

# Boosting the Accuracy of Phishing Detection with Less Features Using XGBOOST

Hajara Musa<sup>1</sup>; Dr. A.Y Gital<sup>2</sup>; Mohzo Gideon Bitrus<sup>3</sup>; Dr. Nurul Farhana Juma'at<sup>4</sup>; Muhammad Abubakar Balde<sup>5</sup>

Department of Computer Science, Gombe State Univeisty Gombe, Nigeria<sup>1,3</sup>; Department of Mathematical Sciences, Abubakar Tafawa Balewa University Bauchi, Nigeria<sup>2</sup>; School of Education, Faculty of Science and Humanities, Universiti Teknologi, Malaysia<sup>4</sup>; Department of computer science, Federal Colloge of Education Yola, Nigeria<sup>5</sup>

*email:mhajara86@gmail.com*

## ABSTRACT

Phishing has been for a long time a difficult threat in every society as it changes form with time and it has taken billions of dollars from governments, companies and individuals alike. It is an identity theft which employs a kind of social engineering attack to get vital information from individuals or group of individuals. In this paper we focus on studying various features employed in different phishing attacks. So many studies have been conducted on single feature to have high accuracy for attack detection while others advanced on the use of many features to detect different attack behaviors with high accuracy. Researchers have advanced the study to the adoption and standardization of thirty (30) features to be examined in phishing attack in order to achieve high accuracy of detection. We examined all the features used so far and used XGBOOST classification model to categories the features into different kinds to detect important features. The analysis revealed that some features hampers on the accuracy and are unfruitful which also contributes in slowing the whole detection process. The model helps us to select useful features and weeds out the useless features. This yields higher accuracy and less time in detection process.

**Keywords** - Boosting, Detection, Features, Phishing, XGBOOST

## 1. INTRODUCTION

Phishing is a cybercrime in which cybercriminals attempt to obtain sensitive information of cyber users such as username, passwords credit card details often for malicious intent by disguising as a trustworthy entity in an electronic communication (websites, email, phone call, sms, etc.) [12]. The information gained by phishers are often used to access users important accounts (facebook, twitter, email and bank) which may result in identity theft and financial losses [15]. Phishing as a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credential. Phishers attempt to acquire these credentials; user names, password and credit card details by masquerading as trustworthy entities while exchanging data with Internet users during users' interaction with a system. Some time, they appear as a third party in between users and a legitimate system without any awareness of their presence.

Phishers followed these steps to achieve their goal through phishing life cycle. Example, A fake webpage generally contains a login form, and when a user opens the fake webpage and inputs personal information, this information is accessed by the attacker. Furthermore, the attackers use this

*Hajara Musa; Dr. A.Y Gital; Mohzo Gideon Bitrus; Dr. Nurul Farhana Juma'at; Muhammad Abubakar Balde Vol 8 Issue 2, pp 81-90, February 2020*

information for some personal and financial gain [13].

The Life cycle of a phishing attack is shown in Fig. 1



Figure 1 Phishing e-mails Life Cycle [13].

Figure 1 demonstrate phishing life cycle which can be described using the following steps:

Step 1: The attacker copies the content from the website of a well-known company or a bank and creates a phishing website. The attacker keeps a visual similarity of the phishing website similar to the corresponding legitimate website to attract more users.

Step 2: The attacker writes an email and includes the link of the phishing website and sends it to a large number of users. In the case of spear phishing, a mail is sent to only select targeted users.

Step 3: The user opens the email and visits the phishing website. The phishing website asks the user to input personal information, for example, if the attacker mimics the phishing website of a well-known bank, then the users of bank are very likely to give up their credentials to the fake website.

Step 4: The attacker gets personal information of the user via the fake website and uses this information of the user for financial or some other benefits.

There are different types of phishing in which the few types of phishing attacks are mentioned in the discussion; pharming, Content-Injection Phishing, Deceptive phishing, Malware-based phishing, Hosts file poisoning, man-in-the-middle and web Trojans phishing.

## 2. RELATED WORKS

The research papers related to Phishing detection selected from 2007 to 2019 are summarized in order to select a collection of features commonly used that could yield high Phishing detection accuracy.

