Journal of Engineering Science and Technology
Vol. 17, No. 1 (2022) 0563 - 0582

INTELLIGENT TREE-BASED ENSEMBLE APPROACHES FOR PHISHING WEBSITE DETECTION

YAZAN A. ALSARIERA[1,*], ABDULLATEEF O. BALOGUN[2], VICTOR E. ADEYEMO[3], OMAR H. TARAWNEH[4], HAMMED A. MOJEED[2]

[1]Department of Computer Science, College of Science,
Northern Border University, Arar 73222, Kingdom of Saudi Arabia
[2]Department of Computer Science, University of Ilorin, 240003 Ilorin, Nigeria
[3]School of Built Environment, Engineering and Computing,
Leeds Beckett University, Headingley Campus, Leeds LS6 3QS, United Kingdom
[4]Department of software engineering, Faculty of Computer
Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
*Corresponding Author: yazan.sadeq@nbu.edu.sa

Abstract

The adverse effects of phishing attacks on unsuspecting victims are damaging and nefarious. Stealing of information from unsuspecting users surges on the internet and various solutions have been proposed to curb this menace. Apparently, the evasiveness of phishing attacks through dynamic processes renders these solutions ineffective. To curb this prevalence, machine learning (ML)-based solutions are developed and deployed as it offers continuous learning of phishing dynamics as opposed to explicit or static countermeasures. However, existing ML solutions suffer drawbacks in the case of high false alarm rates and relatively low accuracy values. Hence, this paper proposed novel intelligent tree-based ensemble approaches for phishing website detection. Particularly, ensemble methods (ABELM, BAELM, MABELM) are developed based on Naïve Bayes Tree (NBTree) and Best-First Tree (BFTree) classifiers. NBTree uses an NB classifier at the leaf nodes of a decision tree while BFTree deploys the best first induction method to add the best split in each step of the decision tree. Experimental results showed that the proposed methods are highly effective for phishing website detection outperforming baseline classifiers and ML-based phishing models from recent studies. Consequently, the tree-based ensemble approaches are viable methods that can be used for detecting phishing websites with dynamic traits.

Keywords: BFTree, Ensemble, Machine learning, NBTree, Phishing.

1. Introduction

The continuous development and acceptance of internet technology have made it possible for the migration of most day-to-day human activities into web-based solutions. These human endeavours span from basic activities like education to somewhat necessities such as communication and financial transactions [1, 2]. The aim is to enhance access and utilization of some essential information technology (IT) infrastructures needed daily. Nevertheless, due to the absence of standard internet protocol, the unregulated access and availability of these IT infrastructures create possibilities for internet threats and attacks [3, 4]. These threats and attacks pose substantial risks and concerns for all parties involved, web-based solutions and unwary consumers alike. An example of the consequences of internet attacks includes identity and monetary losses. A typical case of one of these internet attacks is the phishing website.

A phishing website involves the utilization of illegitimate websites and their resources to wrongful acquire sensitive information from end-users. Biometric data, bank account details, and other sensitive information are taken from innocent users. As a result, phishing website attacks represent a considerable danger to web-based solutions [5, 6]. In particular, the Anti-Phishing Working Group (APWG) reported the existence of more than 50,000 phishing websites online in 2018. According to RSA, global organisations incurred more than $9 billion in 2016 due to phishing attempts [7, 8]. These cases have indicated that phishing attacks through illegal websites are rapidly gaining pace, are extremely detrimental to incur, and current countermeasures may be ineffective in tackling the issue [5, 6, 9].

Different anti-phishing techniques have been suggested and designed by internet security specialists and scientists alike to identify or curb phishing websites [10-12]. The blacklist-based phishing attack detection is one such solution. To evaluate the legality of a phishing website, web browsers use a blacklist-based approach that analyses the provided uniform resource locator (URL) to historically registered phishing website URLs. One key problem of the blacklist-based solution is its dependence on accumulated black-listed phishing URLs, which results in their inability to identify new phishing URLs [13, 14]. Additionally, cyber-attackers use adaptive tactics to easily overcome the blacklist approach.

In order to cope with the evolving complexities of phishing websites, machine learning (ML)-based approaches are used to analyse features retrieved from websites to evaluate their legitimacy. The goal is to improve resiliency in detecting new phishing websites [10-12, 15]. Nonetheless, the efficiency of an ML-based phishing detection model is contingent on the selected ML algorithm's efficacy. Many ML systems for detecting phishing websites have been developed and presented, with generally poor detection accuracy and high false-positive rates [16, 17]. This trait can be linked to data quality problems like high dimensionality and class imbalance, which have an inimical effect (negative) on the efficiency of ML algorithms [18-20]. Additionally, the dynamic of phishing websites necessitates the development of more effective methods with strong phishing website detection rates and reduced false-positive rates (FPR). Therefore, this research proposes intelligent tree-based ensemble techniques for detecting phishing websites. Ensemble variations of hybrid Naive Bayes Tree (NBTree) and Best-First search Tree (BFTree) are used to identify phishing websites. NBTree involves the deployment of Naïve Baye's (NB) model at the decision tree's leaf nodes while the BFTree uses the best first induction method to add the best split at each level of the

decision tree (DT). The ensemble variants of both methods (i.e., NBTree and BFTree) are based on the amplification of their respective detection performances.

This research is organised as follows: Section 2 elucidates the related studies with a succinct summary of previous observations. Section 3 elaborates on the proposed approaches in-depth, together with the research framework. Section 4 presents the experimental findings using various evaluation metrics and approaches, and Section 5 provides a comprehensive conclusion and recommendations.

2. Literature Review

To identify phishing websites, Mohammad et al. [21] proposed an enhanced neural network (NN) established on an adjustable parameter that alters the learning rate before adding extra neurons and, eventually, the network topology. Upon validation, the proposed NN method recorded a detection accuracy of 91.12%. In their investigation, Verma and Das [22] used a Deep Belief Network (DBN) to identify phishing websites. The DBN model is primarily based on Restricted Boltzmann machines (RBM), which generates deep hierarchical representations from the studied dataset. The suggested DBN recorded a 94.43% detection accuracy that outperformed other experimented baseline classifiers. In another similar study, Ali and Ahmed [9] employed attributes culled by a genetic algorithm (GA) on a deep neural network (DNN). As observed in the study, the suggested technique (GA-DNN) outperformed experimented prominent baseline classifiers. Also, Vrbančič et al. [23] successfully enhanced integrated DNN with a metaheuristics method (Bat algorithm). The suggested approach achieved a maximum accuracy of 96.9%. These results indicate that NN models can recognise phishing websites just as well as baseline classifiers. However, the unexplained functionalities of NN variations such as DBN and DNN are a significant disadvantage. Furthermore, the performance of NN versions is largely determined by the design of the hardware utilised for their implementation. As a result, there is a lack of generalizability.

