Heliyon 7 (2021) e07437

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Improving the phishing website detection using empirical analysis of Function Tree and its variants

Abdullateef O. Balogun [a], Kayode S. Adewole [a], Muiz O. Raheem [a], Oluwatobi N. Akande [b,*], Fatima E. Usman-Hamza [a], Modinat A. Mabayoje [a], Abimbola G. Akintola [a], Ayisat W. Asaju-Gbolagade [a], Muhammed K. Jimoh [c], Rasheed G. Jimoh [a], Victor E. Adeyemo [d]

[a] Department of Computer Science, University of Ilorin, Ilorin, Nigeria
[b] Department of Computer Science, Landmark University, Omu-Aran, Kwara State, Nigeria
[c] Department of Education Technology, University of Ilorin, Ilorin, Nigeria
[d] School of Built Environment, Engineering and Computing, Leeds Beckett University, Headingley Campus, Leeds, LS6 3QS, United Kingdom

ARTICLE INFO

Keywords:
Bagging
Boosting
Ensemble
Functional trees
Machine learning
Meta-learning
Phishing websites
Rotation forest

ABSTRACT

The phishing attack is one of the most complex threats that have put internet users and legitimate web resource owners at risk. The recent rise in the number of phishing attacks has instilled distrust in legitimate internet users, making them feel less safe even in the presence of powerful antivirus apps. Reports of a rise in financial damages as a result of phishing website attacks have caused grave concern. Several methods, including blacklists and machine learning-based models, have been proposed to combat phishing website attacks. The blacklist anti-phishing method has been faulted for failure to detect new phishing URLs due to its reliance on compiled blacklisted phishing URLs. Many ML methods for detecting phishing websites have been reported with relatively low detection accuracy and high false alarm. Hence, this research proposed a Functional Tree (FT) based meta-learning models for detecting phishing websites. That is, this study investigated improving the phishing website detection using empirical analysis of FT and its variants. The proposed models outperformed baseline classifiers, meta-learners and hybrid models that are used for phishing websites detection in existing studies. Besides, the proposed FT based meta-learners are effective for detecting legitimate and phishing websites with accuracy as high as 98.51% and a false positive rate as low as 0.015. Hence, the deployment and adoption of FT and its meta-learner variants for phishing website detection and applicable cybersecurity attacks are recommended.

1. Introduction

The increasing acceptance and adoption of information technology (IT) have led to an increase in the number of web-based solutions provided via cyberspace [1]. These activities range from essential services such as financial transactions to basic activities like e-health applications and education [2, 3]. Research has shown that financial transactions, online gaming services and social media are considered top web-based solutions with vast popularity and enormous users. The enormous magnitude of users of these web-based solutions indicates its acceptance in recent times. The aim is to increase the accessibility and availability of web-based solutions needed on a day-to-day basis. Nonetheless, the open accessibility and availability of these web-based solutions in cyberspace create avenues for cyber-attacks as there are no generic control measures to cyberspace [4, 5]. These cyber-attacks generate critical vulnerabilities and threats for both the web-based solutions and end-users with information as well as financial losses as major aftermaths of cyber-attacks. The website phishing attack is a typical example of these cyber-attacks. Nowadays, illegitimate websites are created by cyber-criminals to steal sensitive information from unsuspecting users for illegal activities [6]. The website phishing attack is a serious cybersecurity problem that overwhelms cyberspace and has a devastating effect on internet users and web-based businesses [7, 8]. According to Vrbančič, et al. [9], the website phishing attack is a pervasive fraud that happens when an illegitimate website looks exactly like a legitimate website with the sole purpose of acquiring data from unsuspecting users. This makes phishing attack a notable threat to web-based infrastructures [10, 11]. Specifically, in 2018, the Anti-Phishing Working Group (APWG) disclosed the presence of 51,401 phishing websites in cyberspace. In 2016, RSA reported that international organizations lost about $9 billion to phishing attacks [12,

* Corresponding author.
E-mail address: akande.noah@lmu.edu.ng (O.N. Akande).

https://doi.org/10.1016/j.heliyon.2021.e07437
Received 6 March 2021; Received in revised form 30 April 2021; Accepted 25 June 2021

13]. These events have shown that phishing attacks via illegitimate websites are fast gaining momentum; thus causing huge financial losses and burden just as available solutions may not be efficient in addressing the problem [10, 11, 14].

Several anti-phishing solutions have been proposed and developed by different cybersecurity experts and researchers for detecting phishing sites [15, 16, 17]. One of these solutions is the blacklist-based identification of website phishing attacks. To assess its validity, the blacklist method that is implemented by web browsers compares the requested universal resource locator (URLs) with stored phishing website URLs. A significant disadvantage of blacklist anti-phishing methods is its failure to detect new phishing URLs due to its reliance on compiling blacklisted phishing URLs [18, 19]. Moreover, cyber-attackers are deploying dynamic strategies that can easily elude the blacklist method [20]. Concerning the dynamism of cyber-attacks, Machine learning (ML) based solutions are used to assess the validity of websites to manage the complex existence of website phishing attacks on features derived from websites [21]. The purpose of this is to provide resilience in identifying new websites from phishing websites [15, 16, 17, 22]. However, in detecting phishing websites, the efficiency of the ML-based phishing detection solution depends on the performance of the chosen ML technique. Many ML methods in detecting phishing websites have been used and reported with relatively low detection accuracy values and high false-positive rates [23, 24]. This can be due to the existence of data quality issues such as class imbalance that have adverse effects on ML method performance [25, 26, 27]. The dynamism of phishing websites also calls for more sophisticated ML techniques with a high detection rate of phishing and low false-positive rates [28].

Therefore, this study proposed Functional Tree (FT) based meta-learning models for the detection of phishing websites. FT, by way of positive induction, combines a decision tree with a linear function such that the developed decision tree will have multivariate decision nodes and leaf nodes that uses discriminant functions to make predictions.

Specifically, the following are contributions of this study to the body of knowledge:

1) Implementation of the FT algorithm and its variants for detecting both legitimate and phishing websites.
2) Implementation of Bagging, Boosting, and Rotation Forest Meta-learners for improving FT performance; and
3) An empirical comparison of proposed methods with existing state-of-the-art phishing methods.

More so, it is the intention of this study to answer the following research questions:

1) How effective are FT algorithm implementations for detecting phishing and legitimate websites?
2) How effective is the Meta-learners (Bagged-FT, Boosted-FT, and Rotation forest FT) in detecting phishing and legitimate websites?
3) How well is the performance of the proposed FT and its variants compared with existing state-of-the-art methods?

The remaining part of this paper is organized as follows. Section 2 discusses the review of related works. Section 3 illustrates the research methodology, the overview of the proposed models and implemented algorithms. Section 4 presents the research experiment and experimental results analyses. Lastly, Section 5 concludes and indicates future works.

2. Related works

This section reviews and discusses existing phishing detection methods developed with different ML techniques.

Mohammad, et al. [29] applied a self-structuring neural network to detect phishing websites. Their model is based on an adaptive measure that adjusts its learning rate before adding new neurons and subsequently the network structure. On evaluation, the proposed model had accuracy values of 94.07%, 92.48% and 91.12% of the training, testing and validation sets respectively. Verma and Das [30] in their study deployed a Deep Belief Network (DBN) to detect phishing websites. The DBN model extracts deep hierarchical representation from the given dataset by using Restricted Boltzmann machines (RBM) to develop its model. The performance of the proposed DBN was superior to the decision tree and Random Forest (RF) with an accuracy of 94.43%. Ali and Ahmed [14] in their study used features selected by genetic algorithm (GA) on a deep neural network (DNN) to detect phishing websites. From their experimental results, the performance of the proposed approach outperformed baseline classifiers such as decision tree (DT), k-Nearest neighbour (KNN), support vector machine (SVM), back-propagation neural network (BP) and Naïve Bayes (NB). Vrbančič, et al. [9] used bat meta-heuristics algorithm to enhance DNN. The proposed method had a maximum accuracy of 96.9%. These studies show that neural network models can be as effective as base-line classifiers in detecting phishing websites.

Alqahtani [31] deployed a novel association rule induction method for phishing website detection. The proposed method uses an association rule method to determine the legitimacy of a website. Their experimental results showed the effectiveness of the proposed method as it outperforms baseline classifiers such as decision tree, RIPPER and some classification models based on associative learning with an accuracy value of 95.20% and an F-measure value of 0.9511. Similarly, Abdelhamid, et al. [32] used a Multi-label Classifier based Associative Classification (MCAC) approach for phishing detection. The MCAC technique employed rules discovery, classifier building and class assignment to extract sixteen (16) unique features from website URL for the detection task. From the experimental result, MCAC outperformed RIPPER, DT, PART, CBA, and MCAR base classifiers in terms of accuracy. Dedakia and Mistry [24] proposed a Content-Based Associative Classification method (CBAC) for phishing detection. The proposed method extends the Multi-Label Class Associative Classification (MCAC) algorithm by considering content-based features. From the experimental results, the proposed method (CBAC) had an accuracy value of 94.29%. Hadi, et al. [33] developed and investigated the performance of a fast AC algorithm (FACA) with other existing associative classification (AC) methods (CBA, CMAR, MCAR and ECAR) on phishing website detection. Their experimental results show the superiority of FACA over other AC methods based on accuracy and F-measure values. The effectiveness of these associative-based approaches shows their applicability to phishing detection. However, their relatively low accuracy value is a drawback and phishing detection models with high detection accuracy are imperative.