In 2010, they used some Anti-Phishing and Network analysis tool (CANTINA) heuristic features with a new additional attributes that are combined with the new ones (Domain Top page Similarity) to detect Phishing pages. The experiment was done in three metrics: first, with the intend to test CANTINA's reduced features, secondly, to test the new feature and thirdly to test the performance of some machine learning features using the combination of CANTINA's reduce features and the new feature. Their experiment used some machine learning techniques such as Naive Bayes (NB), Neural Network (NN), Support Vector Machine (SVM), Random Forest (RF), J48 Decision tree and Adaboost with the features set (F37, F24, F2, F19, F20, F38, and F40) [16].

In 2011, they defined a set of rules in order to identify Phishing web pages. Two groups of rules were distinguished: -the simple rules, based on web page URL and the higher complex and time consuming rules based on the analysis of meta-data, query search engines and blacklists the search engine based rules, the red aged keywords based rules, the obfuscation based rules, the blacklists based rules, the reputation based rules, the content based rules. In order to build the rules, they identify and examine a tactic employed by phishers over a well known dataset of Phishing websites. Rules are initially assigned the same weight and while using a carefully chosen threshold, if the number of rules found in a web page is superior to the threshold, then the page is considered as Phishing. Decision tree and logistic regression

were used and performed good results. The following are the features used: ( F29, F70, F71, F72, F76, F7, F28, F38, F95, F25, F96, F31, F42, F8, F97, F2, F24, F91, F19, F20, F34, and F35) [6].

In 2011, they paid attention on obfuscation techniques operate on URLs domain name in order to detect Phishing web pages. Below are features used for this purpose; (F4, F5, and F76) [11].

In 2011, they contributed in WebPages Phishing detection by checking some characteristics of the web page source code that are not in respect with the W3C standards. A webpage is considered secured when the computed security percentage is 80% or higher, doubtful when it is between (50%-80%) and Phishing when it is less than 50%. Below are features selected to achieve their framework; (F41, F9, F2, F24, F10, F1, F42, F43, and F44) [2].

In 2012, they extracted identities from some features such as Meta title, meta description, content attributes and "href" attributes of tag <a> for detecting Phishing attacks. These features are tokenized with the objective to retain the first five keywords with high weight as identity set. Secondly, they extracted seventeen features based on the identities extracted in the previous step. Finally, the techniques used for evaluation were Multi Layer Perceptron (MLP), Decision tree induction and Naive Bayes. Their set of features was as follow: (F17, F18, F2, F19-F20, F2, F22, F23, F24, F25, F4, F26, F27, F28, F29, F30, and F31) [8].

In 2012, they used a feature vector of size 23, in which four were structural features from URLs, nine were lexical features and ten were features targeting mostly brand and websites. SVM was used for experiment with feature set: (F2, F24, F25, F19-F20, F46, F47, F11, F48, F53, F54, and F63) [14].

In 2012, they evaluated two features selection techniques: The correlation based and wrapper based feature selection techniques. The correlation-based technique has the ability to generate a subset of features with the goal to improve the classification accuracy and reduce the feature dimension while exploiting the

predictability of one variable with another. The wrapper-base technique uses a machine learning algorithm while taking into consideration the fact that the method that has to use the feature subset should yield a good accuracy. Their experimental result demonstrates that a feature selection technique can improve classification results when trained and tested on a disjoint subset of dataset. They also show in their experiment that the Wrapper-base classification technique significantly improved the accuracy compared to the correlation-base technique even though it was extremely slow. The feature selection techniques were evaluated using the following machine learning techniques: Naive Bayes (NB), Logistic Regression (LR) and Random Forest (RF). Features used were as follow: (F8, F24, F42, F29, F31, F7, F38, F19, F20, F46, F94, F2, F24, and F11-F16)[7].

In 2013, they proposed a framework to detect hidden URLs based on lexical features. A URL is hidden if its corresponding page is hosted within a legitimate site without the site's administrator being aware. Hence, the HTTPs authentication becomes inefficient in detecting Phishing URLs. This approach uses only lexical features as shown below and yielded a high accuracy: (F36, F107) [5].

In 2013, they used lexical and domain features extracted from Phishing URLs to detect Phishing websites. They equally evaluated the effectiveness of machine learning based Phishing detection when they targeted websites are known. An optimal set of features was selected for this purpose: (F64, F65, F66, F67, F68, F69, F70, F75, F76, F3, F77, F78, F79, and F80) [9].

