O.[1*], Ayo, F. E.[2], Abdullah, K-K. A.[1], Ogunyinka, P. I[1]
[1]Department of Mathematical Sciences, Olabisi Onabanjo University, Ago-Iwoye, Nigeria
[2]Department of Physical and Computer Sciences, McPherson University, Seriki Sotayo, Nigeria

Received: 3/10/2019 Accepted: 21/1/2020

Abstract

Phishing is an internet crime achieved by imitating a legitimate website of a host in order to steal confidential information. Many researchers have developed phishing classification models that are limited in real-time and computational efficiency. This paper presents an ensemble learning model composed of DTree and NBayes, by STACKING method, with DTree as base learner. The aim is to combine the advantages of simplicity and effectiveness of DTree with the lower complexity time of NBayes. The models were integrated and appraised independently for data training and the probabilities of each class were averaged by their accuracy on the trained data through testing process. The present results of the empirical study on phishing website dataset suggest that the ensemble model significantly outperformed the hybrid model in terms of the measures used. Finally, DTree and STACKING methods showed superior performances compared to the other models.

Keywords: Phishing, Feature selection, Classification, Stacking, Ensemble, Social engineering

1. Introduction

Phishing is a fraudulent act that utilizes deceptive methods to deceive naïve internet users into sharing their personal information, such as usernames, passwords, credit card information, and bank account information, being under the impression of the website to be authentic [1]. Electronic-mail (Email) popularity as a means of low-cost and secure message transfer has made it a candidate for phishing. Hence, a phished Email can cause malware when used in fraud schemes, including advertisements and others [2]. Therefore, early detection of these phishing websites, through building machine learning models, is required so as to warn inexperienced users against sharing their personal and sensitive information through these s websites. An earlier report [3] discussed the blacklist and whitelist-based approach and the intelligent heuristics-based approach, which are the most popular methods for phishing website detection. Machine learning methods extract hidden patterns and useful information from different domain datasets, but most of these methods are still lacking real-time and computational efficiency.

Another report [4] discussed the characteristics that distinguish these websites and explored some extracted features in deciding their legitimacy. Furthermore, three different rule-based classification models were compared, namely Decision Tree (DTree), Repeated Incremental Pruning to Produce Error Reduction (RIPPER) and algorithm for Inducing Modular Rules (PRISM), and Classification Based on Association (CBA) algorithm [5]. The classification performance of these models was measured based on accuracy on all features in the dataset and reduced features based on Chi-Square. An exploratory technique for the detection and prevention of phishing along with user training solutions was proposed [6]. Rules were also generated, using association rule mining, to detect phishing from a website [7]. In addition, seven different machine learning models were applied to detect phishing and legitimate emails [2]. Count based and distributed representations were used for

*Email: sakinat.folorunso@oouagoiwoye.edu.ng,

word representation. Also, seven machine learning models such as Backpropagation Neural Network (BPNN), Radial Basis Function Network (RBFN), Support Vector Machine (SVM), Naïve Bayes (NBayes), C4.5, k-Nearest Neighbor (k-NN), and Random Forest (RF) were applied to detect phishing websites [3]. The study also applied three different feature selection methods to the dataset and compared their results based on True Positive Rate (TPR), True Negative Rate (TNR) and Geometric Mean (GM). Some feature selection schemes, such as individual, forward selection, backward selection, association rules, and plus-l take away-r were applied to phishing website datasets [8]. The resultant datasets were then applied to Naïve Bayes (NBayes), Bayesian Network (BN), Stochastic Gradient Descent (SGD), Iterative Dichotomiser (ID3), lazy.KStar, RandomizableFilteredClassifier, Logistic Model Tree (LMT), Multilayer Perceptron, RIPPER, PARTial Decision Tree (PART), DTree, RF, and Random Tree models for classification. These models were evaluated based on their classification accuracy. The use of Genetic Algorithm to perform feature selection and a two-stage Projection Pursuit (PP) algorithm was proposed to generate new features [9]. These new features were evaluated on machine learning models, such as SVM, Logistic Regression (LR), k-NN, Artificial Neural Network (ANN), and NBayes. Other authors [10] proposed filtering the phishing websites at the client side using URL, hyperlink, CSS, login form, and identity features. They created a new heuristic for each feature of the dataset collected from various sources and included the variety of websites to validate the proposed solution. The dataset obtained was evaluated on RF, SVM, ANN, LR and NBayes machine learning models. The results revealed very high efficiency, with 99.39% true positive rate and only 1.25 % false positive rate. A meta-heuristic-based nonlinear regression algorithm with wrapper feature selection scheme was proposed [11]. The University of California, Irvine (UCI)[12], phishing website was evaluated on Harmony Search (HS) and SVM models. DTree and wrapper feature selection methods obtained the highest classification accuracy of 96.32%, while the proposed scheme obtained accuracy rates of 94.13 and 92.80% for train and test processes, respectively.