Rahman, et al. [34] investigated the effectiveness of selected ML methods and ensemble methods (KNN, DT, SVM, RF, Extreme Randomized Tree (ERT) and Gradient Boosting Tree (GBT)) in website phishing detection. Similarly, Chandra and Jana [23] studied the improvement in phishing website detection using meta-classifiers. Their respective results showed that the performances of ensemble methods are superior to the single classifiers. Alsariera, et al. [11] developed ensemble variants of Forest Penalizing by Attributes (ForestPA) for phishing website detection. ForestPA is based on weight assignment and an increment approach to construct efficient trees. Their experimental results depicted that the proposed meta-learner variants of ForestPA are very effective in detecting phishing websites with a minimum accuracy of 96.26%.

Chiew, et al. [12] proposed a hybrid ensemble FS (HEFS) method based on a novel cumulative distribution function gradient (CDF-g) method to select optimal features. The evaluation of HEFS with RF had an accuracy value of 94.6%. Similarly, Aydin and Baykal [35] used subset-based features that were extracted from a website URL for phishing detection. Alphanumeric character, keyword, security, domain identity and rank based analysis was carried out on the extracted features. Afterwards, NB and Sequential Minimal Optimization (SMO) were applied to the extracted features. An accuracy of 83.96% and 95.39% were achieved with NB and SMO respectively. Ubing, et al. [36]

proposed a phishing method based on feature selection (FS) and ensemble learning method (ELM). Random Forest Regressor (RFG) was used as the FS method and majority voting for the ELM. From the experimental results, the proposed framework outperforms existing methods such as NB, SVM, multilayer perceptron (MLP), RF, KNN, logistic regression (LR) and gradient boosting classifiers with accuracy, precision, f-measure values of 95.4%, 0.935 and 0.947 respectively. Although the proposed FS method is very effective, the ensuing accuracy value can be improved.

From the preceding reviews, there is a need for more effective and efficient solutions as most of the existing methods have comparatively low performance. Hence, this study proposes FT based meta-learners for phishing website detection.

3. Methodology

This section presents the research methodology employed in this study. Specifically, the proposed approaches, studied phishing datasets, evaluation metrics and experimental framework discussed.

3.1. Functional Tree and its variants

Functional Trees (FT) as proposed by Gama [37], is the hybridization of multivariate decision trees and discriminant function via constructive induction. FT is also referred to as a generalization of multivariate trees. FT incorporates features at leaf nodes and decision nodes. In some cases, FT incorporates features at both nodes and leaves for building classification trees such that decision nodes are created based on the growth of the classification tree and functional leaves are constructed as the tree is pruned [37]. For prediction tasks, FT can be deployed to predict the value of class variables for a given dataset. Specifically, the dataset traverses the tree from the root node to a leaf in which the set of features of the dataset is expanded at each decision node using the node-built constructor functions. The decision test of the node is subsequently applied to determine the path on which the dataset will proceed. Finally, the dataset use labelled as a leaf using either the constructor function based on the leaf or the leaf-related constant [37, 38]. The key distinction between conventional decision tree algorithms and FT is that these traditional algorithms split the input data into tree nodes by comparing the value with a constant of certain input attributes, while FT uses logistic regression functions for internal node splitting (called oblique split) and leaf prediction [39]. There are three variants of FT:

(1) FT full (FT-1) with regression models for both the inner nodes and the leaves;
(2) FT inner (FT-2) with regression models for only the inner nodes; and
(3) FT leaves (FT-3) used regression models for only leaves.

FT uses the gain ratio as the splitting criterion to select an input attribute to split upon, the standard decision tree for tree construction (in this case C4.5) to avoid overfitting and iterative reweighting (LogitBoost) for fitting the logistic regression functions at leaves with least-squares fits for each class $\boldsymbol{Y_i}$ as depicted in Eq. (1).

$$f_{Y_i} = \sum_{i=1}^{10} \beta_i X_i + \beta_0 \tag{1}$$

where $\boldsymbol{P}(\boldsymbol{X})$ is the probability predicted value; $\boldsymbol{\beta_i}$ is the coefficient of the i^{th} component in the input vector $\boldsymbol{X_i}$. The posterior probabilities in the leaf, $\boldsymbol{P}(\boldsymbol{X})$, are calculated using Eq. (2) [40].

$$\boldsymbol{P}(\boldsymbol{X}) = \frac{\boldsymbol{e}^{2\boldsymbol{f}\boldsymbol{y}_i(\boldsymbol{x})}}{1 + \boldsymbol{e}^{2\boldsymbol{f}\boldsymbol{y}_i(\boldsymbol{x})}} \tag{2}$$

In this study, the three (3) variants of FT leaves are explored and implemented.

3.1.1. Functional Tree only (FT-1)

Functional Tree (FT-1) can be used to predict the value of the target attribute by performing a complete traversal of the tree from the root node to a leaf. At the decision nodes of the tree, the attributes of a given dataset can be extended using a constructor function implemented at each of these nodes. In the end, the decision test of the node is used for defining the path the dataset will traverse. If a leaf is encountered, the dataset is classified using either the constant associated with the leaf or the constructor function built at this leaf.

3.1.2. Functional Tree with leaves only (FT-2)

In Functional Tree with leaves only represented as FT-2, the functional models are used as leaves instead of splitting test. A similar approach is used in developing Naïve Bayes Tree (NBTree) and M5 model tree [41, 42]. It involves restricting the selection of test attributes to the original attributes as shown in Algorithm 1 (step 4). However, the constructor function is still implemented at each node which is used later for pruning. Consequently, the original attributes are used for constructing the decision nodes. The only case whereby a leaf node has a constructor model is when the estimated error of the constructor is less than the back-up-error and

Algorithm 1.
Function GrowTree (Dataset, Constructor)

1. if Stop_Criterion (Dataset)
   - Return a Leaf Node with a constant value.
2. Construct a model α using Constructor
3. For each example $\vec{x} \in$ Dataset
   - Compute $\hat{y} = \alpha(\vec{x})$
   - Extend $\vec{x}$ with new attributes $\hat{y}$
4. Select the attributes of original as well as newly constructed attributes that maximizes some merit-functions
5. For each partition i of the Dataset using selected attributes
   - $Tree_i = GrowTree\ (Dataset_i, Constructor)$
6. Return a $Tree$ as a decision node based on the selected attribute, containing the α model and descendants $Tree_i$

End Function

Figure 1. Building a functional tree.

Algorithm 2.
Function Prune (Constructor)

1. Estimate **Leaf_Error** as the error at this node.
2. If Tree is a leaf, Return **Leaf_Error**.
3. Estimate Constructor_Error as the estimated error at α
4. For each dependent i
   - Let p_i be the probability that an example goes through branch i
   - Backed-up-Error += $p_i \times Prune\ (Tree_i)$
5. If argmin (Leaf_Error, Constructor_Error, Backed-up-Error)
   - Is Leaf_Error
     - Tree = Leaf
     - Tree_Error = Leaf_Error
   - Is Model_Error
     - Tree = Constructor_Leaf
     - Tree_Error = Constructor_Error
   - Is Backed-up-Error
     - Tree_Error = Backed-up-Error
6. Return Tree_Error

End Function

Figure 2. Pruning a functional tree.

static error in the pruning phase as indicated in Algorithm 2. Summarily, FT-2 partitions input space into hyper-rectangles whereby the data in each partition is fitted with a constructor function.

3.1.3. Functional Tree with inner nodes (FT-3)

For Functional Tree inner nodes depicted as FT-3, multivariate models are exclusively used at decision nodes (internal nodes). This is a result of conditioning the pruning algorithm to back-up-error and static error options. FT-3 partitions the input space oblique decision surfaces and data in each partition is fitted with a constant that minimizes the given loss function.

3.2. Meta-learners

3.2.1. Bagging

Bootstrap Aggregating (Bagging) method is a homogeneous meta-learner used for amplifying the prediction performance of base-line learners [43, 44]. In bagging, baseline learners are trained using insights derived from the original dataset. These insights are extracted from different subsets formed from the original dataset [43]. Bagging guarantees the reduction in the variance of developed ensuing models while keeping the bias of the same models from increasing by applying aggregation technique on all the developed models. Also, bagging meta-leaner deploys random resampling on the given dataset and generates multiple base-line models by fitting base-line learners on the resampled subsets. In the end, bagging aggregates generated baseline models into single model prediction processes [11]. Algorithm 3 presents the pseudocode for the Bagging algorithm as used in this study (see Figures 1, 2, 3, 4 and 5).

3.2.2. Boosting

Boosting is a meta-learner that sequentially applies weak base-line learner on a re-weighted training dataset [45]. According to Sun, et al.

Algorithm 3.
The Bagging Algorithm
Input: training set S,
Base-Line Learner: *FT-1, FT-2, FT-3*
integer T (number of bootstrap samples).

1. for $i = 1$ to T {
2. S' = bootstrap sample from S (i.i.d. 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

Figure 3. Bagging algorithm.

Algorithm 4.
The Boosting Algorithm
Input: Training set S = $\{x_i, y_i\}, i = 1 \ldots m, y_i \in Y, Y = \{c_1, c_2, \ldots, c_k\}, c_k$is the class label;
The number of Iterations T;
Base-Line Learner: *FT-1, FT-2, FT-3*
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 weak hypothesis
$h_t = X \rightarrow \{c_1, c_2, \ldots, c_k\}$
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 ε_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 hypothesis
$$H(x) = \arg max\left(\sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right)[Y \neq h_t(X)]\right)$$

Figure 4. Boosting algorithm.