Alqahtani [24] detected phishing websites using a novel association rule induction method. The proposed approach leverages an association rule mechanism to determine the authenticity of a website. Their experimental findings demonstrated the efficacy of the suggested technique, as it beats (Accuracy: 95.20%, F-measure:0.9511) baseline classifiers such as DT, RIPPER, and various associative learning classification models. Dedakia and Mistry [17] suggested a content-based Associative Classification (CBAC) technique for detecting phishing. By taking into account content-based characteristics, the suggested technique enhances the Multi-Label Class Associative Classification (MCAC) algorithm. Similar work was conducted by Abdelhamid et al. [25]. The suggested approach (CBAC) has an accuracy value of 94.29% based on the test data. Hadi et al. [26] created and tested a fast AC algorithm (FACA) against other current associative classification (AC) techniques (CBA, CMAR, MCAR, and ECAR) for phishing website detection. Based on the accuracy and F-measure values, their experimental findings suggest that FACA outperforms other AC approaches. The usefulness of these associative-based techniques for phishing detection is shown by their effectiveness. Its poor detection accuracy, however, is a hindrance.

Rahman et al. [27] investigated the effectiveness of various ML algorithms and ensemble methods in identifying website phishing. Likewise, Chandra and Jana [16] investigated the deployment of meta-classifiers to improve phishing website detection. Findings from their respective studies indicated the superiority of

ensemble techniques over individual classifiers. Alsariera et al. [6] created ensemble variations of Forest Penalizing by Attributes (ForestPA) for detecting phishing websites. Their testing findings showed that the suggested ForestPA meta-learners' versions are extremely successful in identifying phishing websites, with a minimum accuracy of 96.26%.

To choose optimum features, Chiew et al. [7] introduced a hybrid ensemble FS (HEFS) technique based on a unique cumulative distribution function gradient (CDF-g) method. The accuracy of the RF assessment of HEFS was 94.6%. Correspondingly, Aydin and Baykal [28] employed subset-based attributes collected from a website URL to identify phishing. Following that, the duo of NB and Sequential Minimal Optimization (SMO) was deployed on the collected features. The deployed models recorded 83.89% (NB) and 95.39 (SMO) detection accuracy accordingly. Ubing et al. [29] presented a phishing approach based on feature selection (FS) and the ensemble method. The Random Forest Regressor (RFG) was utilised as the FS technique, and majority voting was employed as the ensemble method. The proposed framework recorded a significant phishing detection performance, however, the choice of implemented FS method is not generalizable. And the resulting detection accuracy value can be improved.

In their work, Aziz et al. [30] evaluated the performance of NBTree and BFTree classifiers with other approaches such as NB, DT, MLP, and RF for intrusion detection. Their findings demonstrated that NBTree and BFTree may be utilised efficiently for anomalous intrusion detection and categorization. Also, Guzmán et al. [31] used NBTree and BFTree to detect aberrant heart rates in patients with long-term cardiovascular disorders. Furthermore, their experimental findings demonstrated the advantages of the NBTree and BFTree pair for categorization procedures. However, utilising ensemble approaches, the performance of these methods (NBTree and BFTree) may be improved [2, 32].

Prior research indicates that there is a need for more effective and efficient solutions, as most current options perform poorly. As a consequence of this, this research presents intelligent tree-based ensemble algorithms for identifying phishing websites.

3. Methodology

This section describes the research approach that was used in this study. The suggested solutions, the investigated phishing datasets, performance evaluation metrics, and experimental methodology were specifically explored.

3.1. Naïve Bayes Tree (NBTree)

Naïve Bayes Tree (NBTree) is the combination of NB and DT algorithms. It is very close to classical recursive partitioning. NBTree deploys an NB classifier at the leaf node of the generated DT [33]. As with DT algorithms, NBTree uses the entropy minimization method to select attributes. The utility of each node is generated by discretizing and estimating the data using the NB classifier. The utility of a split is the weighted sum of the utility, where the weight assigned to a node is proportionate to the number of instances that traverse that node [34]. NBTree has been used and reported in various research domains to yield high classification accuracy values [34-36].

3.2. Best First Tree (NBTree)

Best First Tree (BFTree) involves the deployment of the best first induction method to add the best split in each step of a decision tree. BFTree generates binary trees intending to utilize the intra-node homogeneity characteristics [37]. BFTree selects the best node to split at every step by first sorting in a descending order based on the entropy properties (Gini index or information gain). Iteratively, the best node which is produced at first is the node whose split leads to the highest reduction of impurity amongst the nodes available for splitting. This will, in the end, alter the ordering and structure of the ensuing DT [34, 38]. Table 1 presents the parameter settings of NBTree and BFTree as used in this study.

Table 1. Classification algorithm.

Classification Algorithm	Parameter Setting
Naïve Bayes Tree (NBTree)	BatchSize=100; UseKernelEstimator=False; UseSupervisedDiscretization=True; BinarySplit=True; CollapseTree=True;ConfidenceFactor=0.25; minNumObj=2; UseMDLCorrection=True; Unpruned=False; NumFold=3
Best First Tree (BFTree)	BatchSize=100; heuristic=True; minNumObj=2; numFold=5; PruningStrategy=Post-Pruning; UseGini=True; UseErrorRate=True;

3.3. Proposed tree-based ensemble approaches

This section presents and discusses the proposed tree-based ensemble approaches as used in this study. Specifically, the working principles of the bootstrap aggregation ensemble learning method (BAELM) and Adaptive Boost Ensemble Learning Method (ABELM) are discussed.

a. Bootstrap Aggregation Ensemble Learning Method (BAELM)

The bootstrap aggregation ensemble learning method (BAELM) is a homogeneous ensemble strategy for improving prediction model performance [39, 40]. NBTree and BFTree are used as foundation classifiers for BAELM in this technique. It creates models by training them on N instances from the phishing dataset. The phishing dataset is specifically resampled (with replacement) into N subsets, and each subset is trained using BAELM with NBTree and BFTree, respectively. At prediction time, the output models of the corresponding base classifiers (NBTree and BFTree) are aggregated from N subsets. Algorithm 1 [15, 39, 40] shows the pseudocode for BAELM with NBTree and BFTree as base classifiers.

Algorithm 1. Bootstrap Aggregation Ensemble Learning Method

Input: Training set S, Base classifier I ={NBTree, BFTree },
integer T= 100 (number of bootstrap samples).

