Journal of Theoretical and Applied Information Technology
15th March 2019. Vol.97. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A COMPARATIVE ANALYSIS OF PHISHING WEBSITE DETECTION USING XGBOOST ALGORITHM

[1]HAJARA MUSA, [2]DR. A.Y GITAL, [3]F. U. ZAMBUK, [4]ABUBAKAR UMAR,
[5]AISHATU YAHYA UMAR, [6] JAMILU USMAN WAZIRI

[1 & 5] Department of Mathematical Sciences, Gombe State University, Gombe, Nigeria
[2,3,& 4] Department of Mathematical Sciences, Abubakar Tafawa Balewa University Bauchi, Nigeria
[6] Department of Mathematical Sciences, Federal University Gusau, Nigeria

E-mail: [1]mhajara86@gmail.com, [2]asgital@gmail.com
[3]fzambuk2001@yahoo.com, [4]abkamir@gmail.com
[5]aishatuyu@gmail.com, [6] jamiluwaziri@gmail.com

ABSTRACT

As most of human activities are being moved to cyberspace, phishers and other cybercriminals are making the cyberspace unsafe by causing serious risks to users and businesses as well as threatening global security and economy. Nowadays, phishers are constantly evolving new methods for luring user to reveal their sensitive information. To avoid falling victim to cybercriminals, a phishing detection algorithms is very necessary to be developed. Machine learning or data mining algorithms are used for phishing detection such as classification that categorized cyber users in to either malicious or safe users or regression that predicts the chance of being attacked by some cybercriminals in a given period of time. Many techniques have been proposed in the past for phishing detection but due to dynamic nature of some of the many phishing strategies employed by the cybercriminals, the quest for better solution is still on. In this paper, we propose a new phishing detection model based on Extreme Gradient Boosted Tree (XGBOOST) algorithm. Experimental results demonstrated that XGBOOST-based phishing detection model is promising by returning an accuracy of 97.27% which outperformed both probabilistic Neural Network (PNN) and Random forest (RF) that returned accuracies of 96.79% and 95.66% respectively.
Keyword: *Machine Learning, Feature Selection, Classification, XGBOOST, Phishing.*

1. INTRODUCTION

The word phishing was first coined in 1996 as a form of online identity theft after an attack by hackers on AmericaOnline account and the first phishing lawsuit was filed in 2004 against a California teenager who created an imitation of the website "AmericaOnline" to gain access to user sensitive information including credit card details causing them huge financial lost [1].

Phishing is a cyber-crime which involves the fraudulent act of illegally capturing private information like credit card details, usernames, password, account information by pretending to be authentic and esteemed in instant messaging, email and various other communication channels. The traditional approaches used by majority of the email filters for identifying these emails are static which make it weak to deal with latest developing patterns of phishing since the defrauders are dynamic in actions and keep on modifying their activities to dodge any kind of detection [2].

Phishers are sending fake emails to their victims pretending to be from legitimate and well known organizations such as banks, university, communication network etc., where they will require updating some personal information including their passwords and usernames to avoid losing access right to some of the services provided by that organization. Phishers use this avenue to obtain users sensitive information which they in turn use it to access their important accounts resulting in identity theft and financial loss [3].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Many approaches have been proposed in an attempt to curb the problems caused by phishers [3-7]. To further investigate the problem of phishing [7] proposed novels approach for detecting phishing websites based on probabilistic neural networks (PNNs). However, due to the dynamic nature of the attacks and the challenging nature of the problem, it still lacks a complete solution. This paper proposed Extreme Gradient Boosted trees (XGBOOST) to improve the performance that a predictive model that can achieve in the detection of a phishing website from a legitimate website. We evaluate its performance measures on a publicly available dataset obtained from UCI machine learning repository which contained 2456 websites instances preclassified as benign (non phishing) and phishing websites with 30 features and compared the results obtained with that of Probabilistic Neural Networks (PNN) and Random forest (RF) methods

.