[46], at the end of boosting the meta-learner training phase, a majority vote rule is applied to the generated hypotheses from the weak base-line learner for making its final decision into a final hypothesis. In this study, boosting meta-learner is implemented based on an extended version of AdaBoost meta-leaner (AdaBoost.M1) [11]. AdaBoost.M1 algorithm as presented in Algorithm 4, is developed for binary classification purposes and thus justifies the selection of the algorithm for detecting a phishing website.

3.2.3. Rotation Forest

Rotation Forest (RF) meta-learner generates classifier models using feature extraction. RF creates training data for a baseline learner by randomly splitting the feature set into *N* subsets and principal component analysis (PCA) is deployed on each of the generated subsets [47, 48]. To maintain the variability in the data, all principal components are kept. Hence, *N* axis rotations occur to create new features for the baseline learner. The essence of the rotation is to allow concurrent independent accuracy and diversity within the ensemble. Diversity is attained via feature extraction for each baseline learner.

RF algorithm is presented in Algorithm 5 with the assumption that *X* is the training dataset, *Y* is the class label and *F* is feature sets.

3.3. Experimental framework

This section discusses the experimental method presented in Figure 6 as used in this analysis. The experimental system is structured to empirically test and verify the efficacy of the proposed methods for phishing website detection. For training and evaluating the proposed techniques, three phishing datasets from the UCI repositories are used and the K-fold (where k = 10) cross-validation (CV) approach is used for the creation and evaluation of the phishing models. The 10-fold CV option is based on its ability to create phishing models with the low impact of the issue of class imbalance [49, 50, 51, 52, 53]. Moreover, the K-fold CV approach ensures that each instance can be used iteratively for both

Algorithm 5.
Rotation Forest Algorithm
Input: Training set X = $\{x_i, y_i\}, i = 1 \ldots m, y_i \in Y, Y = \{c_1, c_2, \ldots, c_k\}, c_k$is the class label;
Base-Line Learner: *FT-1, FT-2, FT-3*
1. Choose a value for K which is a factor of n, let F randomly divided into K parts of the distinct subsets while each subset must contain $N = n/k$ number of features.
2. Select the corresponding columns of attributes in the subset $T_{i,j}$ from the training dataset X, then form a new matrix $X_{i,j}$. Extract a bootstrap subset of objects ¾ of X to make a new training dataset $X'_{i,j}$.
3. Use Matrix $X'_{i,j}$ as feature transform to produce the co-efficient in the matrix $P_{i,j}$, which j[th] column coefficient is the characteristic component j[th].
4. Construct a sparse rotation matrix S_i using the obtained coefficient obtained in the matrix $P_{i,j}$.
Output: classifier T_i of $d_{i,j}(XS_i^f)$ to determine x belonging to the class y_i
Then, Calculate class confidence:
$$\alpha_j(x) = \frac{1}{L}\sum_{i=1}^{L} d_{i,j}(XS_i^f)$$
Assign the category with the largest α_j x value to x.

Figure 5. Rotation forest algorithm.

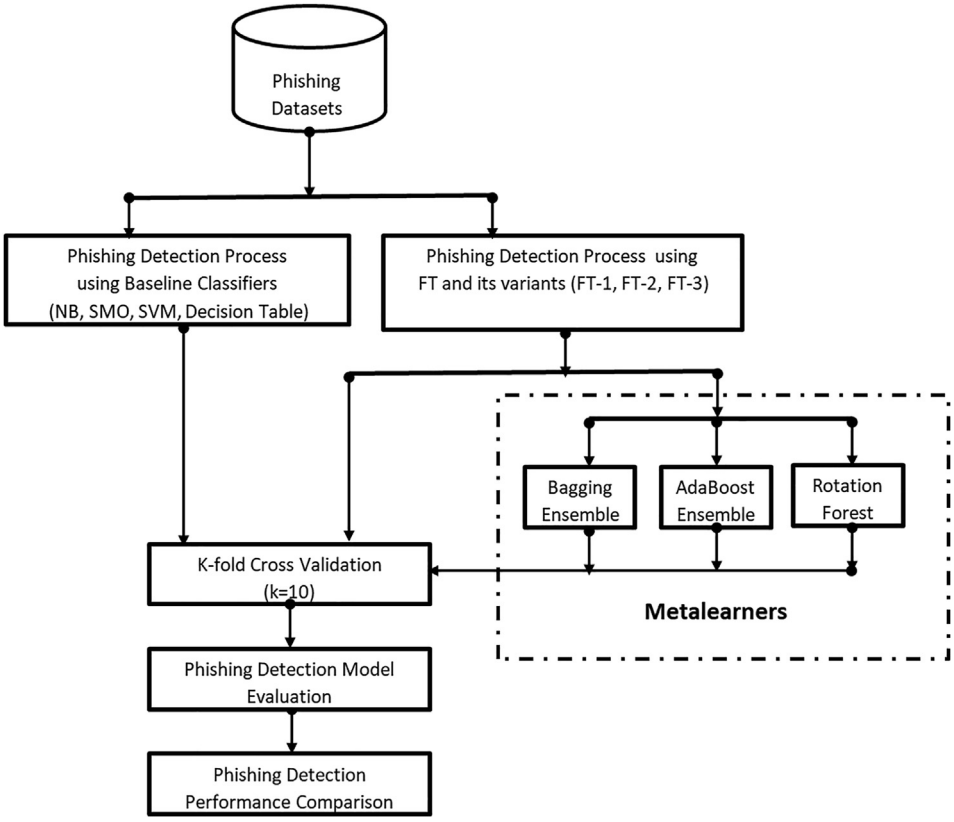

Figure 6. Experimental framework.

training and testing [54, 55, 56]. On phishing datasets, based on 10-fold CV, the proposed methods and the chosen baseline classifiers (NB, SMO, SVM, and Decision Table (Dec Table)) are then implemented. The phishing detection efficiency of the developed phishing models is then tested and contrasted with other experimented methods of phishing detection. All experiments were performed using the WEKA machine learning tool in the same environment [57].

3.4. Website phishing datasets

Three phishing datasets were used in the experimentation phase of this study. These datasets are readily accessible and commonly used in current studies [11, 12, 29, 34, 58]. The first dataset (Dataset 1) consists of 11,055 instances (4,898 phishing and 6,157 legitimate instances). Dataset 1 has 30 independent features which describe the dataset [29]. The second dataset (Dataset 2) has 10,000 instances divided equally into 5,000 phishing and 5,000 legitimate instances. Dataset 2 has 48 features which range from discrete, continuous and categorical values [12, 34]. The third dataset (Dataset 3) has 1,353 (702 phishing, 548 legitimate, and 103 suspicious) instances with 10 features. Dataset 3 has 3 class labels which make it quite different from Dataset 1 and Dataset 2. For more information on the phishing datasets, refer to [11, 12, 29, 34, 58].

3.5. Performance evaluation metrics

The detection performances of the developed phishing models are evaluated using Accuracy, F-measure, Area under the Curve (AUC), false-positive rate (FPR), true positive rate (TPR), and Mathew's correlation coefficient (MCC) performance evaluation metric. Preference for these metrics is based on the wide and frequent usage of these evaluation metrics for phishing website detection from existing studies [2, 4, 10, 11, 34, 36].

i. Accuracy measures the overall rate at which the actual labels of all instances are correctly predicted [59]. It was calculated using Eq. (4):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

Table 1. Performance comparison of FT and its variants with baseline classifiers on Dataset 1.

	FT-1	FT-2	FT-3	NB	SVM	SMO	Dec Table
Accuracy (%)	95.50	**96.07**	95.22	90.70	94.60	92.70	93.44
F-Measure	0.955	**0.961**	0.952	0.907	0.946	0.927	0.934
AUC	0.973	**0.987**	0.951	0.962	0.944	0.925	0.981
TP-Rate	0.955	**0.961**	0.952	0.907	0.946	0.927	0.934
FP-Rate	0.048	**0.041**	0.051	0.098	0.059	0.078	0.073
MCC	0.909	**0.920**	0.903	0.811	0.891	0.852	0.867

Table 2. Performance comparison of FT and its variants with baseline classifiers on Dataset 2.

	FT-1	FT-2	FT-3	NB	SVM	SMO	Dec Table
Accuracy (%)	96.79	**97.86**	96.64	85.15	91.49	93.87	95.79
F-Measure	0.968	**0.979**	0.966	0.850	0.915	0.939	0.958
AUC	0.977	**0.992**	0.966	0.949	0.915	0.939	0.982
TP-Rate	0.968	**0.979**	0.966	0.852	0.915	0.939	0.958
FP-Rate	0.032	**0.021**	0.034	0.149	0.085	0.061	0.042
MCC	0.936	**0.957**	0.933	0.715	0.830	0.878	0.916

Table 3. Performance comparison of FT and its variants with baseline classifiers on Dataset 3.