1. for i = 1 to T {
2. S' = bootstrap sample from S (sample with replacement)
3. $C_i = I(S')$
4. }
5. $C^*(x) = \arg\max \sum_{i:C_i(x)=y} 1$ (the most frequently predicted label y)

Output: classifier C^*

b. Adaptive Boost Ensemble Learning Method (ABELM)

Adaptive Boost Ensemble Learning Method (ABELM) repeatedly trains weighted phishing datasets using N base classifiers (NBTree and BFTree) in succession [41, 42]. ABELM employs weighted averages to improve the prediction performance of each base classifier, with each model determining which features to focus on in the following iteration. A majority vote technique is used at the end for the final decision. This way, every model developed by the base classifier is considered in selecting the final superior model [43]. Algorithm 2 details the ABELM algorithm [41, 42].

Algorithm 2. Adaptive Boost Ensemble Learning Method

Input: Training set $S = \{x_i, y_i\}, i = 1 \ldots m, y_i \in Y, Y = \{c_1, c_2\}$, c_k is the class label; The number of Iterations T=100;
Base classifier I ={NBTree, BFTree };
1 Initializing weights distribution of $D_1(i) = 1/m$
2 For t = 1 to T
3 Train classifier $I(S, D_t)$, get a classifier $h_t = X \rightarrow \{c_1, c_2, \ldots, c_k\}$
4 Compute the error rate of h_t, $\varepsilon_t \leftarrow \sum_{i=1}^{m} D_t(i)[y_i \neq h_t(x_i)]$
5 If $\varepsilon_t > 0.5$ then
6 $T \leftarrow t - 1$
7 Continue
8 End if
9 Set $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
10 For i = 1 to m
11 Update weight $D_{t+1})i) = D_t(i)\beta_t^{1-[y_i \neq h_t(x_i)]}$
12 End for i
13 End for t
Output: the final classifier

$$H(x) = \arg\max\left(\sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right)[Y \neq h_t(X)]\right)$$

c. Multi-boost Adaptive Boost Ensemble Learning Method (MABELM)

Multi-boost Adaptive Boost Ensemble Learning Method (MABELM) is regarded as advanced ABELM forming decision committees. Specifically, MABELM can be said to consist of the combination of MABELM and the wagging method. This gives MABELM the advantage of deploying and building a model with low bias and variance [44]. MABELM provides a computational edge of working with parallel execution. However, the parallelization of MABELM will warrant a change in the process of termination of training a subcommittee [45]. As with ABELM, the MABELM is used for the amplification of the prediction performance of the base classifiers (NBTree and BFtree) for phishing website detection. Algorithm 3 [44, 45]. presents the pseudocode for MABELM. Table 2 shows the parameter settings of the ensemble methods used in this study.

Table 2. Ensemble methods.

Ensembles	Parameter Setting
BAELM	Classifier={NBTree, BFTree} , bagSizePercent=100; numIteration=100; seed=1; calcOutOfBag=False; batchSize=100
ABELM	Classifier={ NBTree, BFTree }, weightThreshold=100; numIteration=100; seed=1; useResampling=True; batchSize=100
MABELM	Classifier={NBTree, BFTree}, BatchSize=100; numIterations=100; numSubCometys=3; UseResampling=True; weightedThreshold=100

Algorithm 3. Multi-boost Adaptive Ensemble Learning Method

Input: Training set $S = \{x_i, y_i\}, i = 1 \ldots m, y_i \in Y, Y = \{c_1, c_2\}$, $c_k$is the class label; The number of Iterations T=100;
The number of Subcommittee Iterations Ji=10, i≥1
Base classifier I ={NBTree, BFTree };
1 Initializing weights distribution of $K(i) = 1$
2 for t = 1 to T
3 if Jk=t then
3 S' is set to random weights based on continuous Poisson Distribution.
4 Standardize S' to sum to n.
5 ++k.
6 Train classifier $I(S, D_t)$, get a classifier $h_t = X \rightarrow \{c_1, c_2\}$
7 Compute the weighted error rate of $h_t, \varepsilon_t \leftarrow \frac{\sum_{i=1}^{m} D_t(i)[y_i \neq h_t(x_i)] \cdot x}{m}$
9 if $\varepsilon_t > 0.5$ then
10 S' is set to random weights based on continuous Poisson Distribution.
11 standardize S' to sum to n.
12 ++k.
13 go to Step 7
14 else if $\varepsilon_t = 0$ then
15 set $\beta_t = 10\text{-}10$
16 S' is set to random weights based on continuous Poisson Distribution.
17 standardize S' to sum to n.
18 ++k.
19 else
20 set $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
21 for i = 1 to m
22 if $[Y \neq h_t(X)]$
23 weight (Xi)/$2\varepsilon_t$
24 else
25 weight (Xi)/$2(1 - \varepsilon_t)$
26 if weight(Xi) < 10^{-8}
27 weight(Xi) = 10^{-8}
28 End for i
29 End for t
Output: the final classifier

$$H(x) = \arg\max\left(\sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right)[Y \neq h_t(X)]\right)$$

3.4. Experimental framework

This section discusses the experimental framework employed in this investigation, as shown in Fig. 1. The experimental framework is aimed at empirically analysing and verifying the efficiency of the suggested solutions for phishing websites detection. Phishing websites datasets from the UCI repos are used to train and test the proposed approaches, and the K-fold (k=10) cross-validation methodology is used to develop and evaluate the phishing models. The preference of 10-fold CV is due to its capacity to develop ML models with a modest effect of the class imbalance issue [46-50]. Furthermore, the K-fold CV approach ensures that each instance is utilised repeatedly for both training and testing [51-54].

The suggested tree-based ensemble techniques and selected classifiers (NB, SMO, SVM, DT, and Decision Table (DTable)) are trained using the phishing website datasets. As a result, the phishing website detection performances of the proposed phishing models is assessed and related to experimented and existing phishing approaches. All tests were carried out in the same setting, utilising the WEKA machine learning programme [55].