2. RELATED WORKS

A good number of recent research papers related to Phishing detection are reviewed in order to achieve our goal, [4] proposed anti-phishing detection of phishing attacks using Genetic Algorithm (GA) to evolve rules that are used to differentiate phishing link from legitimate link. Their Experiments shows that, the approach is effective to detect phishing hyperlink with minimal false negatives at a speed adequate for online application, but the genetic parameter leads to more complex algorithm due to the fine tuning of all parameters for G.A. [8] proposed an approach to automatic identification of the phishing target of a given webpage by clustering the webpage set consisting of all its associated webpages and the given webpage itself. Their Experiments show that the approach can successfully identify 91.44% of their phishing targets. But it is difficult to identify the initial cluster. [9] proposed CANTINA+, the most comprehensive feature-based approach which include eight novel features. [5] proposed an intelligent anti phishing strategy model for phishing website detection using Hierarchical clustering technique and categorization through learning and training samples from large and real daily phishing websites collected from Kingsoft Internet Security Lab. Experiments on real life datasets demonstrate that the method outperforms existing popular detection methods and commonly used anti-phishing tools in phishing detection. But using hierarchical clustering algorithms, it is sometimes difficult to identify the correct number of cluster. [10] proposed the study of new inputs which were not considered previously in a single protection platform. The idea is to utilize a Neuro-Fuzzy Scheme with 5 inputs to detect phishing sites with high accuracy in real-time. The main challenge on using Neuro-Fuzzy Inference System is that it is much complex, specifically, it must have a single output obtained using weighted average defuzzification. Also all output membership functions must be the same type, either be linear or constant.

[6] proposed an intelligent model for predicting phishing attacks based on Artificial Neural Network (ANN) particularly self-structuring neural networks. The model solves problem by automating the process of structuring the network and shows high acceptance for noisy data, fault tolerance and high prediction accuracy. Experiments were conducted in the research. The results shows that, all produced structures have high generalization ability. [11] proposed a new phishing webpage detection approach based transductive support vector machine (TSVM). The features of sensitive information are examined by using page analysis based on DOM objects. The method introduces the TSVM to train classifier that it takes into account the distribution information implicitly embodied in the large quantity of the unlabeled samples, and have better performance than SVM. The experimental result shows that the proposed method not only achieves better classification accuracy, but also has strong applicability as the independent method of phishing detection. This approach has been observed to overfit for some datasets with noisy classification tasks. [3] investigated the problem of website phishing using a developed AC method called Multi-label Classifier based Associative Classification (MCAC) to seek its applicability to the phishing problem. They also want to identify features that distinguish phishing websites from legitimate ones. Experimental results

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

using real data collected from different sources show that AC particularly MCAC detects phishing websites with higher accuracy than other intelligent algorithms. The problem of the approach is that, many algorithms suffer from defects to varying degrees. It is obviously imperative to achieve correct prediction but also equally or perhaps more important to avoid false and potentially misleading ones. [12] Proposed a hybrid model to classify phishing emails using machine learning algorithms with the aspiration of developing an ensemble model for email classification with improved accuracy. They have used the content of emails and extracted 47 features from it. Going through experiments, it is observed and inferred that Bayesian net classification model when ensemble with CART gives highest test accuracy of 99.32%. The approach creates over-complex trees that do not generalize the data well (overfitting).

[13] proposed heuristic-based phishing detection technique that employs URL-based features. The system first extracts the features which clearly differentiate that whether website are phished or legitimate. The experiment shows that SVM has accuracy of 96% and very low false-positive rate. The proposed model can reduce damage caused by phishing attacks because it can detect new and temporary phishing sites. Heuristic evaluation does not allow a way to assess the quality of redesigns. [14] compared different features assessment techniques in the website phishing context in order to determine the minimal set of features for detecting phishing activities. Experimental results on real phishing datasets consisting of 30 features has been conducted using three known features selection methods. Their approach can be hard to find a usable formal representation and it deals badly with quantitative measurements. The emails have been classified as phish using the prediction of Ensemble Classifier of the five ML Algorithms [2]. Experiment shows that the comparison of the accuracy of algorithms for Different Feature Groups based on the decisive values of the features demonstrated that best accuracy is obtained for Random Forest by 96.07%. Random forests have been observed to overfit for some datasets with noisy classification tasks. The evaluation of model size is slow because it could easily end up with a forest that takes hundreds of megabytes of memory [7]. In their work, they presented a novel approach for detecting phishing websites based on probabilistic neural networks (PNNs). They tried to investigate the integration of PNN with K-medoids clustering to significantly reduce complexity without jeopardizing the detection accuracy. The experimental results show that 96.79% accuracy is achieved with low false errors. But their approach requires large memory spaces to store and the execution of network of this approach is slow.