	FT-1	FT-2	FT-3	NB	SVM	SMO	Dec Table
Accuracy (%)	88.91	**90.24**	88.99	84.10	85.66	86.00	84.47
F-Measure	0.890	**0.903**	0.891	0.825	0.825	0.846	0.839
AUC	0.950	**0.970**	0.910	0.948	0.867	0.900	0.954
TP-Rate	0.889	**0.902**	0.890	0.841	0.857	0.860	0.845
FP-Rate	0.074	0.074	**0.071**	0.120	0.123	0.109	0.110
MCC	0.810	**0.826**	0.817	0.722	0.734	0.757	0.737

Table 4. Performance comparison of FT-based Meta-learners on Dataset 1.

	FT-1	FT-2	FT-3	RoF-FT-1	RoF-FT-2	RoF-FT-3	BG-FT-1	BG-FT-2	BG-FT-3	BT-FT-1	BT-FT-2	BT-FT-3
Accuracy (%)	95.50	96.07	95.22	96.78	96.83	96.49	96.77	96.57	96.44	97.00	**97.19**	96.9
F-Measure	0.955	0.961	0.952	0.968	0.968	0.965	0.968	0.966	0.964	0.970	**0.972**	0.969
AUC	0.973	0.987	0.951	0.995	**0.996**	0.988	0.995	0.995	0.990	**0.996**	0.995	0.995
TP-Rate	0.955	0.961	0.952	0.968	0.968	0.965	0.968	0.966	0.964	0.970	**0.972**	0.969
FP-Rate	0.048	0.041	0.051	0.035	0.033	0.037	0.035	0.036	0.037	0.032	**0.031**	0.033
MCC	0.909	0.920	0.903	0.935	0.936	0.929	0.935	0.93	0.928	0.939	**0.943**	0.937

Table 5. Performance comparison of FT-based Meta-learners on Dataset 2.

	FT-1	FT-2	FT-3	RoF-FT-1	RoF-FT-2	RoF-FT-3	BG-FT-1	BG-FT-2	BG-FT-3	BT-FT-1	BT-FT-2	BT-FT-3
Accuracy (%)	96.79	97.86	96.64	97.43	98.32	97.4	97.58	98.21	97.33	98.11	**98.51**	97.84
F-Measure	0.968	0.979	0.966	0.974	0.983	0.974	0.976	0.982	0.973	0.981	**0.985**	0.978
AUC	0.977	0.992	0.966	0.996	**0.998**	0.994	0.996	0.997	0.994	0.997	**0.998**	0.997
TP-Rate	0.968	0.979	0.966	0.974	0.983	0.974	0.976	0.982	0.973	0.981	**0.985**	0.978
FP-Rate	0.032	0.021	0.034	0.026	0.017	0.026	0.024	0.018	0.027	0.019	**0.015**	0.022
MCC	0.936	0.957	0.933	0.949	0.966	0.948	0.952	0.964	0.947	0.962	**0.970**	0.957

ii. F-measure is the weighted average of both the Recall (R) and Precision (P) metrics. It emphasizes how good a classifier is in maximizing both precisions and recall simultaneously. F-measure can be computed as defined in Eq. (5)

$$F-\text{measure} = \frac{2 \text{ x } TP}{2 \text{ x } TP + FP + FN} \tag{5}$$

iii. The area under the curve (AUC) plots the FP rate on the X-axis and plots the TP rate on the Y-axis. AUC is not susceptible to the majority class bias and does not ignore the minority class during its evaluation.

iv. False Positive Rate (FPR) is the number of legitimate instances that were incorrectly identified as phishing attacks. This was computed using Eq. (6):

Table 6. Performance comparison of FT-based Meta-learners on Dataset 3.

	FT-1	FT-2	FT-3	RoF-FT-1	RoF-FT-2	RoF-FT-3	BG-FT-1	BG-FT-2	BG-FT-3	BT-FT-1	BT-FT-2	BT-FT-3
Accuracy (%)	88.91	90.24	88.99	89.87	**91.06**	89.80	88.77	90.32	88.70	89.06	89.28	87.73
F-Measure	0.890	0.903	0.891	0.899	**0.911**	0.898	0.888	0.903	0.887	0.891	0.893	0.877
AUC	0.950	0.970	0.910	0.973	**0.977**	0.954	0.972	0.978	0.962	0.963	0.967	0.966
TP-Rate	0.889	0.902	0.890	0.899	**0.911**	0.898	0.888	0.903	0.887	0.891	0.893	0.877
FP-Rate	0.074	0.074	0.071	0.071	**0.065**	0.07	0.079	0.073	0.076	0.082	0.079	0.091
MCC	0.810	0.826	0.817	0.824	**0.842**	0.825	0.808	0.828	0.810	0.808	0.812	0.785

Table 7. Detection Comparison of proposed methods with existing methods on Dataset 1.

Phishing Models	Accuracy (%)	F-Measure	AUC	TP-Rate	FP-Rate	MCC
Aydin and Baykal [35]	95.39	0.938	0.936	-	0.046	-
Dedakia and Mistry [24]	94.29	-	-	-	-	-
Mohammad, et al. [29]	92.18	-	-	-	-	-
Ubing, et al. [36]	95.40	0.947	-	-	0.041	-
Ali and Ahmed [14]	91.13	-	-	-	-	-
Verma and Das [30]	94.43	-	-	-	-	-
Hadi, et al. [33]	92.40	-	-	-	-	-
Chiew, et al. [12]	93.22	-	-	-	-	-
Rahman, et al. [34] (KNN)	94.00	-	-	-	0.049	-
Rahman, et al. [34] (SVM)	95.00	-	-	-	0.039	-
Chandra and Jana [23]	92.72	-	-	-	-	-
Folorunso, et al. [60] (Stacking)	95.97	-	-	-	-	-
Folorunso, et al. [60] (Hybrid NBTree)	94.10	-	-	-	-	-
Al-Ahmadi and Lasloum [61]	96.65	0.965	-	-	-	-
Alsariera, et al. [11]	96.26	-	0.994	-	0.050	-
Ali and Malebary [62]	96.43	-	-	-	-	-
Ferreira, et al. [6]	87.61	-	-	-	-	-
Vrbančič, et al. [9]	96.50	-	-	-	-	-
*RoF-FT-1	96.78	0.968	0.995	0.968	0.035	0.935
*RoF-FT-2	96.83	0.968	**0.996**	0.968	0.033	0.936
*RoF-FT-3	96.49	0.965	0.988	0.965	0.037	0.929
*BG-FT-1	96.77	0.968	0.995	0.968	0.035	0.935
*BG-FT-2	96.57	0.966	0.995	0.966	0.036	0.930
*BG-FT-3	96.44	0.964	0.990	0.964	0.037	0.928
*BT-FT-1	97.00	0.97	**0.996**	0.97	0.032	0.939
*BT-FT-2	**97.19**	**0.972**	0.995	**0.972**	**0.031**	**0.943**
*BT-FT-3	96.9	0.969	0.995	0.969	0.033	0.937

* Indicates methods proposed in this study.

Table 8. Detection Comparison of proposed methods with existing methods on Dataset 2.

Phishing Models	Accuracy (%)	F-Measure	AUC	TP-Rate	FP-Rate	MCC
Chiew, et al. [12]	94.60	-	-	-	-	-
Rahman, et al. [34] (KNN)	87.00	-	-	-	0.078	-
Rahman, et al. [34] (SVM)	91.00	-	-	-	0.067	-
*RoF-FT-1	97.43	0.974	0.996	0.974	0.026	0.949
*RoF-FT-2	98.32	0.983	**0.998**	0.983	0.017	0.966
*RoF-FT-3	97.4	0.974	0.994	0.974	0.026	0.948
*BG-FT-1	97.58	0.976	0.996	0.976	0.024	0.952
*BG-FT-2	98.21	0.982	0.997	0.982	0.018	0.964
*BG-FT-3	97.33	0.973	0.994	0.973	0.027	0.947
*BT-FT-1	98.11	0.981	0.997	0.981	0.019	0.962
*BT-FT-2	**98.51**	**0.985**	**0.998**	**0.985**	**0.015**	**0.970**
*BT-FT-3	97.84	0.978	0.997	0.978	0.022	0.957

* Indicates methods proposed in this study.

$$FPR = \frac{FP}{FP + TN} \times 100 \quad (6)$$

v. True Positive Rate (TPR) is the rate at which actual phishing website instances are correctly classified as that phishing website. This was computed using Eq. (7):

$$TPR = \frac{TP}{TP + FN} \times 100 \quad (7)$$

vi. Mathews Correlation Coefficient (MCC) is a valid statistical rate that only yields a high score if the forecast yields good results in all four groups of the confusion matrix (true positives, false negatives, true negatives and false positives), in proportion to the size of the positive elements and the size of the negative elements in the dataset. This was computed using Eq. (8):

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (8)$$

4. Results and discussion

The results obtained from the various experiments conducted are discussed in this section. The goal is to investigate and answer the research

Table 9. Detection Comparison of proposed methods with existing methods on Dataset 3.

Phishing Models	Accuracy (%)	F-Measure	AUC	TP-Rate	FP-Rate	MCC
Rahman, et al. [34] (KNN)	88.00	-	-	-	0.099	-
Rahman, et al. [34] (SVM)	87.00	-	-	-	0.087	-
*RoF-FT-1	89.87	0.899	0.973	0.899	0.071	0.824
*RoF-FT-2	**91.06**	**0.911**	**0.977**	**0.911**	**0.065**	**0.842**
*RoF-FT-3	89.80	0.898	0.954	0.898	0.070	0.825
*BG-FT-1	88.77	0.888	0.972	0.888	0.079	0.808
*BG-FT-2	90.32	0.903	0.978	0.903	0.073	0.828
*BG-FT-3	88.70	0.887	0.962	0.887	0.076	0.810
*BT-FT-1	89.06	0.891	0.963	0.891	0.082	0.808
*BT-FT-2	89.28	0.893	0.967	0.893	0.079	0.812
*BT-FT-3	87.73	0.877	0.966	0.877	0.091	0.785

* Indicates methods proposed in this study.

