IOSR Journal of Research & Method in Education (IOSR-JRME)
e- ISSN: 2320–7388,p-ISSN: 2320–737X Volume 9, Issue 1 Ser. II. (Jan. – Feb. 2019), PP 53-59
www.iosrjournals.org

Investigation the Effect of Dataset Size on the Performance of Differentalgorithm in Phishing Website Detection.

Hajara Musa[1], Ali Ahmad Aminu[2], Amina Nuhu Muhammad[3], Jamilu usman waziri [4], Ahmad J. Kawu[5] ,
[1](Department of Mathematics, Gombe State University, Gombe Nigeria)
[2](Department of Mathematics, Gombe State University, Gombe Nigeria)
[3](Department of Mathematics, Gombe State University, Gombe Nigeria)
[4](Department of Mathematics, Federal University Gusau, Zamfara, Nigeria)
[5](Department of Mathematics, Gombe State University, Gombe Nigeria)
Corresponding Author: Hajara Musa

Abstract: *Phishers and other cybercriminals are making the cyberspace unsafe by posing serious risks to users and businesses as well as threating global security and economy. Nowadays, phishers are constantly evolving the techniques using luring user to revealing their sensitive information. Many techniques have been proposed in past for phishing detection, but due to static nature of some of the current and challenging nature of the problem, the quest for better solution is still on. In this paper, we developed phishing website model using XGBOOST algorithm to investigate the effect of dataset size using publicly available dataset composed of phishing and benign websites as in [1]. Experimental results demonstrated that as the number of instances of the dataset increases, the XGBOOST performance improve simultaneously, which shows that the XGBOOST has the highest performance than PNN algorithm.*

Keywords: *Machine learning, Feature selection, Xgboost, Phishing, RF, and PNN*.

Date of Submission: 23-01-2019 Date of acceptance: 07-02-2019

I. Introduction

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 the 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 includes; passwords and usernames to avoid losing access right to some of the services provided by that organization. Phishers use this avenue to obtained users sensitive information which in turn used it to access their important accounts resulting in identity theft and financial loss [3]

Many approaches have been proposed in an attempt to curb the problems caused by phishers [1-6]. El-Alfy in [1] proposed model for detecting phishing websites based on Probabilistic Neural Networks (PNNs),where investigated that the integration of PNN with K-medoids clustering significantly reduces the complexity without jeopardizing the detection accuracy. To assess the feasibility of the proposed approach, will be conducted in depth study to evaluate various performance measures on a publicly available data, the experimental results shown that 96.79% accuracy is achieved with low false errors. The problems of this approach reported require large memory spaces to store and the execution of network is slow. But XGBOOST has many advantages over traditional gradient boosting implementations. Among which include 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 [7].

II. Related Report.

The research paper was conducted by looking at the recent papers. Liu *et al*[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. Zhuang *et al* [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. Barraclough *et al* [9]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. Li *et al* [10] 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 unlabelled 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. Abdelhamid *et al*[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 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. Vaishnaw and Tandan [11] 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). Thabtah and Abdelhamid [12] 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. In [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 [1] 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. In recent time, machine learning techniques have been found to be very successful in phishing website detection [13-15]. 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 [16-18].

III. Methodology

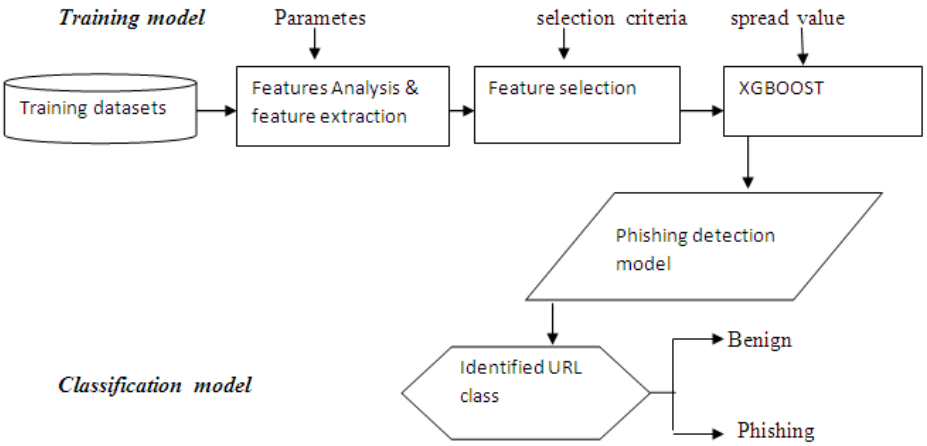


Figure1. illustrate the framework of the proposed model

A. Training model.

The model extract the dataset for removing the data or fixing the missing data, that is when data instances that are incomplete and do not carry the data we need to address the problem. These instances need to be removed. Removing empty cell from the dataset is important since is prevent the model to run with errors in order to get a best performance, handling missing data is also important as many machine learning algorithms. Clean missing data is the way of removing rows for which the selected column is empty by theprocessor. The goal of such cleaning is to prevent problems caused by missing data that can arise when training model [23]. Selected features used to train machine learning models have great influence on the performance of the models.Noisy features to the underlying relationship may adversely affect the perfomance of a model [24].