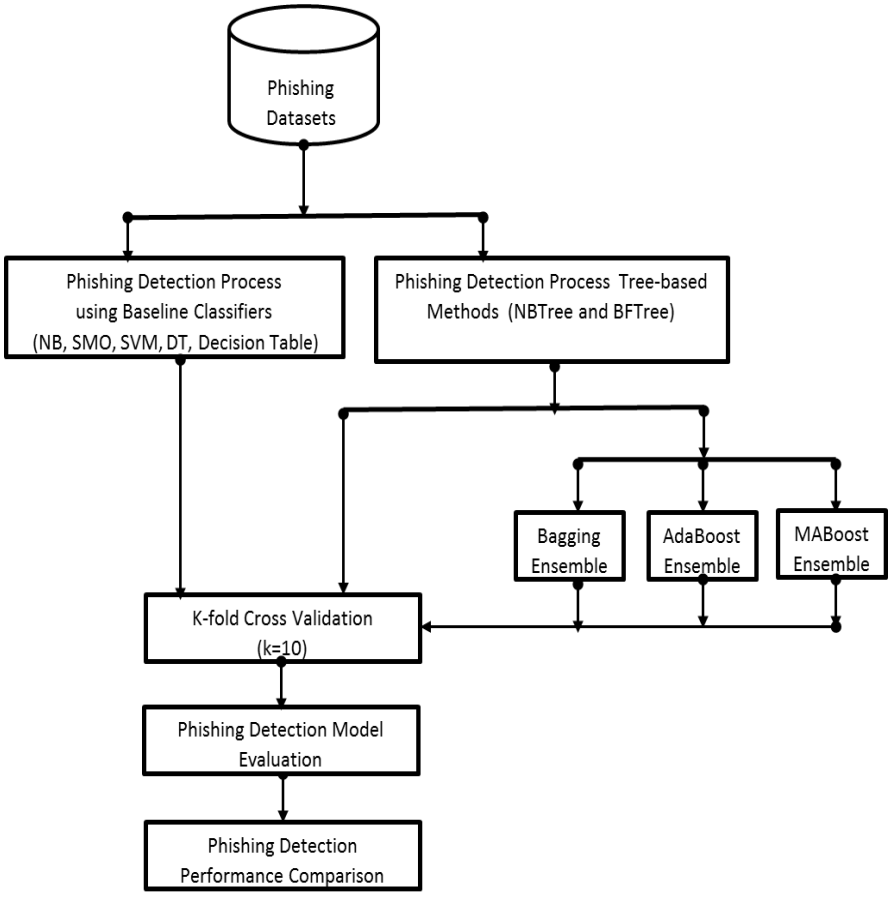

Fig. 1. Experimental framework.

3.5. Phishing datasets

Three phishing datasets were employed in the experimental procedure in this work. These phishing datasets are freely accessible and have been utilised extensively in past research [6, 7, 21, 27]. Dataset A contains 11,055 variables, 4,898 of which are phishing and 6,157 of which are genuine. Dataset A has 30 distinct characteristics that define the dataset [21]. Dataset B has 10,000 variables, 5,000 of which are phishing and 5,000 of which are genuine. Also, Dataset B has 48 characteristics with discrete, continuous, and categorical values [27]. Dataset C comprises 1,353 variables with 10 attributes (702 phishing, 548 genuine, and 103 suspicious). Dataset C differs from Datasets A and B in that it contains three class labels. [6, 7, 21, 27] provide further information about the phishing datasets.

3.6. Performance evaluation metrics

Accuracy, F-measure, Area under the Curve (AUC) and false-positive rate (FPR) performance measures are used to analyse the detection performance of the generated phishing models. The selection of these metrics is based on current research that demonstrates broad and consistent usage of these evaluation criteria for phishing website detection [1, 3, 5, 6, 27, 29].

4. Results and Discussion

Based on the experimental framework (See Fig. 1) used in this study, this section highlights and analyses the experimental results.

Tables 3 to 5 compare the phishing detection performance of the suggested methods with experimented baseline classifiers using selected metrics (See Section 3.6) on Dataset A, Dataset B and Dataset C respectively. The baseline classifiers used (See Section 3.4) were chosen for their computing capability and use in phishing website detection [3, 11, 56].

Table 3. Phishing detection of proposed methods with experimented base classifiers on Dataset A.

Models	Accuracy (%)	F-Measure	AUC	FPR
***BaggedBFTree**	96.54	0.965	0.993	0.038
***BaggedNBTree**	95.84	0.958	0.993	0.044
***BoostedBFTree**	**97.09**	0.971	**0.996**	0.032
***BoostedNBTree**	96.94	0.969	**0.996**	0.033
***MultiBoostBFTree**	97.07	0.971	**0.996**	**0.031**
***MultiBoostNBTree**	96.68	0.967	0.995	0.036
BFTree	95.69	**0.977**	0.978	0.046
NBTree	94.52	0.945	0.986	0.058
NB	92.98	0.930	0.980	0.076
DT	96.50	0.966	0.967	0.036
SVM	94.50	0.945	0.943	0.060
SMO	93.80	0.938	0.936	0.066
DTable	93.24	0.932	0.979	0.075

Table 4. Phishing detection of proposed methods with experimented base classifiers on Dataset B.

Models	Accuracy (%)	F-Measure	AUC	FPR
***BaggedBFTree**	97.59	0.976	0.994	0.024
***BaggedNBTree**	98.22	0.982	**0.998**	0.018
***BoostedBFTree**	98.37	**0.984**	0.997	**0.016**
***BoostedNBTree**	**98.38**	**0.984**	**0.998**	**0.016**
***MultiBoostBFTree**	98.00	0.980	0.997	0.020
***MultiBoostNBTree**	98.15	0.981	0.997	0.019
BFTree	97.02	0.977	0.977	0.030
NBTree	96.56	0.966	0.988	0.034
NB	85.15	0.850	0.949	0.0149
DT	97.13	0.971	0.975	0.027
SVM	91.49	0.915	0.915	0.085
SMO	93.87	0.939	0.939	0.061
DTable	95.79	0.958	0.982	0.042

Table 5. Phishing detection of proposed methods with experimented base classifiers on Dataset C.

Models	Accuracy (%)	F-Measure	AUC	FPR
***BaggedBFTree**	90.10	0.901	0.966	0.074
***BaggedNBTree**	**90.61**	**0.906**	**0.975**	**0.071**
***BoostedBFTree**	88.17	0.882	0.963	0.084
***BoostedNBTree**	89.06	0.891	0.962	0.084
***MultiBoostBFTree**	89.98	0.896	0.968	0.079
***MultiBoostNBTree**	89.95	0.899	0.967	0.076
BFTree	89.87	0.899	0.941	0.074
NBTree	89.06	0.891	0.961	0.087
NB	84.10	0.825	0.948	0.120
DT	87.58	0.891	0.916	0.082
SVM	85.66	0.825	0.867	0.123
SMO	86.00	0.846	0.900	0.109
DTable	84.47	0.839	0.954	0.110

As presented in Table 3, amongst the base classifiers, DT (96.5%) recorded the best detection accuracy value followed by BFTree (95.69% and NBTree (94.52%). NB (92.98%) had the least accuracy value amongst the baseline classifiers. However, concerning f-measure and AUC values, BFTree was superior to all other baseline methods. Also, NBTree (0.945, 0.986) recorded a high f-measure and AUC values which were better than other methods except for DT and BFTree. On Dataset B (See Table 4), similar detection performance was observed for the baseline classifiers. DT (97.13%) had the best detection accuracy value followed by BFTree (97.02%) and NBTree (96.56%) respectively. Also on Dataset B, BFTree (0.977, 0.977) had the highest f-measure and AUC values respectively when compared with other baseline classifiers except for the AUC values of NBTree (0.988) and DTab (0.982). However, from Table 5, the duo of BFTree and NBTree outperformed other baseline classifiers on all evaluation metrics. Specifically, BFTree and NBTree had accuracy values of 89.87% and 89.06% respectively which were better than the nearest classifier DT (87.98%). These experimental results showed that tree-based methods such as BFTree and NBTree can produce a competitive performance as some of the baseline classifiers. Most especially in the case of NB and DT, on all studied datasets, BFTree and NBTree were superior in performance to NB but the tree-based methods were competitive

with DT classifier. This finding supports the use of a tree-based approach for phishing website detection as used in this study. However, the detection performances of BFTree and NBTree methods can be enhanced by combining the respective methods with ensemble techniques.