In order to combat the limitations with machine learning models, this study presents an ensemble of base learners which includes DTree and NBayes to classify phishing website dataset. The simplicity and effectiveness of DTree is integrated with the low time complexity of NBayes to build an NBTree, unlike the hybrid classification models that incurs very high time complexity. The ensemble of DTree and NBayes presented by this work is achieved by STACKING method with DTree as base learner. Wrapper feature selection scheme was also applied to the original dataset to select features that are important and contribute more to the classification process. Furthermore, the performance of the full and reduced feature datasets was compared based on accuracy, recall, RMSE, and computational time. The remaining parts of this paper are organized as follows. Section 2 describes research methods to phishing websites detection, feature selection, and some machine learning models used in this study. Section 3 presents the performance measures used in the study. Section 4 presents and discusses the results obtained. Finally, the outcomes presented in this study are concluded in section 5.

2.Research Method

The goal of this study is to compare four machine learning models, i.e. NBTree, DTree, NBayes and STACKING in terms of the improvement of performance to classify a phishing website dataset, called Feature 31. Another dataset was created, called Feature 20, as a result of the application of the wrapper feature selection method to the original dataset, by selecting the highest contributing features with respect to the class feature. All experiments were performed using Waikato Environment for Knowledge Analysis (WEKA)[1], an open-source tool with numerous machine learning models. The flow of methodology employed for this study is illustrated in Figure-1

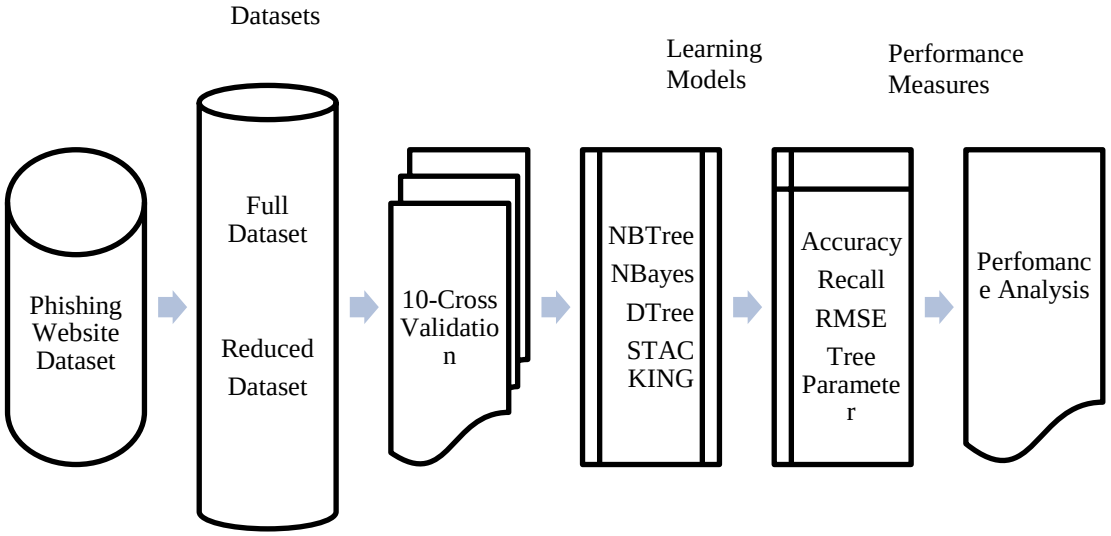

Figure 1-Flow of methodology for the study.