B. Classification model.

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 [19]. 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 $\boldsymbol{\mathcal{F}}$ 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

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

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:

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

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:

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

optimization objective using square loss can written as:

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

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

While Using Taylor expansion,

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

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

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

Where:

g_i and h_i comes from definition of loss function.

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 [20-21] . Ω (ft) is a regularization term which describe the complexity of the tree ft and is defined in the XGBOOST algorithm as

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

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:

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

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.

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

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:

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

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 (Chen, 2014).

IV. 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.

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

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

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

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

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

V. Result and Discussion

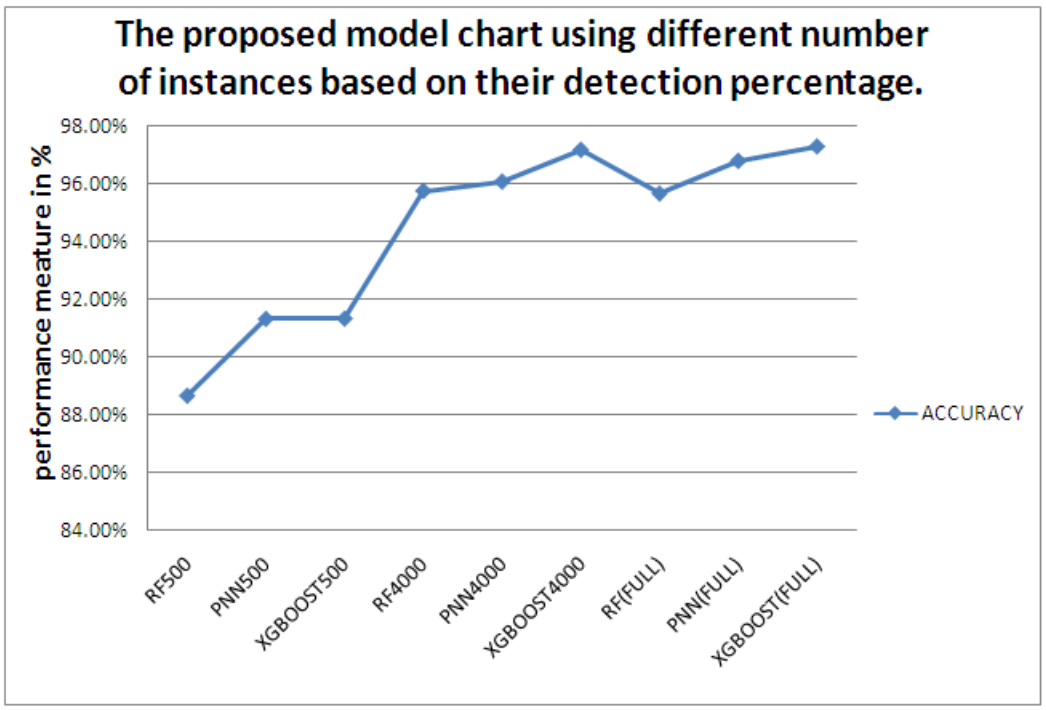
To evaluate the effect of the number of instance of the dataset on the performance of both the XGBOOST, PNN and RN, we built the model using different set of data. The first model utilized 500 instances of the datasets, the second model uses 4000 instances of the datasets, and finally the last model used the entire datasets. The first experiment investigates the effect of the number of instances of the dataset on the performance of the proposed model. Table 1 illustrates the experimental results obtained by the proposed model using the different number of instance of the dataset.

Table 1 RF, PNN and XGBOOST Results, using different setting.

Number of instances	Precision	Recall	F.Score	MCC	Accuracy
RF500	0.9195	0.8889	0.9039	0.7669	0.8866
PNN500	0.8981	0.9521	0.9243	0.8246	0.9132
XGBOOST(500)	0.9077	0.9239	0.9089	0.8179	0.9133
RF4000	0.951	0.9745	0.9628	0.9136	0.9575
PNN4000	0.9666	0.9626	0.9646	0.9203	0.9607
XGBOOST(4000)	0.9711	0.9503	0.9711	0.9423	0.9717
RF(FULL)	0.9433	0.9796	0.9611	0.9128	0.9566
PNN(FULL)	0.964	0.9789	0.9714	0.935	0.9679
XGBOOST(full)	0.9730	0.9801	0.9724	0.9449	0.9729

Table 1 shows that the result obtained from the experiment demonstrated that all the algorithms using 500 instances has the least number of instances which has the poorest result with smallest computational time when compared with 4000 number of instances, but 4000 number of instances has better result at the expense of computational time. Although the 4000 number instances attained nearly the same level of performance as XGBOOSTfull and PNNfull, but in case of RN4000 is outperformed the RNfull because it has the problems of pruning. XGBOOSTfull outperformed the other two compact form of the model in the five performance metrics employed; this clearly shows that as the number of instances of the dataset increase, there is a huge improvement on the performance of both the XGBOOST and PNN.