As shown in Tables 3-5, the detection performances of the proposed tree-based ensemble methods (BaggedBFTree, BaggedNBTree, BoostedBFTree, BoostedNBTree, MultiBoostBFTree, and MultiBoostNBTree)were superior to the baseline classifiers. Specifically, on Dataset A (See Table 3), the proposed methods had better accuracy values than the baseline classifiers except in the case of BaggedNBTree which had a lower accuracy value when compared with DT. However, BaggedBFTree (+0.88%), BoostedBFTree (+1.46%) and MultiBoostBFTree (+1.44%) had increments in accuracy values over BFTree while BaggedNBTree (+1.4%), BoostedNBTree (+2.56%) and MultiBoostNBTree (+2.29%) had increments in accuracy values over NBTree. Concerning f-measure values, BFTree had superior values to the proposed methods. However, the AUC values of the proposed methods were better than those of BFTree and NBTree. On Dataset B (See Table 4), BaggedBFTree (+0.6%), BoostedBFTree (+1.39%) and MultiBoostBFTree (+1%) had increments in accuracy values over BFTree while BaggedNBTree (+1.72%), BoostedNBTree (+1.88%) and MultiBoostNBTree (+1.65%) had increments in accuracy values over NBTree. A similar occurrence was observed on Dataset C (See Table 5) except for BoostedBFTree (-1.89%) which had a lower accuracy value to BFTree while BaggedNBTree (+1.74%) and MultiBoostNBTree (+0.99%) had increments in accuracy values over NBTree. BoostedNBTree recorded the same accuracy value as NBTree. Nonetheless, on average, the proposed methods were superior to the baseline classifiers which as a result of the amplification of the detection performances of the tree-based methods (BFTree and NBTree) by BAELM, ABELM, and MABELM as developed in this study. These findings are congruent with reports made about the use of ensemble methods in other research areas [46, 47, 57-60].

Furthermore, considering the respective detection performances of the proposed methods (BaggedBFTree, BaggedNBTree, BoostedBFTree. BoostedNBTree, MultiBoostBFTree, and MultiBoostNBTree) on Dataset A, Dataset B and Dataset C respectively, the proposed methods achieved high phishing website detection efficacy. Notably, from the experimental results on Dataset A and Dataset B, the proposed methods almost obtained perfect AUC values. This signifies that the respective abilities of the proposed methods in detecting phishing websites are substantial and are not subjective to randomness. Also, it was observed that phishing models based on BAELM were less superior to other proposed methods with ABELM, and MABELM on Dataset A and B. This may be attributed to the inefficiency of BAELM in dealing with high-dimensional features, as shown in Datasets A and B [33, 44].

Forthwith, the high AUC, low FPR and high AUC values of the proposed methods on the studied datasets as presented in Fig. 2, Fig. 3 and Fig. 4 respectively, show that the proposed methods can handle latent class imbalance that is present in the studied datasets than the experimented baseline classifiers. In other words, with the existence of data quality constraints in phishing datasets, the proposed methods outperformed BFTree and NBTree in phishing website detection.

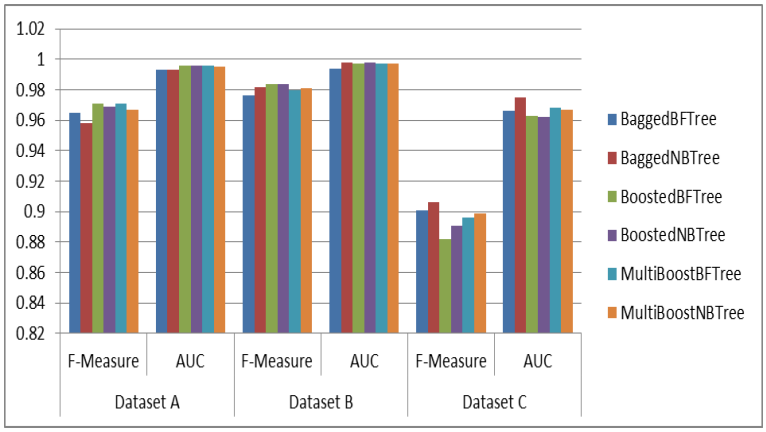

Fig. 2. Graphical illustration of the performance of the proposed methods using F-measure and AUC values.

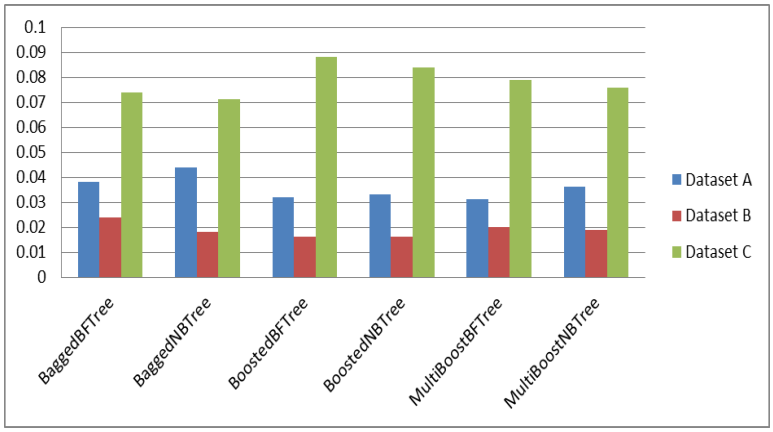

Fig. 3. Graphical illustration of the performance of the proposed methods using FPR value.

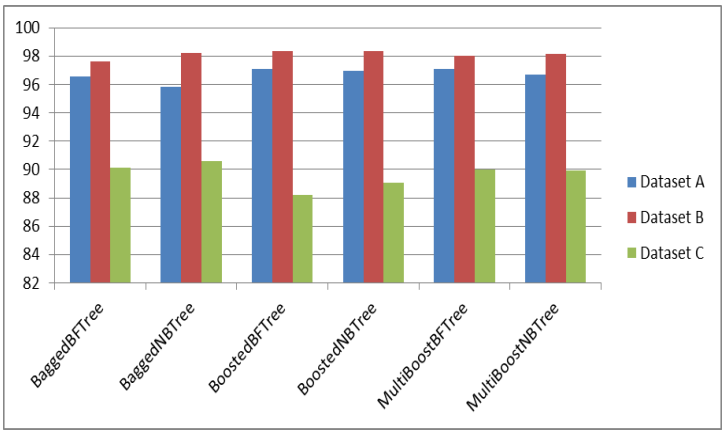

Fig. 4. Graphical illustration of the performance of the proposed methods using accuracy value.