Phishing is a continuous problem. Thus, there is a need to constantly improve the network structure in order to cope with these changes [6] and hence the quest for a better solution is still on. In recent time, machine learning techniques have been found to be very successful in phishing website detection [15][16]. This research proposes XGBOOST algorithm to improve the performance that a predictive model can achieve in the task of phishing website detection. Advantages of XGBOOST have made it an excellent tool of choice for many researchers in data science and machine learning. In light of the above, XGBOOST has been recently employed in many machine learning task with great success [17-19].

3. RESEARCH FRAMEWORK.

In order to achieve the goal of this research, we followed the steps; designed in the framework as shown in Figure1.0

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

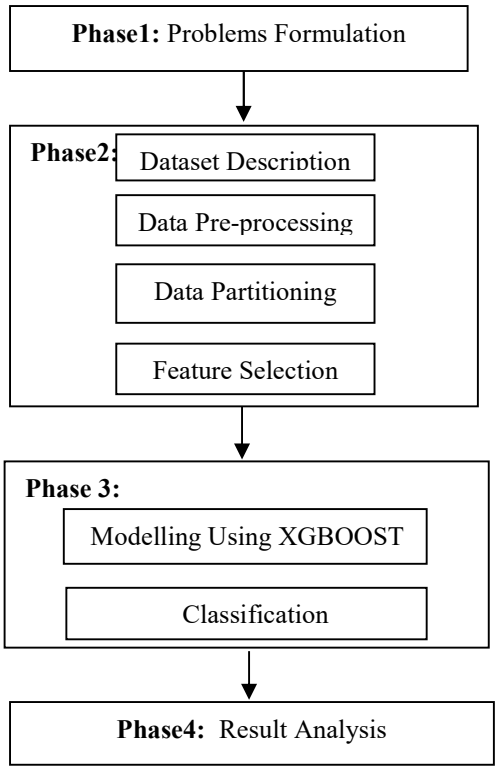

Figure1 .0: Research framework

Figure 1.0 illustrate the stages followed in this research where some selected papers were reviewed in order to determine the research gap and as such the research problem was formulated; special attention was given to feature selection, phishing website detection, classification, and clustering. Others includes; random forest, probabilistic neutral network, k-nearest neighbor, naive bayes and extreme gradient boosted tree techniques as these are the constituents of this research work. Special attention was also given to data pre-processing being a crucial stage in classification task. It is interested to know that, most researchers in phishing detection make use of datasets constructed by them. However, with such type of datasets, it is difficult to evaluate and compare the performance of a model with other models from the literature since the datasets used were not publicly available for others to use and confirm their results, therefore such results cannot be generalized [7].

3.1 *Dataset Description*

In order to assess and compare the predictive performance of our proposed model, we employed a recently created phishing websites dataset from UCI machine learning repository. This dataset was created by Mohammmed, Thabtah and McChushy at the university of Huddesfied, united Kingdom and denoted to UCI machine learning repository in 2015. The dataset has a total of 2456 websites instances preclassified as benign (non phishing) and phishing websites with 30 features. Each website is converted into a vector x = (x,x,x) where x are the values corresponding to specific feature (variable) of a particular website. features in datatset are divided into four categories. The first category (f1 - f12) are the address bar based features, the second category (f13 -f18) are abnormal based features. The third category (f19 -f23) are html and javascript based features and the last category (f24-f30) are domain based features. In the value range column, a value of -1 means benign, 0 means suspicious and 1 means phishing.