2.1 Dataset

The analyzed phishing website dataset consists of 11055 samples with 31 features and 2 classes. All of the attributes are of the integer data type. The classes contain binary value features of "Phishy" or "Legitimate", translated to -1 and 1, respectively. This dataset was obtained from the archive of Phishtank, MillerSmiles and Googleâ€™s searching operators. [4]. The phishing website dataset was downloaded from the UCI Machine Learning Repository [8]. The steps of analysis and interpretation of data are presented in Table-1.

Table 1-Phishing website dataset description

No	Feature name	Label	Count	Type	No	Featue name	Label	Count	Type
1	having_IP_Address	-1	3793	nominal	17	Submitting_to_email	-1	2014	nominal
		1	7262				1	9041	
2	URL_Length	1	1960	nominal	18	Abnormal_URL	-1	1629	nominal
		0	135				1	9426	
		-1	8960		19	Redirect	0	9776	nominal
3	Shortining_Service	1	9611	nominal			1	1279	
		-1	1444		20	on_mouseover	1	9740	nominal
4	having_At_Symbol	1	9400	nominal			-1	1315	
		-1	1655		21	RightClick	1	10579	nominal
5	double_slash_redir	-1	1429	nominal			-1	476	
		1	9626		22	popUpWidnow	1	8918	nominal
6	Prefix_Suffix	-1	9590	nominal			-1	2137	
		1	1465		23	Iframe	1	10043	nominal
7	having_Sub_Domain	-1	3363	nominal			-1	1012	
		0	3622		24	age_of_domain	-1	5189	nominal
		1	4070				1	5866	
8	SSLfinal_State	-1	3557	nomin	25	DNSRecord	-1	3443	nomin

		1	6331	al			1	7612	al
		0	1167		26	web_traffic	-1	2655	nominal
9	domain_registration_length	-1	7389	nominal			0	2569	
		1	3666				1	5831	
10	Favicon	1	9002	nominal	27	Page_Rank	-1	8201	nominal
		-1	2053				1	2854	
11	port	1	9553	nominal	28	Google_Index	1	9516	nominal
		-1	1502				-1	1539	
12	HTTPS_token	-1	1796	nominal	29	Links_pointing_to_page	1	4351	nominal
		1	9259				0	6156	
13	Request_URL	1	6560	nominal			-1	548	
		-1	4495		30	Statistical_report	-1	1550	nominal
14	URL_of_Anchor	-1	3282	nominal			1	9505	
		0	5337		31	Result	-1	4898	nominal
		1	2436				1	6157	
15	Links_in_tags	1	2650	nominal					
		-1	3956						
		0	4449						
16	SFH	-1	8440	nominal					
		1	1854						
		0	761						

0=suspicious* 1=legitimate** -1=Phishing***

2.2 Model Selection

2.2.1 Feature Selection

Feature selection is a method to select a subset of features from the multi-dimensional dataset, which can improve the classification accuracy in a diversity of datasets. This study proposes the wrapper approach of feature selection. The feature selection algorithm is wrapped round the induction algorithm, which is a black box. This algorithm conducts a search for a good subset as part of the evaluation function. Cross validation is used to estimate the accuracy of the induction algorithm for a set of features [14]. Upon the application of this algorithm to the selected phishing dataset, the number of features was reduced from 31 to 20, including the class label. The reduced dataset created is named Feature 20 while the original dataset with all the 31 features is named Feature 31. In this study, DTree was used as the induction algorithm with 5- folds classification accuracy. A greedy forward search through the space of feature subsets is used. The merit of the best subset found was 0.96.

2.2.2 Naïve Bayes

This model uses the Bayes rule of conditional probability, as well as all the features in the dataset, and analyses them individually on the assumption that they are equal and independent of each other. This model is simple and converges quickly.