Table 5. Phishing detection of proposed methods with existing methods on Dataset A.

Phishing Methods	Accuracy (%)	F-Measure	AUC	FPR
Aydin and Baykal [28]	95.39	0.938	0.936	0.046
Dedakia and Mistry [17]	94.29	-	-	-
Mohammad et al. [21]	92.18	-	-	-
Ubing et al. [29]	95.40	0.947	-	0.041
Ali and Ahmed [9]	91.13	-	-	-
Verma and Das [22]	94.43	-	-	-
Hadi et al. [26]	92.40	-	-	-
Chiew et al. [7]	93.22	-	-	-
Rahman et al. [27] **(KNN)**	94.00	-	-	0.049
Rahman et al. [27] **(SVM)**	95.00	-	-	0.039
Chandra and Jana [16]	92.72	-	-	-
Al-Ahmadi and Lasloum [56]	96.65	0.965	-	-
Alsariera et al. [6]	96.26	-	-	0.04
Ali and Malebary [61]	96.43	-	-	-
***BaggedBFTree**	96.54	0.965	0.993	0.038
***BaggedNBTree**	95.84	0.958	0.993	0.044
***BoostedBFTree**	**97.09**	**0.971**	**0.996**	0.032
***BoostedNBTree**	96.94	0.969	**0.996**	0.033
***MultiBoostBFTree**	97.07	**0.971**	**0.996**	**0.031**
***MultiBoostNBTree**	96.68	0.967	0.995	0.036

For further validation, the proposed methods are benchmarked with recent existing phishing detection techniques. The comparison was done on each of the studied phishing datasets (See Section 3.4). Table 6 compares the experimental results of the proposed methods and recent phishing website detection models on Dataset A. For clarification, the existing phishing methods are grouped based on their underlining computation technique and were tested using Dataset A in their respective studies. Thus, this makes the comparison appropriate and justified. The proposed methods were superior to NN-based phishing website detection methods as proposed by Mohammad et al. [21], Verma and Das [22], Vrbančič et al. [23], Al-Ahmadi and Lasloum [56] and Ali and Ahmed [9]. Also, Associative classification (AC)-based phishing models developed by Dedakia and Mistry [17] and Hadi et al. [26] and ensemble-based phishing methods by Rahman et al. [27], Chandra and Jana [16], Chiew et al. [7], Ubing er al. [29], and Alsariera et al. [6] were outperformed by the proposed methods. Besides, the proposed methods were better than FS-based solutions by Aydin and Baykal [28] and Ali and Malebary [61].

As shown in Tables 6 and 7, the proposed methods were superior to phishing models by Chiew, Tan [7] and Rahman et al. [27] on phishing accuracy values. More benchmark comparison was done on Dataset A than other studied datasets

(Dataset B and Dataset C) due to its wide usage by most existing studies. On a variety of performance metrics, the proposed methods were superior in phishing website detection performance than available phishing website detection techniques. In conclusion, it is noticeable that the proposed methods are more effective than existing solutions in detecting phishing websites.

Table 6. Phishing detection of proposed methods with existing methods on Dataset B.

Models	Accuracy (%)	F-Measure	AUC	FPR
Chiew, Tan [7]	94.60	-	-	-
Rahman et al. [27] (KNN)	87.00	-	-	0.078
Rahman et al. [27] (SVM)	91.00	-	-	0.067
***BaggedBFTree**	97.59	0.976	0.994	0.024
***BaggedNBTree**	98.22	0.982	0.998	0.018
***BoostedBFTree**	98.37	**0.984**	0.997	**0.016**
***BoostedNBTree**	**98.38**	**0.984**	**0.998**	**0.016**
***MultiBoostBFTree**	98.00	0.98	0.997	0.020
***MultiBoostNBTree**	98.15	0.981	0.997	0.019

Table 7. Phishing detection of proposed methods with existing methods on Dataset C.

Models	Accuracy (%)	F-Measure	AUC	FPR
Rahman et al. [27] **(KNN)**	88.00	-	-	0.099
Rahman et al. [27] **(SVM)**	87.00	-	-	0.087
***BaggedBFTree**	90.10	0.901	0.966	0.074
***BaggedNBTree**	**90.61**	**0.906**	**0.975**	**0.071**
***BoostedBFTree**	88.17	0.882	0.963	0.088
***BoostedNBTree**	89.06	0.891	0.962	0.084
***MultiBoostBFTree**	89.98	0.896	0.968	0.079
***MultiBoostNBTree**	89.95	0.899	0.967	0.076

5. Conclusion

This study proposes intelligent tree-based ensemble approaches for phishing website detection. BFTree and NBTree were augmented with ensemble methods for efficient phishing website detection models. Experimental findings indicated the efficiency of the proposed methods in identifying phishing websites with high phishing detection accuracy and low FPR values. The ensemble methods positively enhance the detection performances of BFTree and NBTree confirming their validation for phishing website detection. Also, the proposed methods were superior to existing phishing models which further confirm their applicability for phishing website detection. Hence, the proposed intelligent tree-based ensemble approaches are recommended for phishing website detection. Due to the dynamic nature of phishing website tactics, we intend to integrate efficient FS methods with tree-based ensemble models in the future.

Nomenclatures

N	Number of Folds
S	Training Set
T	Number of Iterations

Abbreviations

ABELM	Adaptive Boost Ensemble Learning Method
AC	Associative Classification
APWG	Anti-Phishing Working Group
AUC	Area Under Curve
BAELM	Bootstrap Aggregation Ensemble Learning Method
BFTree	Best First Tree
BP	Back Propagation
CBA	Context-Based Associative Classification
CDF-g	Cumulative Distribution Function-gradient
CV	Cross-Validation
DBN	Deep Belief Network
DNN	Deep Neural Network
DT	Decision Tree
ELM	Ensemble Learning Method
ERT	Extreme Randomized Tree
FACA	Fast Associative Classification Algorithm
ForestPA	Forest Tree with Penalizing Attributes
FPR	False Positive Rate
FS	Feature Selection
GB	Gradient Boosting Tree
HEFS	Hybrid Ensemble Feature Selection
KNN	K Nearest Neighbour
LR	Logistic Regression
MABELM	Multi-boost Adaptive Boost Ensemble Learning Method
MCAC	Multi-label Classifier based Associative Classification
ML	Machine Learning
MLP	Multi-Layer Perceptron
NB	Naïve Bayes
NBTree	Naïve Bayes Tree
RBM	Restricted Boltzmann Machines
RFG	Random Forest reGressor
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
ROC	Receiver Operation Characteristics
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
WEKA	Waikato Environment for Knowledge Analysis

References

1. Adewole, K.S.; Akintola, A.G.; Salihu, S.A.; Faruk, N.; and Jimoh, R.G. (2019). *Emerging technologies in computing*. Chapter: Hybrid rule-based model for phishing URLs detection. Switzerland: Springer, 119-135.