4. PROPOSED ALGORITHM

XGBOOST (Extreme Gredient Boosted Tree) is an optimized implementation of gradient boosted trees first introduced by [20] . It is mostly employed in classification task where it is used as a classifier for mapping input pattern into a specific class. It is a recent supervised learning algorithm that implements a process known as boosting to improve the performance of gradient boosted trees. XGBOOST has many strengths when compared to the traditional gradient boosting implementations. Among its strengths are better regularization ability which helps to reduce overfitting, high speed and performance owing to the parallel nature in which trees are built, flexibility due to it costume optimization objectives and evaluation criteria, and inbuilt routines for handling missing values [21]. These and many other advantages of XGBOOST have made it an excellent tool of choice for many researchers in data science and machine learning. Some of the researchers employed this techniques [17][18][19]. ***Below is an algorithms for XGBOOST.***

1. *Procedure XGBOOST(X, y, l, f)*
2. *Input:*
3. *X: the training set |X|=[N,M]*
4. *y: the label*
5. *l: the lost function*
6. *f: the base model*

Journal of Theoretical and Applied Information Technology
15th March 2019. Vol.97. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7. *Steps:*
8. *Initialize:* $f_0(x) = 0$
9. *for k = 1 to t do*
10. *for i = 1 to N do*
11. $g_i = \partial_{\hat{y}^{(t-1)}}(\hat{y}^{(t-1)} - y_i)^2 = 2(\hat{y}^{(t-1)} - y_i$
12. $hi = \partial^2 y^{(t-1)} l(yi, \hat{y}i^{(t-1)}) = 2$
13. *end for*
14. *use gi, hi to compute objective function obj(t)*
15. *greedily grow a tree* $ft(x)$
16. $ft(x) = f_{(t-1)}(x) + \epsilon f_t(x)$
17. *end for*
18. *Output:*
19. $f(x) = f_t(x)$
20. *end procedure.*

5. CLASSIFICATION MODEL.

Phishing detection is a supervised learning problem where the training data xi was used to predict a target variable yi. The inputs to the phishing detection model are usually pairs of training instances (x_1,y_1), (x_2,y_2)…(xn,yn) where x is a vector of features extracted from a number of websites and y is their corresponding label which is either a 0 (benign) or 1 (phishing). In this research, our target is to determined some usefull parameters of the model using the available dataset so that at any given instance, the model can use those parameter to tell whether a new website is benign or phishing. Tree based models generally, do not have the same level of performance when compared with some other classification and regression techniques. Nonetheless, by combining many trees using technique like boosting, the predictive performance of trees can be improved subtstantially. XGBOOST is tree based model that aggregates trees using the boosting technique. In XGBOOST, the training data xi will be used to predict the target variable yi iteratively until the parameters of the model are optimized. Mathematically, the proposed phishing detection model can be represent as follows:

The *prediction model* ($\hat{y}$) can be written as the aggregation of all the prediction score for each tree for a *sample* (x). Particularly for *i*-th sample,

$$\hat{y}i = \sum_{k}^{K} f_k(x), f_k \in F \qquad eqn.\ (1)$$

Where K is the number of trees, f is the function in the functional space $\mathcal{F}$ and $\mathcal{F}$ is the all possible set of trees having prediction score in each leaf.

Boosted trees are trained via a strategy known as additive training. New tree is added at each iteration of the phishing detection process. The final prediction score of the model is obtained by summing the predictive score of individual tree.

The predictive value at step t of the training can be written as

$$\hat{y}_i^{(0)} = 0$$
$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$$
$$\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)$$
$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \qquad eqn.\ (2)$$

The newest tree is created to compensate for the instances of the websites wrongly predicted by the previous learners. We need to optimize certain objective function to choose the best model for the training data. Here, we encourage a model to have high predictive power as well as to have a simple in nature (deals with less number of features). As we know minimizing loss function ((Θ)) encourages predictive models as well as optimizing regularization (Ω(Θ)) encourages simpler model to have smaller variance in future predictions, making prediction stable (Chen, 2014). The closed form of the objective is given below:

$$obj(\theta) = l(\theta) + \Omega(\theta)$$
$$\theta = \{f_1, f_2, \ldots \ldots f_k\}$$
$$obj(\vartheta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{g} \Omega(f_k) \qquad eqn.\ (3)$$