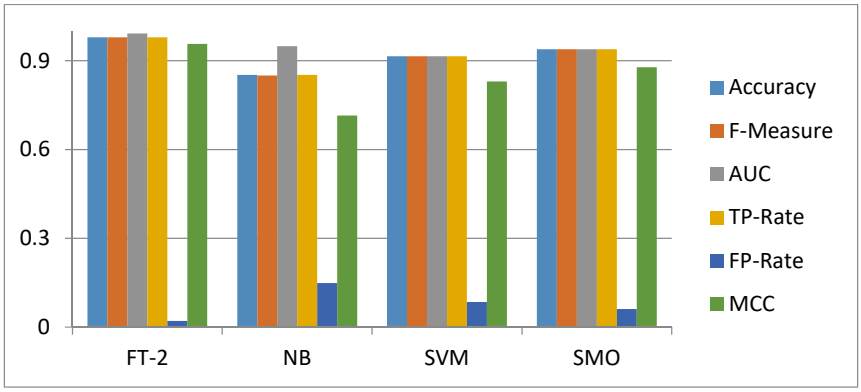

Figure 7. Performance comparison of FT-2 with baseline classifiers on Dataset 1.

questions raised in this study. We first conducted experiments based on the different variants of Functional Trees (FT), that is, FT-1, FT-2 and FT-3. The results obtained from these experiments were compared with baseline classifiers from existing studies to ascertain the effectiveness and efficacy of FT for phishing detection. The experiments were conducted using three datasets employed in this study. More specifically, Tables 1, 2, and 3 show FT variants' (FT-1, FT-2 and FT-3) performance comparisons with baseline classifiers on Dataset 1, Dataset 2 and Dataset 3 respectively. It was observed that Functional Trees variant FT-2 yielded the best performance in terms of the metrics used with Datasets 1-3. Tables 4, 5, and 6 compared the performance of the FT variants based on meta-learners. The best performance recorded were put in bold. Lastly, Tables 7, 8, and 9 compared the results of the proposed methods with existing state-of-the-art approaches. This section also presents some figures to visualize the implications of the results obtained. The best performance recorded were also put in bold while our proposed methods were asterisked.

4.1. Comparison of FT variants with baseline classifiers

As illustrated in Table 1, FT-2 outperformed the baseline classifiers in accuracy and other performance evaluation metrics. Regarding accuracy, FT-2 (96.07%) outperformed the baseline classifiers (NB (90.7%), SVM

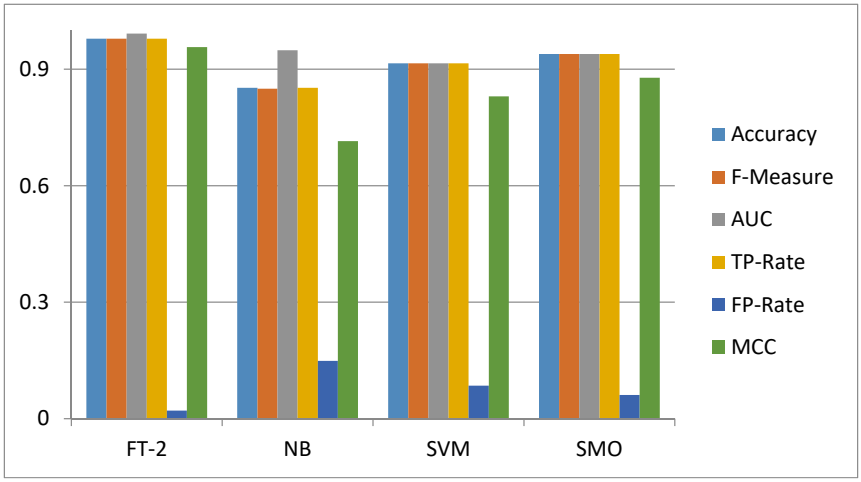

Figure 8. Performance comparison of FT-2 with baseline classifiers on Dataset 2.

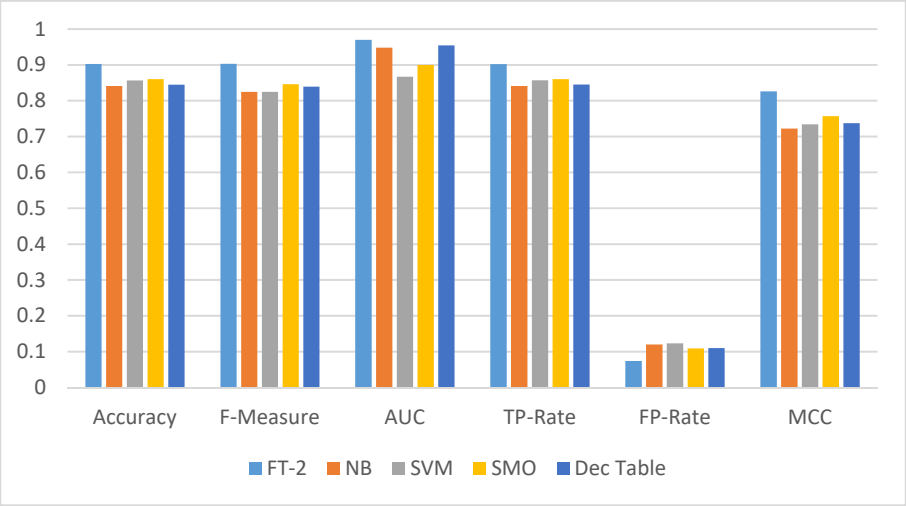

Figure 9. Performance comparison of FT-2 with baseline classifiers on Dataset 3.

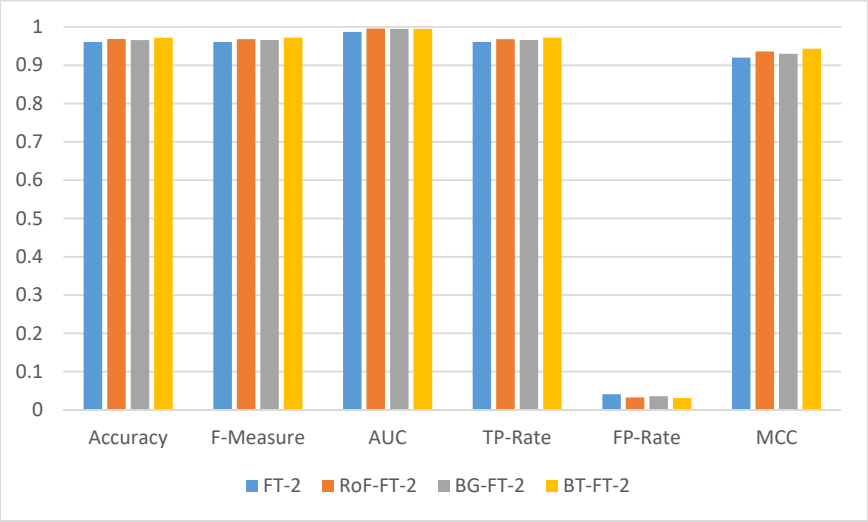

Figure 10. Performance comparison of FT-2 variant as a base classifier for meta-learners on Dataset 1.

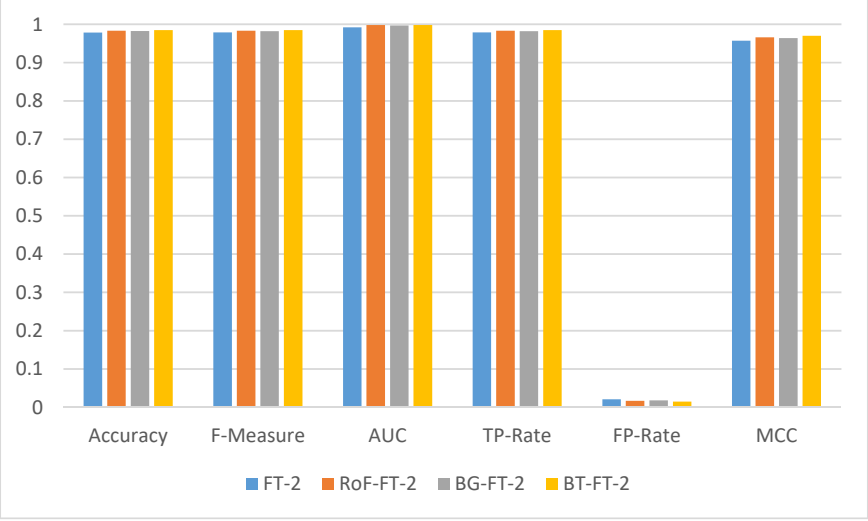

Figure 11. Performance comparison of FT-2 variant as a base classifier for meta-learners on Dataset 2.