2. Elijah, A.V.; Abdullah, A.; JhanJhi, N.; Supramaniam, M.; and Abdullateef, B.O. (2019). Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: an empirical study. *International Journal of Advanced Computer Science and Applications*, 10(9), 520-528.
3. Abdulrahaman, M.D., Alhassan, J.K., Adebayo, O.S., Ojeniyi, J.A., and Olalere, M. (2019). Phishing attack detection based on random forest with wrapper feature selection method. *International Journal of Information Processing and Communication*, 7(2), 209-224.
4. Adil, M.; Khan, R.; and Ghani, M.A.N.U. (2020). Preventive techniques of phishing attacks in networks. *Proceedings of the* 3*rd International Conference on Advancements in Computational Sciences* (*ICACS*). Lahore, Pakistan, 1-8.
5. Aleroud, A.; and Karabatis, G. (2020). Bypassing detection of url-based phishing attacks using generative adversarial deep neural networks. *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*. New Orleans, USA, 53-60.
6. Alsariera, Y.A.; Elijah, A.V.; and Balogun, A.O. (2020). Phishing website detection: forest by penalizing attributes algorithm and its enhanced variations. *Arabian Journal for Science and Engineering*, 45, 10459-10470.
7. Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.C.; and Tiong, W.K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484, 153-166.
8. Tan, C.L.; Chiew, K.L.; Yong, K.S.C.; Sze, S.N.; Abdullah, J.; and Sebastian, Y. (2020). A graph-theoretic approach for the detection of phishing webpages. *Computers and Security*, 95.
9. Ali, W.; and Ahmed, A.A. (2019). Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting. *IET Information Security*, 13(6), 659-669.
10. Yang, P.; Zhao, G.; and Zeng, P. (2019). Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access*, 7, 15196-15209.
11. Zamir, A.; Khan, H.U.; Iqbal, T.; Yousaf, N.; Aslam, F.; Anjum, A.; and Hamdani, M. (2020). Phishing web site detection using diverse machine learning algorithms. *The Electronic Library*, 38(1).
12. 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.
13. Gupta, B.B.; Arachchilage, N.A.G.; and Psannis, K.E. (2018). Defending against phishing attacks: taxonomy of methods, current issues and future directions. *Telecommunication Systems*, 67(2), 247-267.
14. Ghafir, I.; and Prenosil, V. (2015). Blacklist-based malicious IP traffic detection. *Proceedings of the* 2015 *Global Conference on Communication Technologies*. Thuckalay, India, 229-233.
15. Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O.; and Alazzawi, A.K. (2020). AI meta-learners and extra-trees algorithm for the detection of phishing websites. *IEEE Access*, 8, 142532-142542.
16. Chandra, Y.; and Jana, A. (2019). Improvement in phishing websites detection using meta classifiers. *Proceedings of the* 6*th International Conference on*

Computing for Sustainable Global Development (*INDIACom*). New Delhi, India, 637-641.

17. Dedakia, M.; and Mistry, K. (2015). Phishing detection using content based associative classification data mining. *Journal of Engineering Computers and Applied Sciences*, 4(7), 209-214.
18. Balogun, A.O.; Basri, S.; Abdulkadir, S.J.; Adeyemo, V.E.; Imam, A.A.; and Bajeh, A.O. (2019). Software defect prediction: analysis of class imbalance and performance stability. *Journal of Engineering Science and Technology*, 14(6), 3294-3308.
19. Balogun, A.O.; Basri, S.; Abdulkadir, S.J.; and Hashim, A.S. (2019). Performance analysis of feature selection methods in software defect prediction: a search method approach. *Applied Sciences*, 9(13).
20. Balogun, A.O.; Basri, S.; Mahamad, S.; Abdulkadir, S.J.; Almomani, M.A.; Adeyemo, V.E.; Al-Tashi, Q.; Mojeed, H.A.; Imam, A.A.; and Bajeh, A.O. (2020). Impact of feature selection methods on the predictive performance of software defect prediction models: an extensive empirical study. *Symmetry*, 12(7).
21. Mohammad, R.; Thabtah, F.; and McCluskey, T.L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, 25(2), 443-458.
22. Verma, R.; and Das, A. (2017). What's in a url: fast feature extraction and malicious URL detection. *Proceedings of the* 3*rd ACM on International Workshop on Security and Privacy Analytics*. Scottsdale Arizona, USA, 55-63.
23. Vrbančič, G.; Fister Jr, I.; and Podgorelec, V. (2018). Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing websites classification. *Proceedings of the* 8*th International Conference on Web Intelligence, Mining and Semantics*. Novi Sad, Serbia, 1-8.
24. Alqahtani, M. (2019). Phishing websites classification using association classification (PWCAC). *Proceedings of the International Conference on Computer and Information Sciences* (*ICCIS*). Sakaka, Saudi Arabia, 1-6.
25. Abdelhamid, N.; Ayesh, A.; and Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13), 5948-5959.
26. Hadi, W.; Aburub, F.; and Alhawari, S. (2016). A new fast associative classification algorithm for detecting phishing websites. *Applied Soft Computing*, 48, 729-734.
27. Rahman, S.S.M.M.; Rafiq, F.B.; Toma, T.R.; Hossain, S.S.; and Biplob, K.B.B. (2020). Performance assessment of multiple machine learning classifiers for detecting the phishing URLs. *Data Engineering and Communication Technology*, 285-296.
28. Aydin, M.; and Baykal, N. (2015). Feature extraction and classification phishing websites based on URL. *Proceedings of the IEEE Conference on Communications and Network Security* (*CNS*). Florence, Italy, 769-770.
29. Ubing, A.A.; Jasmi, S.K.; Abdullah, A.; Jhanjhi, N.; and Supramaniam, M. (2019). Phishing website detection: an improved accuracy through feature selection and ensemble learning. *International Journal of Advanced Computer Science and Applications*, 10(1), 252-257.