Let Y be the class of an instance X. By predicting the class of the instance X by using the Bayes rule, the maximum posterior probability will be found by eqn (1):

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{1}$$

The NBayes classifier assumes that variables $X_1, X_2, \cdots, X_n$ are conditionally independent of each other, given the class, thereby resulting in eqn (2) which is adequate for classification problems:

$$P(Y|X) = P(Y)\prod_{i=1}^{n} P(X_i|Y)P(X) \tag{2}$$

2.2.3 Decision Tree

DTree (C4.5) model uses a top-down greedy approach to rapidly and effectively classify data instances. The trees are generated by recursive partitioning and this stops when all the instances are in the same class value. The algorithm (Figure-2) calculates the information gain for all features

according to their entropy before adding a new node/ feature to the tree. Then, a decision is made based on the highest valued feature. The iteration of the feature selection is continued until all features are considered [15, 16, 17].

Input: S, where S is a set of classified instances
Output: DTree
Requirement: $S \neq \emptyset,\ num\_attributes > 0$

```
Procedure BUILD TREE
    repeat
        maxGain ← 0
        splitA ← null
        e ← Entropy(Attributes)
        for all Attributes a in S do
            gain ← InformationGain(a, e)
            if gain > maxGain then
                maxGain ← gain
                splitA ← a
            end if
        end for
        Partition(S, splitA)
    until all partitions processed
end procedure
```

Figure 2-The DTree Algorithm

2.2.4 Naïve Bayes Tree (NBTree)

This model is a fusion of DTree and NBayes models. DTree nodes contain univariate splits as regular DTree, but the leaves contain Naive-Bayesian models. This approach retains the interpretability of NBayes and DTree, while resulting in models that frequently outperform both components [18]. The algorithm is displayed in Figure-3.

Input: a set S of categorized instances
Output: a decision-tree with NBayes categorizers at the leaves

i. For each attribute X_i, evaluate the utility, $u(X_i)$ of a split on attribute X_i.
ii. Let $j = \arg max_i\ (u_i)$
iii. if u_j is not significantly better than the utility of the current node, create a Naïve-Bayes classifier
for the current node and return
iv. Partition S according to the test on X_j. If X_j is continuous, a threshold split is used; if X_j is discrete,
a multi-way split is made for all possible values.
v. For each child, call the algorithm recursively on the portion of S that matches the test leading to
the child

Figure 3-The NBTree Algorithm

2.2.5 STACKING

STACKING is an ensemble method of a combination of a set of diverse base learners at the initial level, by a meta learner at the higher level. The role of the meta learner is to discover how far it is better to combine the output of the base learners and learn any existing pattern of misclassification by base classifiers [19, 20]. Stacking is a heterogeneous ensemble, as different base learners are composed at the base level. In this study, each run of DTree and NBayes were used as the base learners, whereas one of them (DTree) was the meta learner. The algorithm is summarized by Figure-4.

Algorithm	STACKING
i	Input: Training Data $S = \{x_i, y_i\}_{i=1}^{m}$
ii	Output: Ensemble Classifier H
iii	Phase 1: Learn base classifiers (NBayes, Naïve Bayes Tree, DTree)
iv	**for** $t = 1\ to\ T$ **do**
v	learn h_t based on S
vi	**end for**
vii	Phase 2: Construct new data set of predictions
viii	**for** $i = 1\ to\ m$ **do**
ix	$S_h = \{x_i', y_i\},\quad where\ x_i' = \{h_1(x_i), \cdots, h_T(x_i)\}$
x	**end for**
xi	Phase 3: Learn a meta-classifier (DTree)
xii	learn H based on S_h
xiii	return H

Figure 4-STACKING Algorithm

3. Performance Evaluation Measure

Four metrics were used to evaluate these models on the phishing website data. From the confusion matrix table (Table-2), accuracy and recall can be calculated, as shown in Table 3. Rate is the performance loss or gain on the model upon application of the feature selection method.