(94.60%), SMO (92.7%) and Dec Table (93.44%)). Also, FT-2 (0.961) recorded a higher F-Measure value than the baseline classifiers (NB (0.907), SVM (0. 946), and SMO (0.927)). While on AUC, MCC, TP-Rate and FP-Rate values, FT-2 still has excellent performances over other prominent baseline classifiers. These results imply that the FT-2 variant of Functional Trees shows promising performances when used to detect phishing websites. This notable performance further shows the superiority of the proposed methods over the existing baseline approaches.

Table 2 indicates a comparison of phishing detection performance, using FT variants (FT-1. FT-2, FT-3) and baseline classifiers (MB, SMO, SVM, Dec Table) on Dataset 2. The table also signposts an exclusive comparison between the three FT variants and the implications of their results; then a broad performance comparison against the baseline classifiers. The performance evaluation comparison of the FT indicates the highest records at the FT-2 variant, considering almost all of the evaluation metrics (Accuracy 97.86, F-measure 0.979, AUC 0.992, TP-Rate 0.979, and MCC 0.957). The implication is that the selection of test attributes is quite restricted to the original attributes on Dataset 2. At the same time, the constructor function is still applied at each node and used later for pruning. Consequently, the input space is rather mostly partitioned into hyper-rectangles, so that the data in each of the divider is fitted with a constructor function. This is manifested, comparing the evaluation metrics of FT-2 with other FT variants. The FP-Rate evaluation metrics are observed lowest under it with a 0.021 score; implying that there are least instances of incorrectly identified phishing attacks under the FT-2 variant. The table also indicates that next to the FT-2 variant is the FT-1 counterpart concerning performance. A broad comparison of the three FT variants against the baseline classifiers indicates that the least performance evaluation in Table 2, recorded against the FT-3 variant (Functional Tree with inner nodes) outperforms the most performance evaluation record (Dec Table) under the baseline column of Dataset 2.

Table 3 shows a slight difference in comparison, considering established interpretations in Table 2. This difference is connected to the earlier stated difference of Dataset 3 from other Datasets. Therefore, the least performance FT variant in Dataset 2; that is FT-3 scores the most performance records in Dataset 3, specifically because of the least performance recorded against it (i.e FP-Rate 0.071). This suggests that the FT-3 variant performs better than other FT variants on small dataset than large datasets in terms of the number of instances of incorrectly identified phishing attacks. However, the overall rate of instances of correctly predicted labels is observed with the FT-2 variant (90.24% Accuracy). The same best performance is observed under other performance metrics of FT-2, indicating

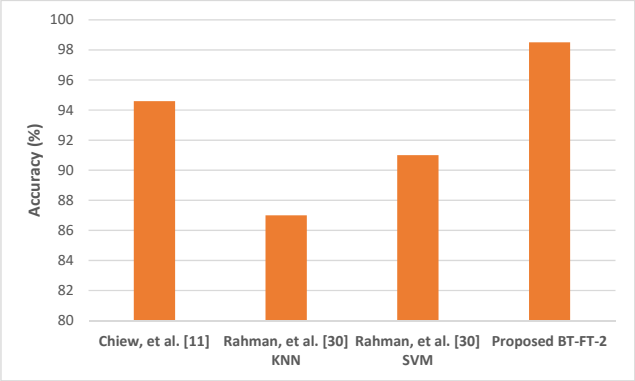

Figure 13. Comparison of BT-FT-2 with existing methods on Dataset 2.

that this Functional Tree (FT-2) still restricts the selection of test attributes to the original attributes despite the size of the dataset. In general, FT-2 recorded best performance of Accuracy 90.24%, F-measure 0.903, AUC 0.970, TP-Rate 0.902, and MCC 0.826. However, a broad comparison of these FT variants against the baseline classifiers shows that the latter leaves more to be desired, in terms of phishing detection performance, particularly at the FP-Rate performance evaluation matric.

Figures 7, 8, and 9 show the comparison of the results of FT-2 with other baseline classifiers on Dataset 1, Dataset 2 and Dataset 3 respectively.

4.2. Comparison of FT variants based on meta-learners

This section compares the results of the FT variants algorithms when incorporated with meta-learners. Results were obtained for each dataset based on the FT variants (FT-1, FT-2 and FT-3) and the three different types of meta-learners considered in this study. Rotation Forest (RoF) with the three variants of FT algorithms, namely, RoF-FT-1, RoF-FT-2 and RoF-FT-3, present the results obtained when the Rotation Forest meta-learner was used in combination with the variants of FT as base classifiers. Similarly, Bagging (BG) with the three variants of FT algorithms, namely, BG-FT-1, BG-FT-2 and BG-FT-3, present the results obtained when the Bootstrap Aggregating (Bagging) method for meta-learner was used in combination with the variants of FT as base classifiers. Lastly, Boosting (BT) with the three variants of FT algorithms, namely, BT-FT-1, BT-FT-2 and BT-FT-3, present the results obtained

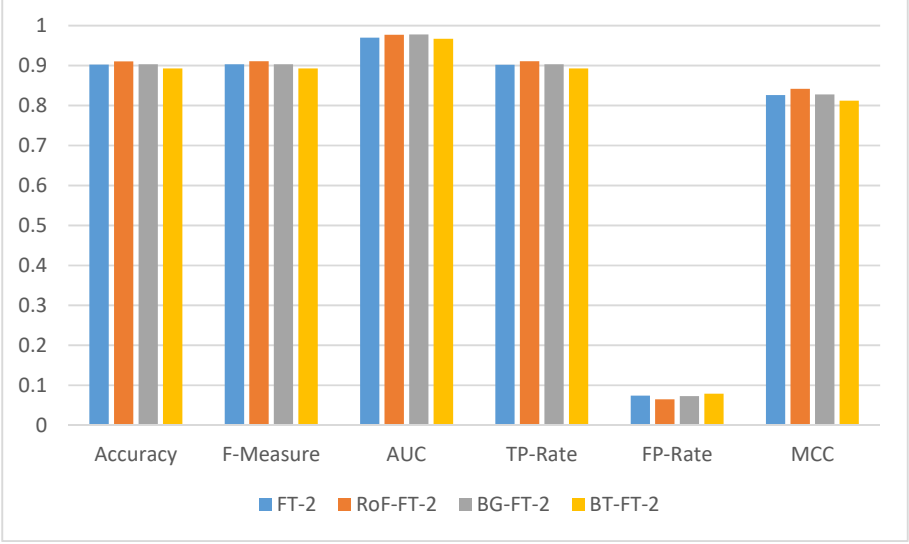

Figure 12. Performance comparison of FT-2 variant as a base classifier for meta-learners on Dataset 3.

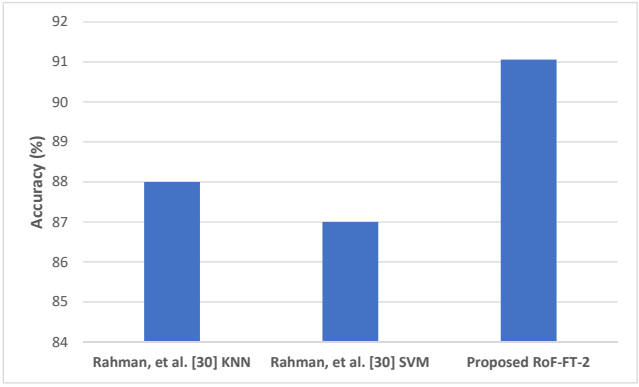

Figure 14. Comparison of RoF-FT-2 with existing methods on Dataset 3.

when Boosting (AdaBoost) meta-learner was used in combination with the variants of FT as base classifiers. The details of the results are discussed as follows:

As shown in Table 4, a slight difference in comparison was observed, considering the AUC evaluation metric. As observed, RoF-FT-2 and BT-FT-1 achieved the same AUC value as compared with other approaches. This result (i.e AUC of 0.996) is slightly higher than the AUC value obtained with BT-FT-2, which produced an AUC of 0.995. This result is very much at par when compared with that of RoF-FT-2 and BT-FT-1. However, in all other cases, considering other evaluation metrics, BT-FT-2 achieved promising improvement when compared with other methods. The implication of this is that FT variant (FT-2) when combined with Boosting (BT) meta-learner, achieved significant performance. This achievement can link to the consistently better performance of the FT-2 variant in the previously discussed results. BT-FT-2 produced Accuracy (97.19%, F-Measure (0.972), TP-Rate (0.972), FP-Rate (0.031) and MCC (0.943), which appeared to be the best results as shown in Table 4 based on Dataset 1. The result obtained according to FP-Rate shows that there are least instances of incorrectly identified phishing attacks under the FT-2 variant combined with Boosting meta-learner. Figure 10 shows the comparison of the results based on the FT-2 variant as a base classifier for meta-learners for phishing detection using Dataset 1.

Table 5 shows similar performance in BT-FT-2 on Dataset 2 when compared with what was obtained in Table 4. However, in all cases, considering the evaluation metrics used, except in the case of AUC where BT-FT-2 and RoF-FT-2 produced the same result (0.998), BT-FT-2 outperformed other meta-learners. This means that the consistency in the performance of the FT-2 variant still influenced the results obtained with the meta-learners when applied to classify phishing website as contained in Dataset 2. According to Table 5, BT-FT-2 produced Accuracy (98.51%), F-Measure (0.985), AUC (0.998), TP-Rate (0.985), FP-Rate (0.015) and MCC (0.970), achieving the best performance among the different types of meta-learners considered in this study. Figure 11 shows the comparison of the results based on the FT-2 variant as a base classifier for meta-learners for phishing detection using Dataset 2.