XGBOOST executes t boosting iteration to learn a function f(x) that output the predictions y = f(x) minimizing a loss function and a regularization term. Similary, our optimization objective at step t of the training process can be formulated as:

$$obj^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_i) \qquad eqn.\ (4)$$

optimization objective using square loss can written as:

$$using\ square\ loss, the\ loss\ function\ \ l = (y_i - \hat{y}_i^{(t)})^2$$
$$but\ \hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$obj^{(t)} = \sum_{i=1}^{n} (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^{t} \Omega(f_i)$$

$$obj^{(t)} = \sum_{i=1}^{n} [2(\hat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant \quad eqn.(5)$$

While Using Taylor expansion,

$$obj^{(t)} = \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

Objective, with constants removed, therefore the new form of optimizing goal is:

$$obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad eqn.(6)$$

Where:

g_i and h_i comes from definition of loss function, the learning of function only depend on the objective via g_i and h_i

$$g_i = \partial_{\hat{y}_i^{(t)}} l(y_i, \hat{y}_i^{(t-1)}), \; g_i = \partial_{\hat{y}^{(t-1)}} (\hat{y}^{(t-1)} - y_i)^2 = 2(\hat{y}^{(t-1)} - y_i)$$

$$h_i = \partial^2_{\hat{y}_i^{(t)}} l(y_i, \hat{y}_i^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} (\hat{y}^{(t-1)} - y_i)^2 = 2$$

XGBOOST approximates f(x) by an additive expansion of t regression trees, but instead of minimizing just a lost function, an objective function with two parts is defined, a lost function over the training set as well as a regularization term to prevent overfitting. The objective function is formulated as in equation (5)

Where Loss function can be any convex differential loss function that measures the difference between the prediction and true label for a binary instance [20] . Ω (ft) is a regularization term which describe the complexity of the tree ft and is defined in the XGBOOST algorithm as

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2 \quad eqn.(7)$$

Where T is the number of leaves of tree ft

and ω are the leaf weights (i.e the predicted values at the leaf nodes).

γ and λ are constants, gamma and lamba are the Lagrangian multipliers and can be tuned for accuracy, that is user defined parameters .

XGBOOST uses a shrinkage parameter to reduce the optimal node predictions done in each iteration t before it add this prediction to the current functions ft. moreover, it uses row subsampling and column subsampling. The regularization fuction and these last three features of XGBOOST allows it to avoid overfitting [22].

To derive an expression for structure score substitute (6) in (5), the objective function can be re-written in terms of scores as:

$$obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

$$obj^{(t)} = \sum_{i=1}^{n} \left[g_i \omega_{q(x_i)} + \frac{1}{2} h_i \omega_{q(x_i)}^2 \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2$$

$$obj^{(t)} = \sum_{i=1}^{n} \left[\left(\sum_{i \in I_j} g_i \right) \omega_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) \omega_j^2 \right] + \gamma T \quad eqn.(8)$$

But

$G_j = \sum_{i \in I_j} g_i \qquad H_j = \sum_{i \in I_j} h_i$

The optimal score to optimize the objective function:

$$\omega_j^* = -\frac{G_j}{H_j + \lambda}$$

In this way, in each iteration, we are able to choose an optimized tree which optimizes the objective function which has been already optimized partly up to previous iteration, which ensures better accuracy. The optimal score is the best score function for a given structure of tree and *optimal objective reduction* measures how good is a tree structure for a particular iteration so that it could minimize the objective function which is given below.

$$obj^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \quad eqn.(9)$$

Due to impossibility of enumerating the entire tree from the function space, a greedy approach is of practical use which ensures *an optimal split*. The gain for a split can be formulated as:

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad eqn.(10)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The components are the score on the new leaf, the score on the new right leaf, the score on the original leaf and the complexity cost by introducing additional leaf. It is obvious that, if gain is smaller than γ, we would better not to add that branch, which is nothing but *pruning*.