Table 2-Confusion matrix for a binary class problem

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 3-The performance evaluation measures for this study

Metric	Formula
Accuracy	$\left(\frac{TP + TN}{TP + FP + FN + TN}\right)\%$
Recall	$\frac{TP}{TP + FN}$
RMSE	$\sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}$ Where N is the number of samples, y_j is actual value and $\hat{y}_j$ is predicted value.
Time	CPU Running Time
Rate	$Performance = \left(\frac{X_1 - X_2}{X_1}\right) \times 100$ Where $X = \{Accuracy, Recall, RMSE, Time\}$

4. Results and Analysis

An empirical experiment was conducted to compare NBTree with NBayes, DTree and STACKING in terms of classification accuracy, recall, computational time and RMSE on the two datasets (Feature 20 and Feature 31). All models were used with WEKA default value setting. DTree was used as base learner for both wrapper feature selection and STACKING, with their default values. The classification accuracy, recall, time and RMSE of all 4 models using 2 different datasets were obtained by 10 runs of 10-fold cross-validation. Therefore, we have 800 (10x10x4x2=800) results generated to be averaged. Finally, we conducted a two-tailed *t*-test with 95% confidence level to compare the models.

Tables 4 and 5 shows the detailed classification accuracy and recall of each model on each data set. The symbols * and ** in the tables, respectively, denote statistically significant degradation or upgradation over NBTree with a 95% confidence level. Besides, the averaged classification accuracies and *w*/*t*/*l* (***w***in/***t***ie/***l***ose) values are summarized at the bottom of the tables. Each entry *w*/*t*/*l* in the table means that NBayes, DTree and STACKING win on *w* data sets, tie on *t* data sets, and lose on *l* data sets, respectively, compared to the hybrid NBTree.

Similarly, their running time was also observed. The running time of each model is the averaged CPU time in seconds. Our experiments are performed on an Intel® core™ i5-7200 CPU @ 2.50GHz 2.70 GHz Pentium Windows computer with 8GB RAM. The detailed experimental results are shown in Tables 4 – 7.

Table 4-Classification accuracy comparisons for NBTree against other models

Datasets	NBTree	NBayes	DTree	STACKING
Features (31)	94.1	92.95*	95.92**	95.97**
Features (20)	95.45	92.73*	96.04**	96.04**
Average	94.775	92.84*	95.98**	96.005**
(w/t /l)		(0/0/2)	(2/0/0)	(2/0/0)
Rate	-1.43	0.24	-0.13	-0.07

Table 5-Recall measure comparisons for NBTree against other models

Datasets	NBTree	NBayes	DTree	STACKING
Features (31)	0.94	0.93*	0.96**	0.96**
Features (20)	0.95	0.93*	0.96**	0.96**
Average	0.945	0.93*	0.96**	0.96**
(w/t /l)		(0/0/2)	(2/0/0)	(2/0/0)
Rate	1.05	0.00	0.00	0.00

From our experiments, it can be observed that DTree and STACKING models produced the best results compared to the other models in terms of classification accuracy and recall as shown in Tables 4 and 5 respectively. Using NBTree as the test base, we summarize some observations briefly, as follows:

DTree significantly outperforms NBTree and NBayes on the two datasets. DTree produced an average accuracy and recall of 95.98% and 0.96 respectively, when compared to NBTree with average accuracy and recall of 94.775% and 0.945 respectively.

STACKING significantly outperforms NBTree and NBayes on the two datasets. STACKING produced average accuracy and recall of 96.005% and 0.96 respectively, when compared to NBtree with average accuracy and recall of 94.775% and 0.945 respectively.

NBTree significantly outperforms NBayes on the two datasets. NBTree produced average accuracy and recall of 94.775% and 0.945 respectively, when compared to NBayes with average accuracy and recall of 92.84% and 0.93 respectively.

NBayes performed worse on the two datasets compared to the other models. The classification accuracy was also degraded when the features were reduced to 20. One of the reasons for this result is that NBayes cannot learn about the interactions and relationships between the features in each instance.

There is improvement in classification accuracy when 20 features were selected for classification for all models, except for NBayes which showed a decreased performance.

For the recall measure, no changes in the performance for all models were observed when the features were reduced to 20, except for NBayes whose performance dropped from 0.95 to 0.94.

The results for classification accuracy and recall indicated that DTree and STACKING are the best for classification of phishing website datasets.