Table 6 presents the results of the phishing website detection process using FT variants as a base classifier for the three meta-learners on Dataset 3. As explained in the previous section, it was evident that Dataset 3 has the least number of instances amongst the three datasets employed in this study. However, the results presented above obviously revealed that some of the meta-learners-based classifiers accomplished a promising result from the experiments conducted. In particular, RoF-FT-2 has the best performance compare with others. Based on the performance evaluation metrics employed in this study, the RoF-FT-2 model achieved an accuracy of 91.06%, with its F-Measure, AUC, TP-Rate, FP-Rate and MCC having 0.911, 0.977, 0.911, 0.065 and 0.842 respectively. Figure 12 shows the comparison of the results based on the FT-2 variant as a base classifier for meta-learners for phishing detection using Dataset 3.

4.3. Comparison of proposed approaches with existing methods

To evaluate the performance of the proposed models, assessments with the existing state-of-art techniques are necessary. As discussed in Section 2, several models such as ANN [6], SMO [35], Stacking [60], SVM [34], KNN [34], Hybrid NBTree [60] etc., have been proposed. However, from Table 7, it is evident that the proposed method (BT-FT-2) shows noteworthy improvements in almost all evaluation metrics on Dataset 1. Compared with the existing literature, the proposed model outperforms them with an accuracy of 97.19%, thereby guaranteeing its suitability for detecting phishing websites.

Table 8 shows the comparison of experimental results of the proposed methods in this paper, which are indicated with an asterisk (*) with existing studies as presented in [12, 34]. More specifically, we compared the results of the nine (9) proposed meta-learners' models on Dataset 2. The results show that our proposed Boosting model (BT) with FT-2 variant, namely BT-FT-2 outperformed other approaches including the eight (8) meta-learners and the models proposed in the existing studies. Empirically, BT-FT-2 produced Accuracy (98.51%), F-Measure (0.985), AUC (0.998), TPR (0.985), FPR (0.015) and MCC (0.970). These results are better than the models in [11, 30] as shown in the table when considering all the evaluation metrics. Figure 13 shows the performance of BT-FT-2 with the models in [12, 34] in terms of the accuracy metric. This figure further clarifies the superiority of the proposed meta-leaners as compared with existing studies for phishing website detection. Also, it can be observed that other proposed methods (aside from BT-FT-2) performed comparable well against existing models in [12, 34]. Hence, it can be concluded that the proposed methods are in most cases better and as good as existing models on Dataset 2.

Similarly, Table 9 presents the comparison of the experimental results of the proposed methods with existing studies as presented in [34]. As shown in this table, the proposed approach with Rotation Forest (Ro) with FT-2 variant, namely RoF-FT-2 outperformed other models. RoF-FT-2 produced 91.06%, 0.911, 0.977, 0.911, 0.065 and 0.842 which correspond to Accuracy, F-Measure, AUC, TPR, FPR and MCC respectively. This result also outperformed other meta-learners on Dataset 3 as presented in Table 9. Furthermore, although the proposed methods outperformed the models presented in [34] on studied evaluation metrics, it can be observed that their respective performance is comparable and the differences in their respective performances are in most cases insignificant on Dataset 3. Hence, these results imply that the proposed approaches further demonstrate the superiority of the meta-learners as compared with baseline models in existing studies. The meta-learners proposed in this study are capable of separating phishing websites from legitimate websites with a high level of accuracy and reduced error rate. Figure 14 shows the performance of RoF-FT-2 as compared with the models in [34] based on accuracy values on Dataset 3.

To provide answers to the research questions (RQ) raised in Section 1 (Introduction), the following conclusions were drawn based on the experimental results obtained:

RQ1: How effective are FT algorithm implementations for detecting phishing and legitimate websites?

FT algorithm implementations indeed produced significant improvement as compared with baseline methods such as NB, SVM, SMO and DecTable. Specifically, the FT-2 variant as depicted in Section 3 of this paper outperformed other variants with a high degree of accuracy and low error rate. This superior performance is observed across the three datasets that were considered in this study.

RQ2: How effective is the Meta-learners (Bagged-FT, Boosted-FT, and Rotation forest-FT) in detecting phishing and legitimate websites?

The use of FT variant algorithms as a base learner for the proposed meta-learners such as Bagging (BG), Boosting (BT) and Rotation forest (RoF) significantly improved the performance of the FT variants. The most prominent and outstanding performance was witnessed when the FT-2 variant was implemented as the base learner for the meta-learners. BT and RoF meta-learners outperformed their BG counterpart. Indeed, the BT-FT-2 meta-learner produced the best result in terms of overall

performance across the three datasets considered in this study. The model is recommended for detecting phishing websites, which guaranteed a significant reduction in misclassification.

RQ3: How well is the performance of the proposed FT and its variants compared with existing state-of-the-art methods?

To answer this research question, the implemented FT along with the meta-learners were compared with existing studies. The results have proven the superiority of the proposed methods in this paper with existing state-of-the-art methods.

5. Conclusion

This study aimed at investigating the effectiveness of FT and its meta-learner variants for phishing website detection. Specifically, three different variants of FT (FT-1, FT-2, and FT-3) as defined in Section 3.1 alongside their meta-learners' based on Bagging, Boosting and Rotation Forest were experimented and analysed for phishing website detection. Findings from the experimental results showed that in most cases FT and its variants are superior to some existing phishing website detection models. More importantly, the proposed methods in this study demonstrated the power of meta-learners as an intelligent algorithm for designing models capable of detecting phishing websites more accurately and consistently. One of the proposed methods (BT-FT-2) achieved an exceptionally high predictive accuracy of approximately 99 per cent, as well as a low false-positive rate of 0.015 and a high MCC value of 0.97. These findings demonstrate the efficacy and reliability of the proposed methods, which in most cases have low false alarm rates while maintaining good detection accuracy.

As a limitation, we intend to deploy the proposed methods on real-time datasets. This is to ascertain the generalizability of the proposed methods to reduce the impact of phishing website on the internet. Additionally, we intend to investigate the effect of data quality problems such as class imbalance and high dimensionality problems on the detection of phishing websites.

Declarations

Author contribution statement

A. O. Balogun: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

K. S. Adewole: Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

M. O. Raheem, O. N. Akande, F. E. Usman-Hamza, M. A. Mabayoje, A. G. Akintola, A. W. Asaju-Gbolagade, M. K. Jimoh, R. G. Jimoh, V. E. Adeyemo: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability statement

Data included in article/supplementary material/referenced in article.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

Acknowledgements

The authors appreciate Landmark University Centre of Research and Innovations (LUCRID) for their support in publishing the outputs of this research.

References

[1] J. Jang-Jaccard, S. Nepal, A survey of emerging threats in cybersecurity, J. Comput. Syst. Sci. 80 (5) (2014) 973–993.
[2] K.S. Adewole, A.G. Akintola, S.A. Salihu, N. Faruk, R.G. Jimoh, Hybrid rule-based model for phishing URLs detection, in: Presented at the International Conference for Emerging Technologies in Computing, 2019.
[3] V.E. Adeyemo, A. Azween, N. JhanJhi, S. Mahadevan, A.O. Balogun, Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: an empirical study, Int. J. Adv. Comput. Sci. Appl. 10 (9) (2019) 520–528.
[4] M.D. Abdulrahaman, J.K. Alhassan, O.S. Adebayo, J.A. Ojeniyi, M. Olalere, Phishing attack detection based on random forest with wrapper feature selection method, Int. J. Infor. Proc. Commun. (IJIPC) 7 (2) (2019) 209–224.
[5] M. Adil, R. Khan, M.A.N.U. Ghani, Preventive techniques of phishing attacks in networks, in: Presented at the 2020 3rd International Conference on Advancements in Computational Sciences (ICACS), 2020.
[6] R.P. Ferreira, et al., Artificial neural network for websites classification with phishing characteristics, Soc. Netw. 7 (2) (2018) 97.
[7] G.K. Soon, L.C. Chiang, C.K. On, N.M. Rusli, T.S. Fun, Comparison of ensemble simple feedforward neural network and deep learning neural network on phishing detection, in: Computational Science and Technology, Springer, 2020, pp. 595–604.
[8] B. Wei, et al., A deep-learning-driven light-weight phishing detection sensor, Sensors 19 (19) (2019) 4258.
[9] G. Vrbančič, I. Fister Jr., V. Podgorelec, Swarm intelligence approaches for parameter setting of deep learning neural network: case study on phishing websites classification, in: Presented at the Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, 2018.
[10] A. AlEroud, G. Karabatis, Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks, in: Presented at the Proceedings of the Sixth International Workshop on Security and Privacy Analytics, 2020.
[11] Y.A. Alsariera, A.V. Elijah, A.O. Balogun, Phishing website detection: forest by penalizing attributes algorithm and its enhanced variations, Arabian J. Sci. Eng. (2020) 1–12.
[12] K.L. Chiew, C.L. Tan, K. Wong, K.S. Yong, W.K. Tiong, A new hybrid ensemble feature selection framework for machine learning-based phishing detection system, Inf. Sci. 484 (2019) 153–166.
[13] C.L. Tan, K.L. Chiew, K.S. Yong, J. Abdullah, Y. Sebastian, A graph-theoretic approach for the detection of phishing webpages, Comput. Secur. (2020) 101793.
[14] W. Ali, A.A. Ahmed, Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting, IET Inf. Secur. 13 (6) (2019) 659–669.
[15] P. Yang, G. Zhao, P. Zeng, Phishing website detection based on multidimensional features driven by deep learning, IEEE Access 7 (2019) 15196–15209.
[16] A. Zamir, et al., Phishing Web Site Detection Using Diverse Machine Learning Algorithms, The Electronic Library, 2020.
[17] E. Zhu, Y. Ju, Z. Chen, F. Liu, X. Fang, DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features, Appl. Soft Comput. (2020) 106505.
[18] B.B. Gupta, N.A. Arachchilage, K.E. Psannis, Defending against phishing attacks: taxonomy of methods, current issues and future directions, Telecommun. Syst. 67 (2) (2018) 247–267.
[19] I. Ghafir, V. Prenosil, Blacklist-based malicious ip traffic detection, in: Presented at the 2015 Global Conference on Communication Technologies (GCCT), 2015.
[20] V.E. Urias, W.M. Stout, J. Luc-Watson, C. Grim, L. Liebrock, M. Merza, Technologies to enable cyber deception, in: 2017 International Carnahan Conference on Security Technology (ICCST), IEEE, 2017, pp. 1–6.
[21] G. Harinahalli Lokesh, G. BoreGowda, Phishing website detection based on effective machine learning approach, J. Cyber Sec. Technol. (2020) 1–14.
[22] Y.A. Alsariera, V.E. Adeyemo, A.O. Balogun, A.K. Alazzawi, Ai meta-learners and extra-trees algorithm for the detection of phishing websites, IEEE Access 8 (2020) 142532–142542.
[23] Y. Chandra, A. Jana, Improvement in phishing websites detection using meta classifiers, in: Presented at the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), 2019.
[24] M. Dedakia, K. Mistry, Phishing detection using content based associative classification data mining, J. Eng. Comput. Appl. Sci. (JECAS) 4 (7) (2015) 209–214.
[25] A.O. Balogun, S. Basri, S.J. Abdulkadir, V.E. Adeyemo, A.A. Imam, A.O. Bajeh, Software defect prediction: analysis of class imbalance and performance stability, J. Eng. Sci. Technol. 14 (6) (2019) 3294–3308.
[26] A.O. Balogun, S. Basri, S.J. Abdulkadir, A.S. Hashim, Performance analysis of feature selection methods in software defect prediction: a search method approach, Appl. Sci. 9 (13) (2019) 2764.