The difference between Boosted Trees and Random Forest is how we train them. The major reason is in terms of training objective, Boosted Trees tries to add new trees (additive training) that complement the already built ones. This normally gives you better accuracy with fewer trees. In Random Forest the regularization factor is missing. But in Boosted trees, there is control on model complexity which reduces overfitting.

6. EVALUATION CRITERIA

To evaluate and compare the performance of our proposed model with other models from the literature, the following evaluation metrics were employed; accuracy (ACC) , precision (Prec), recall (Rec), mathew correlation coefficient (MCC), and f-score. ACC measures the ratio of websites which are correctly predicted. Prec measures the fraction of websites correctly predicted as phishing. Rec metric measures the fraction of phishing websites identified by the model.

Table 1.0 Confusion Matrix.

	Predicted Positive Class	Predicted Negative Class
Actual positive class	TP	FN
Actual negative class	FP	TN

Table 1.0 shows the confusion matrix in which TP (True Positive) is a case where a model correctly predicts a website as phishing, TN (True Negative) is a case where a website is wrongly classified as benign. FP (False Positive) is a case where a website is wrongly classified as phishing and lastly FN (False negative) is when the model wrongly classified a website as benign while it is actually phishing. The mathematical equations of the performance metrics are given below respectively.

$$ACC = \frac{TP+TN}{(TP+TN+FP+FN)} \quad \text{(i)}$$

$$Prec = \frac{TP}{(TP+FP)} \quad \text{(ii)}$$

$$Rec = \frac{TP}{(TP+FN)} \quad \text{(iii)}$$

$$F-score = \frac{2*(Prec*Rec)}{(Rec+Prec)} \quad \text{(iv)}$$

$$MCC = \frac{(TP*TN)-(FP*TN)}{Sqrt(TP+FP)(TP+FN)(TN+FP)(TN+FN)} \quad \text{(v)}$$

7. RESULT

The results were obtained after successful execution of the experiments. The results were analyzed and evaluated using the criteria stated earlier in the previous section. Finally a conclusion was reached on the performance of the proposed technique in comparison with other techniques reported in literature.

7.1 Experimental Result and Comparison.

This section present the result of the proposed model and that of the other methods used as benchmark. All the classifiers were implemented using the same performance metric for a fair comparison. Table 2.0 shows the comparison of results obtained from our experiment with that of PNN, RF, NB, and KNN classifiers.

Table 2.0 Comparison of Results

Methods	Precision	Recall	F. score	Mcc	Accuracy
XGBOOST	0.9730	0.9801	0.9724	0.9449	0.9729
PNN	0.9640	0.9789	0.9714	0.9350	0.9679
RF	0.9433	0.9796	0.9611	0.9128	0.9566
NB	0.9338	0.9504	0.9420	0.8679	0.9349
KNN	0.9147	0.9326	0.9236	0.8257	0.9141

The predictive performance of phishing website detection model using XGBOOST

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithms was optimized; thereby bringing the models much closer to accurate measurement of the performance metric than other models in which their models performance is less than our own in case of precision, recall, f.score, MCC and Accuracy. As we can see from Table 2.0 PNN has 0.9640, 0.9789, 0.9714, 0.9350 and 0.9679 accuracies respectively and RF has 0.9430, 0.9796, 0.9611, 0.9128, and 0.9566 accuracies respectively while XGBOOST returned the highest accuracies of 0.9730, 0.9801, 0.9724, 0.9449, and 0.9729 respectively. Therefore the proposed model has the highest accuracy compared with the rivals models. The result can be represented in graphical form for better analysis and understanding.