The accuracy rate for all models are below 0%, except for NBayes with 0.24%. Similarly, there is no gain or loss in the recall rate of the models, except for an insignificant loss (1.05%) in the NBTee model upon wrapper feature selection.

For another group of experiments (Tables 6 and 7), we investigate the ranking performance of NBTree in terms of the running time and RMSE. For both measures, a low values were shown to have better results than those of high values, which contradicts the case for accuracy and recall. Thus, ** reflects significantly higher value for the other two methods than that of NBTree, in terms of running time, whereas * represents significantly lower value performance. The results from Tables 6 and 7 can be summarized as follows:

i Table 6 showed that NBTree has the maximum running time with the two datasets.

ii Table 6 showed that NBayes significantly outperforms all the other models in term of the running time.

iii DTree significantly performs better in both the running time and RMSE than STACKING on the two datasets.

iv There is a significant reduction in running time of all models upon feature selection, except NBayes where there is no changes.

Table 6-Running Time measure comparisons for NBTree against other models

Datasets	NBTree	NBayes	DTree	STACKING
Features (31)	18.39	0.00**	0.06*	0.64*
Features (20)	17.95	0.00**	0.04*	0.54*
Average	18.17	0.00**	0.05*	0.59*
(w/t /l)		(2/0/0)	(2/0/0)	(2/0/0)
Rate	2.39	0.00	33.33	15.63

Table 7-RMSE measure comparisons for NBTree against other models

Datasets	NBTree	NBayes	DTree	STACKING
Features (31)	0.21	0.23*	0.18**	0.19**
Features (20)	0.19	0.23*	0.18**	0.19
Average	0.2	0.23	0.18*	0.19
(w/t /l)		(0/0/2)	(2/0/0)	(1/1/0)
Rate	9.52	0.00	0.00	0.00

v As shown in Table 7, DTree significantly outperforms the other models on the two datasets. The averaged RMSE for DTree (0.18) is significantly lower compared to the other models.

vi STACKING significantly outperforms NBTree and NBayes, with all 31 features dataset, but ties with NBTree when 20 features were present. However, the averaged accuracy (0.19) is lower than that for NBtree (0.2).

vii NBTree outperforms NBayes on the two datasets with averaged accuracy of 0.20 as compared to NBayes averaged accuracy of 0.23.

viii NBayes performed significantly lower on the two datasets when compared to the other models.

ix There is an improvement in RMSE when 20 features were selected for classification for all models, except for NBTree whose performance decreased from 0.19 to 0.21.

x Comparing the RMSE performances, DTrees performance (0.18) is significantly higher than that of STACKING (0.19) on the two datasets. However, either could be recommended for the classification of phishing website dataset.

xi The running time rate values for all models are above zero (0%) except for NBayes whose value is 0.00%. There is an insignificant loss in the performance upon the application of the wrapper feature selection method, as the value is small. Thus, there is no significant gain or loss in performance of the model upon feature selection.

xii The RMSE rate values for all models are zero (0%) except for NBTree whose value is 9.52%. There is an insignificant loss in performance upon the application of the wrapper feature selection method, as the value is small. Thus, there is no significant gain or loss in performance of the model upon feature selection.

For the third group of experiments (Table 8), we also investigated the number of leaves and the size of trees created by the two datasets. Table 8 showed that NBTree and DTree produced reduced number of leaves and size of the tree with reduced features. On the other hand, STACKING produced increased number of leaves and size of tree with reduced features. In general, STACKING outperformed all other related models with smaller number of leaves and size of the tree.

Table 8-Tree models comparison in terms of number of leaves and size of trees created by the two datasets

	Feature (20)		Feature (31)	
Model	No of Leaves	Size of Tree	No of Leaves	Size of Tree
NBTree	175	310	236	422
DTree	139	241	169	297
STACKING	15	29	6	11

5.Conclusions

DTree and NBayes are two very simple, efficient, and effective machine learning models for addressing the classification problems. In this study, we presented two models: a hybrid (NBTree) and Ensemble (STACKING) based on DTree and NBayes base learners. These models were built and evaluated independently at the training time, and the class-membership probabilities are weightily averaged according to their classification accuracies, recall, running time and RMSE on training data at the test time. The experimental results on the two phishing website datasets, Feature 20 and Feature 31, showed that DTree performed as good as STACKING in terms of classification accuracy, recall and RMSE, being significantly higher than NBTree in both datasets. Overall, the results showed that the ensemble of DTree and NBayes performed better than their hybrid.