30. Aziz, A.S.A.; Hanafi, S.E.; and Hassanien, A.E. (2017). Comparison of classification techniques applied for network intrusion detection and classification. *Journal of Applied Logic*, 24(A), 109-118.
31. Guzmán, G.; Torres-Ruiz, M.; Tambonero, V.; Lytras, M.D.; Lopez-Ramirez, B.; Quintero, R.; Moreno-Ibarra, M.; and Alhalabi, W. (2018). A collaborative framework for sensing abnormal heart rate based on a recommender system: Semantic recommender system for healthcare. *Journal of Medical and Biological Engineering*, 38(6), 1026-1045.
32. Adeyemo, V.E.; Balogun, A.O.; Mojeed, H.A.; Oluwatobi, A.N.; and Adewole, K.S. (2020). Ensemble-based logistic model trees for website phishing detection. *Proceedings of the International Conference on Advances in Cyber Security*.
33. Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon, 202-207.
34. Chen, W.; Zhang, S.; Li, R.; and Shahabi, H. (2018). Performance evaluation of the GIS-based data mining techniques of best-first decision tree, random forest, and naïve Bayes tree for landslide susceptibility modeling. *Science of The Total Environment*, 644, 1006-1018.
35. Wang, S.; Jiang, L.; and Li, C. (2015). Adapting naive Bayes tree for text classification. *Knowledge and Information Systems*, 44(1), 77-89.
36. Balogun, A.O.; Balogun, A.M.; Sadiku, P.O.; and Amusa, L.B. (2017). An ensemble approach based on decision tree and bayesian network for intrusion detection. *Annals, Computer Science Series*, 15(1), 82-91.
37. Shi, H. (2006). *Best-first decision tree learning*. Master thesis, The University of Waikato.
38. Kaushik, S.; Chouhan, U.; and Dwivedi, A. (2017). Prediction of protein subcellular localization of human protein using J48, random forest and best first tree techniques. *Advances in Applied Science Research*, 1(12).
39. Collell, G.; Prelec, D.; and Patil, K.R. (2018). A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*, 275, 330-340.
40. Bühlmann, P. (2004). *Bagging, boosting and ensemble methods*. *Handbook of Computational Statistics*, 985-1022.
41. Wang, F.; Li, Z.; He, F.; Wang, R.; Yu, W.; and Nie, F. (2019). Feature learning viewpoint of adaboost and a new algorithm. *IEEE Access*, 7, 149890-149899.
42. Khan, F.; Ahamed, J.; Kadry, S.; and Ramasamy, L.K. (2020). Detecting malicious URLs using binary classification through adaboost algorithm. *International Journal of Electrical and Computer Engineering*, 10(1).
43. Sun, B.; Chen, S.; Wang, J.; and Chen, H. (2016). A robust multi-class AdaBoost algorithm for mislabeled noisy data. *Knowledge-Based Systems*, 102, 87-102.
44. Webb, G.I. (2000). Multiboosting: A technique for combining boosting and wagging. *Machine Learning*, 40(2), 159-196.
45. Pham, B.T.; Jaafari, A.; Prakash, I.; and Bui, D.T. (2019). A novel hybrid intelligent model of support vector machines and the MultiBoost ensemble for

landslide susceptibility modeling. *Bulletin of Engineering Geology and the Environment*, 78(4), 2865-2886.

46. Balogun, A.O.; Bajeh, A.O.; Orie, V.A.; and Yusuf-Asaju, W.A. (2018). Software defect prediction using ensemble learning: An ANP based evaluation method. *FUOYE Journal of Engineering and Technology*, 3(2), 50-55.
47. Jimoh, R.G.; Balogun, A.O.; Bajeh, A.O.; and Ajayi, S. (2018). A PROMETHEE based evaluation of software defect predictors. *Journal of Computer Science and Its Application*, 25(1), 106-119.
48. Xu, Z.; Liu, J.; Yang, Z.; An, G.; and Jia, X. (2016). The impact of feature selection on defect prediction performance: An empirical comparison. *Proceedings of the* 27*th International Symposium on Software Reliability Engineering* (*ISSRE*). Ottawa, Canada, 309-320.
49. Yu, Q.; Jiang, S.; and Zhang, Y. (2017). The performance stability of defect prediction models with class imbalance: An empirical study. *IEICE Transactions on Information and Systems*, 100(2), 265-272.
50. Balogun, A.O.; Lafenwa-Balogun, F.B.; Mojeed, H.A.; Adeyemo, V.E.; Akande, O.N.; Akintola, A.G.; Bajeh, A.O.; and Usman-Hamza, F.E. (2020). SMOTE-based homogeneous ensemble methods for software defect prediction. *Proceedings of the International Conference on Computational Science and Its Applications*. University of Malaga, Spain.
51. Yadav, S.; and Shukla, S. (2016). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. *Proceedings of the IEEE* 6*th International Conference on Advanced Computing* (*IACC*). Bhimavaram, India, 78-83.
52. Arlot, S.; and Lerasle, M. (2016). Choice of V for V-fold cross-validation in least-squares density estimation. *Journal of Machine Learning Research*, 17(1), 7256-7305.
53. Balogun, A.O.; Basri, S.; Jadid, S.A.; Mahamad, S.; Al-momani, M.A.; Bajeh, A.O.; and Alazzawi, A.K. (2020). Search-based wrapper feature selection methods in software defect prediction: an empirical analysis. Advances in Intelligent Systems and Computing, 1224.
54. Balogun, A.O.; Basri, S.; Mahamad, S.; Abdulkadir, S.J.; Capretz, L.F.; Imam, A.A.; Almomani, M.A.; Adeyemo, V.E.; and Kumar, G. (2021). Empirical analysis of rank aggregation-based multi-filter feature selection methods in software defect prediction. *Electronics*, 10(2).
55. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I.H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1), 10-18.
56. Al-Ahmadi, S.; and Lasloum, T. (2020). PDMLP: phishing detection using multilayer perceptron. *International Journal of Network Security and Its Applications*, 12(3), 59-72.
57. Lee, S.-J.; Xu, Z.; Li, T.; and Yang, Y. (2018). A novel bagging C4.5 algorithm based on wrapper feature selection for supporting wise clinical decision making. *Journal of Biomedical Informatics*, 78, 144-155.
58. Bhuyan, M.H.; Ma, M.; Kadobayashi, Y.; and Elmroth, E. (2019). Information-theoretic ensemble learning for DDoS detection with adaptive

boosting. *Proceedings of the* 31*st International Conference on Tools with Artificial Intelligence* (*ICTAI*). Portland, USA, 995-1002.

59. Cheng, K.; Gao, S.; Dong, W.; Yang, X.; Wang, Q.; and Yu, H. (2020). Boosting label weighted extreme learning machine for classifying multi-label imbalanced data. *Neurocomputing*, 403, 360-370.
60. Subasi, A.; Kadasa, B.; and Kremic, E. (2020). Classification of the cardiotocogram data for anticipation of fetal risks using bagging ensemble classifier. *Procedia Computer Science*, 168, 34-39.
61. Ali, W.; and Malebary, S. (2020). Particle swarm optimization-based feature weighting for improving intelligent phishing website detection. *IEEE Access*, 8, 116766-116780.