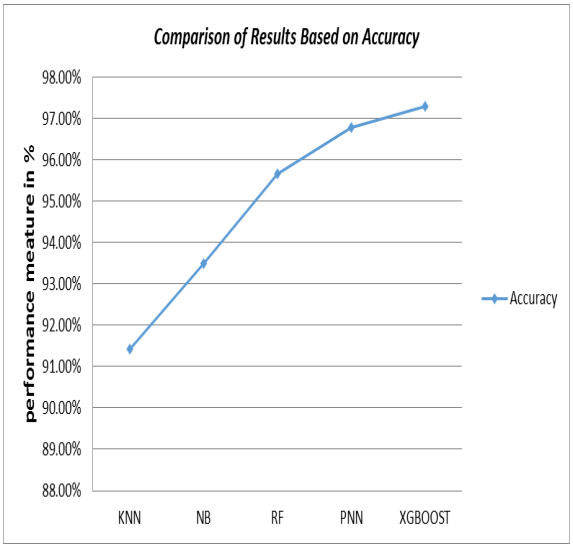

Figure 2.0 Accuracy Chart Representation of Models Comparison

Figure 2.0 shows that, the proposed model returned the best accuracy among all other the algorithms followed by PNN and RF. KNN obtained lowest values of the performance metric used, therefore KNN is least robust of the algorithms. From the experimental results, it can be observed that the proposed method is the most robust among the entire algorithm as it has the highest values in all the performance metric employed. This might be as a result of the technique the proposed method employs in avoiding overfitting. Knowing that the major problem of Random forests has been observed to overfit for some datasets with noisy classification tasks [23]. Therefore, the XGBOOST regularization term, rows subsampling, column subsampling and shrinkage parameters are techniques that allow XGBOOST to avoid overfitting. PNN also has the problem because, it requires large memory space to store and the execution of the network is slow, but XGBOOST has many advantages over the traditional gradient boosting implementations. Among the advantages are better regularization ability which helps to reduce overfitting, high speed and performance owing to the parallel nature in which trees are built, flexibility due to it costume optimization objectives and evaluation criteria, and inbuilt routines for handling missing values [21]. These and many other advantages of XGBOOST have made it an excellent tool of choice for many researchers in data science and machine learning. The fact is Random Forest and Boosted Trees are not different in terms of model, the difference is how we train them. The major reason is in terms of training objective, Boosted Trees tries to add new trees (additive training) that complement the already built ones. This normally gives you better accuracy with fewer trees. In Random Forest the regularization factor is missing. But in Boosted trees, there is control on model complexity which reduces overfitting.

8. CONCLUSOINS AND CONTRIBUTION TO KNOWLEDGE.

This research contributes to knowledge by adapting XGBOOST algorithm in phishing website detection; as well as the introductiuon new method of phishing detection.The experimental results have shown that, the proposed method is robust because it outperforms the PNN and RF in some of the problems considered. It is clearly showed that the predictive performance of phishing website detection model using XGBOOST algorithm is optimized to 97.29%.

In this research, the performance of the XGBOOST with Probabilistic Neural Networks (PNN) and Random forest (RF) method was compared in which all the methods (classifiers) were trained and tested using the same dataset and evaluated using the same performance metrics for a fair comparison.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] M. Khonj, Y. Iraqi and A. Jones. "Phishing detection: a literature survey". *IEEE communications surveys & tutorials,* Vol. 15, No. 4, fourth quarter 2013, pp. 2091–2121.

[2] D. P. Yada, P. Paliwal, D. Kumar and R. Tripathi. "A Novel Ensemble Based Identification of Phishing E-Mails", *Conference* ICMLC, Singapore, February 24–26, 2017, pp. 2-17.

[3] N. Abdelhamid, A. Ayesh and F. Thabtah. "Phishing detection based Associative Classification data mining. Expert Systems with Applications", *De Montfort University, Leicester UK,* Vol 41 No.13, 2014, pp. 5948–5959.

[4] V. Shreeram, M. Suban, P. Shanthi and K. Manjula, (2010). "anti-phishing detection of phishing attacks using genetic algorithm", *sastra university kumbakonam IEEE,* 2010, pp. 4244-7770.

[5] W. Zhuang, Q. Jiang and T. Xiong . "An Intelligent Anti-phishing Strategy Model for Phishing Website Detection", *Xiamen University, Xiamen,, P.R.China,* Vol. 10771176, 2012, pp. 51–56.