[27] A.O. Balogun, et al., Impact of feature selection methods on the predictive performance of software defect prediction models: an extensive empirical study, Symmetry 12 (7) (2020) 1147.
[28] A. Basit, M. Zafar, X. Liu, A.R. Javed, Z. Jalil, K. Kifayat, A comprehensive survey of AI-enabled phishing attacks detection techniques, Telecommun. Syst. (2020) 1–16.
[29] R.M. Mohammad, F. Thabtah, L. McCluskey, Predicting phishing websites based on self-structuring neural network, Neural Comput. Appl. 25 (2) (2014) 443–458.
[30] R. Verma, A. Das, What's in a url: fast feature extraction and malicious url detection, in: Presented at the Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics, 2017.
[31] M. Alqahtani, Phishing websites classification using association classification (PWCAC), in: 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, 2019, pp. 1–6.
[32] N. Abdelhamid, A. Ayesh, F. Thabtah, Phishing detection based associative classification data mining, Expert Syst. Appl. 41 (13) (2014) 5948–5959.
[33] Abikoye Oluwakemi Christiana, Haruna Ahmad Dokoro, Abdullahi Abubakar, Akande Noah Oluwatobi, Asani Emmanuel Oluwatobi, Modified advanced encryption standard algorithm for information security, Symmetry 11 (12) (2019) 1–17.
[34] S.S.M.M. Rahman, F.B. Rafiq, T.R. Toma, S.S. Hossain, K.B.B. Biplob, Performance assessment of multiple machine learning classifiers for detecting the phishing URLs, in: Data Engineering and Communication Technology, Springer, 2020, pp. 285–296.
[35] M. Aydin, N. Baykal, Feature extraction and classification phishing websites based on URL, in: Presented at the 2015 IEEE Conference on Communications and Network Security (CNS), 2015.
[36] A.A. Ubing, S.K.B. Jasmi, A. Abdullah, N. Jhanjhi, M. Supramaniam, Phishing website detection: an improved accuracy through feature selection and ensemble learning, Int. J. Adv. Comput. Sci. Appl. 10 (1) (2019) 252–257.
[37] J. Gama, Functional trees, Mach. Learn. 55 (3) (2004) 219–250.
[38] B.T. Pham, V.-T. Nguyen, V.-L. Ngo, P.T. Trinh, H.T.T. Ngo, D.T. Bui, A novel hybrid model of rotation forest based functional trees for landslide susceptibility mapping: a case study at Kon Tum Province, Vietnam, in: International Conference on Geo-Spatial Technologies and Earth Resources, Springer, 2017, pp. 186–201.
[39] I.H. Witten, E. Frank, Data mining: practical machine learning tools and techniques with Java implementations, Acm Sigmod Rec. 31 (1) (2002) 76–77.
[40] N. Landwehr, M. Hall, E. Frank, Logistic model trees, Mach. Learn. 59 (1-2) (2005) 161–205.
[41] Abikoye Oluwakemi Christiana, Benjamin Aruwa Gyunka, Akande Noah Oluwatobi, Optimizing android malware detection via ensemble learning, Int. J. Inter. Mob. Technol. (iJIM) 14 (9) (2020) 61–78.
[42] E. Frank, Y. Wang, S. Inglis, G. Holmes, I.H. Witten, Using model trees for classification, Mach. Learn. 32 (1) (1998) 63–76.
[43] G. Collell, D. Prelec, K.R. Patil, A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data, Neurocomputing 275 (2018) 330–340.
[44] P. Bühlmann, Bagging, boosting and ensemble methods, in: Handbook of Computational Statistics, Springer, 2012, pp. 985–1022.
[45] V.B. Vaghela, A. Ganatra, A. Thakkar, Boost a weak learner to a strong learner using ensemble system approach, in: 2009 IEEE International Advance Computing Conference, IEEE, 2009, pp. 1432–1436.
[46] B. Sun, S. Chen, J. Wang, H. Chen, A robust multi-class AdaBoost algorithm for mislabeled noisy data, Knowl. Base Syst. 102 (2016) 87–102.
[47] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.
[48] E. Tasci, A meta-ensemble classifier approach: random rotation forest, Balkan J. Electr. Comp. Eng. 7 (2) (2019) 182–187.
[49] A.O. Balogun, A.O. Bajeh, V.A. Orie, W.A. Yusuf-Asaju, Software defect prediction using ensemble learning: an ANP based evaluation method, FUOYE J. Eng. Technol. 3 (2) (2018) 50–55.
[50] R. Jimoh, A. Balogun, A. Bajeh, S. Ajayi, A PROMETHEE based evaluation of software defect predictors, J. Comp. Sci. Appl. 25 (1) (2018) 106–119.
[51] Z. Xu, J. Liu, Z. Yang, G. An, X. Jia, The impact of feature selection on defect prediction performance: an empirical comparison, in: Presented at the 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), 2016.
[52] Q. Yu, S. Jiang, Y. Zhang, The performance stability of defect prediction models with class imbalance: an empirical study, IEICE Trans. Info Syst. 100 (2) (2017) 265–272.
[53] A.O. Balogun, et al., SMOTE-based homogeneous ensemble methods for software defect prediction, in: International Conference on Computational Science and its Applications, Springer, 2020, pp. 615–631.
[54] S. Yadav, S. Shukla, Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification, in: Presented at the 2016 IEEE 6th International Conference on Advanced Computing (IACC), 2016.
[55] Abikoye Oluwakemi Christiana, Abubakar Abdullahi, Ahmed Haruna Dokoro, Akande Noah Oluwatobi, Kayode anthonia aderonke, A novel technique to prevent SQL-injection and cross-site scripting attacks using Knuth-Morris-Pratt string matching algorithm, EURASIP J. Inf. Secur. 14 (2020) 1–14.
[56] A.O. Balogun, et al., Search-based wrapper feature selection methods in software defect prediction: an empirical analysis, in: Computer Science On-Line Conference, Springer, 2020, pp. 492–503.
[57] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. NewsLett. 11 (1) (2009) 10–18.
[58] V.E. Adeyemo, A.O. Balogun, H.A. Mojeed, N.O. Akande, K.S. Adewole, Ensemble-based logistic model trees for website phishing detection, in: International Conference on Advances in Cyber Security, Springer, 2020, pp. 627–641.
[59] L. Aljerf, I. Alhaffar, Salivary distinctiveness and modifications in males with diabetes and Behçet's disease, Biochem. Res. Inter. 2017 (2017).
[60] S.O. Folorunso, F.E. Ayo, K.-K.A. Abdullah, P.I. Ogunyinka, Hybrid vs ensemble classification models for phishing websites, Iraqi J. Sci. (2020) 3387–3396.
[61] S. Al-Ahmadi, T. Lasloum, PDMLP: phishing detection using multilayer perceptron, Int. J. Netw. Secur. Appl. 12 (3) (2020) 59–72.
[62] W. Ali, S. Malebary, Particle swarm optimization-based feature weighting for improving intelligent phishing website detection, IEEE Access 8 (2020) 116766–116780.