Reference

1. Mohammad, R.M., Thabtah, F. and McCluskey, L. **2015**. "Tutorial and critical analysis of phishing websites methods," *Computer Science Review*, **17**: 1–24.
2. Nidhin, A.U., Harikrishnan, N.B., Vinayakumar, R., Soman, K.P. and Sai, S. **2018**. in 1st AntiPhishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018), Tempe, Arizona, USA.
3. Ali, W. **2017**. "Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection," *International Journal of Advanced Computer Science and Applications* (IJACSA), **8**(9): 72-79.
4. Mohammad, R., McCluskey, T.L. and Thabtah, F.A. **2012**. "An Assessment of Features Related to Phishing Websites using an Automated Technique.," in For Internet Technology And Secured Transactions (ICITST 2012), IEEE, London, UK.
5. Mohammad, R., McCluskey, T.L. and Thabtah, F.A. **2014**. "Intelligent Rule based Phishing Websites Classification," *IET Information Security*, **8**(3): 153-160.
6. Varshney, G., Misra, M. and Atrey, P.K. **2016**. "A survey and classification of web phishing detection schemes," *Security And Communication Networks*, **6**: 6266–6284.
7. Jeeva, S.C. and Rajsingh, E.B. **2016**. "Intelligent phishing url detection using association rule mining," *Human-Centric and Computing Information Science*, **6**: 10.
8. Karabatak, K.M. and Mustafa, T. **2018**. "Performance comparison of classifiers on reduced phishing website dataset," in 6th International Symposium on Digital Forensic and Security (ISDFS).
9. Zhang, W., Ren, H. and Q. Jiang, Q. **2016**. "Application of Feature Engineering For Phishing Detection," *IEICE Transaction Information and System*, **E99-D**(4): 1062-1070.
10. Jain, A.K. and B. B. Gupta, B.B. **2018**. "Towards detection of phishing websites on client-side using machine learning based approach," *Telecommunication Systems*, **68**: 687-700, 2018.

11. Babagoli, M., Aghababa, M.P. and Solouk, V. **2019**. "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft Computing*, **23**: 4315–4327.
12. Frank, R. R. E. Hall, M. A. Holmes, B. Reutemann, P. and Witten, I. A. **2010**. "WEKA — Experiences with a Java Open-Source Project. *Journal of Machine Learning Research*," **11**: 2533-2541, 2010.
13. Bouckaert, R.R. Frank, E., Hall, M.A., Holmes, B., Reutemann, P. and Witten, I.A. **2010**. "WEKA — Experiences with a Java Open-Source Project. Journal of Machine Learning Research," vol. 11, pp. 2533-2541.
14. Kohavi, R. and John, G.H. **1997**. "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324.
15. Quinlan, J.R. **1993**. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., 1993, pp. Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA..
16. Hussein, W.N. and Hussain, H.N. **2019**. "A Design of a Hybrid Algorithm for Optical Character Recognition of Online Hand-Written Arabic Alphabets," *Iraqi Journal of Science*, vol. 60, no. 9, pp. 2067-2079.
17. Abbas, A.R. and Kareem, A.R. **2018**. "Intelligent Age Estimation From Facial Images Using Machine Learning Techniques," *Iraqi Journal of Science*, vol. 59, no. 2(A), pp. 724-732.
18. Kohavi, R. **1996**. "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid," in *Second International Conference on Knoledge Discovery and Data Mining*.
19. Wolpert, D.H. **1992**. "Stacked generalization," *Neural Networks*, vol. 5, no. 2, p. 241–259.
20. Panicker, S. Selot and Sharma, M. **2019**. "Improving Accuracy in Human Age Classification Using Ensemble Learning Techniques," *Iraqi Journal of Science*, vol. 60, no. 8, pp. 1830-1836.