[6] R. A. Mohammad, F. Thabtan, and L. Mccluskey. "Predicting phishing websites based on self-structuring neural network", *Springer-Verlag London*, 2013.

[7] E. S. M. El-Alfy. "Detection of Phishing Websites Based on Probabilistic Neural Networks and K-Medoids Clustering", *Information and Computer Science Department, College of Computer Sciences and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia"* 2017.

[8] G. Liu, B. Qiu and L. Wenyin.(2010). "Automatic Detection of Phishing Target from Phishing Webpage" *Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave., HKSAR, China ,* 2010, pp. 4161–4164.

[9] G. Xiang, J. Hong, C. P. Rose and L. Cranor. "CANTINA +: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites", *Carnegie Mellon University,* Vol. 14, No. 2, 2011, pp. 1–28.

[10] P. A., Barraclough, M. A. Hossain, M. A. Tahir, G. Sexton and N. Aslam. "Intelligent phishing detection and protection scheme for online transactions. Expert Systems With Applications", *University of Northumbria at Newcastle, Newcastle Upon Tyne NE1, United Kingdom* Vol. 40 No. 11, 2013, pp. 4697–4706.

[11] Y. Li, R. Xiao, J. Feng and L. Zhao. "A semi-supervised learning approach for detection of phishing webpages". *School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, PR China*, Vol. 124 No. 23, 2013, pp. 6027–6033.

[12] N. Vaishnaw and S. Tandan. "Development of Anti-Phishing Model for Classification of Phishing E-mail", *Raman University, Bilaspur (C.G), India,* Vol. 4 No. 6, 2015, pp. 39–45.

[13] J. Solanki and R. G. Vaishnav. "Website Phishing Detection using Heuristic Based Approach", *Darshan Institute of Engineering and Technology, India,* Vol. 03, May-2016 pp. 2044–2048.

[14] Thabtah, F., and Abdelhamid, N. (2016). "Deriving Correlated Sets of Website Features for Phishing Detection: A Computational Intelligence Approach", *Information Technology Auckland Institute of Studies Auckland, New Zealand ,* Vol. 15, No. 4, 25 November 2016, pp. 1650042-1650056.

[15] H. B. Kazemian and S. Ahmed. "Comparisons of machine learning techniques for detecting malicious webpages. Expert Systems with Applications", Vol. 42 No. 3, 2015, pp. 1166-1177.

[16] A. K. Jain, and B. B. Gupta "Comparative analysis of features based machine learning approaches for phishing detection". *In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on IEEE*. March 2016, pp. 2125-2130.

[17] T. Zimmermann, T. Djürken, A. Mayer, M. Janke, M. Boissier, C. Schwarz, ,and M. flacker. "Detecting Fraudulent Advertisements on a Large E-Commerce Platform*"*. *In EDBT/ICDT Workshops*. 2017.

[18] X. Wei, F. Jiang, F. Wei, J. Zhang, W. Liao and S. Cheng. 2017, May). "An Ensemble Model for Diabetes Diagnosis in Large-scale and Imbalanced Dataset". *In Proceedings of the Computing Frontiers Conference* , 2016, March, pp. 71-78.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[19] L. Zhang and C. Zhan. "Machine Learning in Rock Facies Classification: An Application of XGBOOST*".,In International Geophysical Conference, Qingdao, China,* 17-20 April 2017, pp. 1371-1374.

[20] T. Chen and C. Guestrin. XGBOOST: A scalable tree boosting system. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 785-794.

[21] Jain A. (2016). Complete Guide to Parameter Tuning in XGBOOST (with code in python) Retrieved from https:complete guide to parameter tuning in XGBOOST (with code in Python). 2017/06/13.

[22] A. Gómez-Ríos, J. Luengo, and F. Herrera. "A Study on the Noise Label Influence in Boosting Algorithms: AdaBoost, GBM and XGBOOST". *In International Conference on Hybrid Artificial Intelligence Systems*, *Springer, Cham,* June 2017, pp. 268-280.

[23] Predrag Radenković. "Random forests" *Faculty of Electrical Engineering, University Of Belgrade*, 3237/10, 2010.

.