This result can be represented in a graphical form for analysis Figure 1



VI. Conclusion

Conclusively, this work has shown that XGBOOST can be adapted to obtain a very impressive result in detecting phishing. The performance of XGBOOST has been compared with that of well-known techniques Random forest and probabilistic neural network. The evaluation criteria are used in measuring the performance of phishing detection.Benchmark phishing website dataset were considered in the experiment. The result of the experiments showed that XGBOOST is better in most of the problems than the other methods in terms of MCC and Accuracy.Therefore, the xgboost method represents a very competitive technique for phishing detection.XGBOOST has a 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 which makes it good classification algorithm. In view of that, we recommend the application of XGBOOST to a more complex classification problem in future.

References

[1]. E. S. M. El-Alfy. Detection of Phishing Websites Based on Probabilistic Neural Networks and K-Medoids Clustering. *The Computer Journal*, 2017, 1-15.
[2]. D. P. Yadav, P. Paliwal, D. KumaD and R. Tripathi (2017). A Novel Ensemble Based Identification of Phishing E-Mails, 2017, 2–6.
[3]. N. Abdelhamid, A. Ayesh & F. Thabtah. Phishing detection based Associative Classification data mining. Expert Systems With Applications. *Elsevier 41.(13)*, 2014, 5948–5959.
[4]. V. Shreeram, M. Suban, P. Shanthi and K. Manjula. ANTI-PHISHING DETECTION OF PHISHING ATTACKS USING GENETIC ALGORITHM 978-1-4244-7770-8/10/$26.00 ©2010 IEEE.2010.
[5]. W. Zhuang. An Intelligent Anti-phishing Strategy Model for Phishing Website Detection, (10771176), 2012, 51–56
[6]. R. M. Mohammad, F. Thabtan and L. Mccluskey. Predicting phishing websites based on self-structuring neural network. 2013.
[7]. A. Jain (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.
[8]. G. B. Liu ,Qiu and L. Wenyin. Automatic Detection of Phishing Target from Phishing Webpage 2010, 4161–4164.
[9]. P. A. Barraclough, M. A. Hossain, M. A. Tahir, G. Sexton & N. Aslam. Intelligent phishing detection and protection scheme for online transactions. *Expert Systems With Applications*, 40. (11), 2013, 4697–4706.
[10]. Y. Li, R. Xiao, J. Feng and L. Zhao. A semi-supervised learning approach for detection of phishing webpages. Optik - International Journal for Light and Electron Optics, 124(23), 2013, 6027–6033.
[11]. N. Vaishnaw, S. R. Tandan and C. G. Bilaspur. Development of Anti-Phishing Model for Classification of Phishing E-mail, 4(6), 2015, 39–45.
[12]. F. Thabtah and N. Abdelhamid. Deriving Correlated Sets of Website Features for Phishing Detection: A Computational Intelligence Approach, 15(4), 2016, 1–17.
[13]. M. Kaytan and D. Hanba. Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines. 2017.

[14]. H. B. Kazemian and S. Ahmed. Comparisons of machine learning techniques for detecting malicious webpages. Expert Systems with Applications, 42(3), 2015, 1166-1177.
[15]. 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), IEE 2016 3rd International Conference on 2016, March., pp. 2125-2130.
[16]. T. Zimmermann, T. Djürken, A. Mayer, M. Janke, M. Boissier, C. Schwarz and M. Uflacker.Detecting Fraudulent Advertisements on a Large E-Commerce Platform. In EDBT/ICDT Workshops. 2017.
[17]. X. Wei, F. Jiang, F. Wei, J. Zhang, W. Liao and S. Cheng. An Ensemble Model for Diabetes Diagnosis in Large-scale and Imbalanced Dataset. In Proceedings of the Computing Frontiers Conference, may, 2017, (pp. 71-78). ACM.
[18]. 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). Society of Exploration Geophysicists and Chinese Petroleum Society.
[19]. G. James, D. Witten and T. Hastie. An Introduction to Statistical Learning: With Applications in R. 2014.
[20]. Tianqi Chen Oct. 22 2014. Introduction to Boosted Trees, university of Washington.
[21]. Tianqi Chen and Carlos Guestrin. 2016. XGBOOST: A scalable tree boosting system. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 785-794.
[22]. A. ÀGómez-Ríos, J. Luengo and F. Herrera. *A Study on the Noise Label Influence in Boosting Algorithms: AdaBoost, GBM and XGBOOST*. International Conference on Hybrid Artificial Intelligence Systems, Springer, Cham. 2017, 268-280
[23]. J. Brownlee (2013, Dec 25). Machine Learning Process. Retrieved February 14, 2018, from https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/.
[24]. G. Hackeling. Mastering Machine Learning With Scikit-Learn. Birmigham, United Kindom: Packt publishing Ltd. 2014.

Hajara Musa. "Investigation The Effect of Dataset Size on The Performance of Differentalgorithm in Phishing Website Detection.". IOSR Journal of Research & Method in Education (IOSR-JRME) , vol. 9, no. 1, 2019, pp. 53-59.