In 2014, they put forth a technique called Tabsol to fight against Tab nabbing. Tab nabbing is a recent variant of Phishing attack in which a malicious page opened in a tab disguises itself to a popular website's login page such as Gmail, Facebook login pages with the objective to steal credentials. The framework identifies Tab nabbing attacks base on hash value comparison of the web page at different instances. This means that if any inconsistency of hash values is found between two states of a web page, then there is

Phishing activities going on. An example of web page states are: When a page is focus (occupies the screen) and when the page regains its focus after the focus has been lost. The features used in Tabsol are F104, F105, F7, and F38 [14].

In 2014, they proposed a technique for detecting Phishing web pages based on the discrepancy between the claimed identity and the domain name owner of the website. From a given web page, this technique extracts domain name, then tries to build a strategy to determine the domain name based on brand names from the web page content and compare to see if the extracted domain name matches the domain name generated based on brand names. In case of any mismatch found, they concluded a Phishing activities taking place. Features used for this purpose are F39, F29, F104, and F106 [3].

In 2015, they put forth a framework for Phishing web page detection based on URL analysis. The analysis consists to extract lexical features combine with bag of words approach for Phishing detection. They claimed that their framework achieved good accuracy while maintaining a low time. N-grams features, counting features, length features, pattern features and ratio features are some types of features they extracted from URLs. The features were F108, F34, F92, F35, F68, F94, F64, F65, F66, F100, F80, F109, F110, F111, F112, and F113 [10].

In 2016 [18] 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. [19] 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

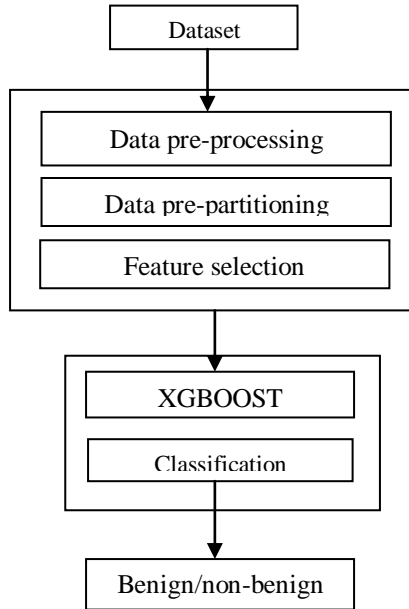
formal representation and it deals badly with quantitative measurements. The emails have been classified as phish using the prediction of Ensemble Classifier of the five ML Algorithms. Experiment shows that the comparison of the accuracy of algorithms for Different Feature Groups based on the decisive 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 [20]. 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.

In 2019 they proposed XGBOOST algorithm which improved the performance that a predictive model can achieve in the task of phishing website detection, their research 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, XGBOOST algorithm returned the accuracy of 97.27% [1].

More Phishing e-mails nowadays come in an attractive ways mostly as advertisements e-mails or pornography e-mails, an e-mail could redirect to a website via its links and none of these above researchers took into consideration advertisement e-mails and pornography e-mails and they equally do not pay attention on defining a set of rules that could lead to good classification. Based on the various operational modes of phishing attacks, we can conclude that an efficient detection of phishing email attacks could be successful by identifying a good set of features that may serve as rules for our framework. Hence, we proposed an Alerting system for detecting and alerting Phishing-emails based on a well selected set of features. Rules are defined based on good

collection of features which have shown high detection rate.

**3. Workflow Framework of the research.**



In order to test the dataset , feature selection is important because dataset may contain irrelevant noisy and redundancy, feature in which if they are included (incorporated), it will surely affect the model negatively. Feature selection is one of the data mining techniques used in data pre-processing stage. Firstly, the relevant datasets are collected and pre-processed before being fed into the proposed model for training and testing. Finally, the model is evaluated based on standard evaluation metrics and the model classified either the website is benign or phishing.

**4. Evaluation Criteria**

To evaluate and compare the performance of the 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. MCC measures the correlation coefficient between the predicted and actual class. F-score measures the weighted harmoni mean of precision and recall. All metrics employed are functions of the confusion matrix as can be seen in the mathematical formulations. The confusion matrix shown in table 3.2 which is used to describe the performance of a classification model on a set of test data for which the true values are given.

Table 1 Confusion Matrix

	Predicted positive class	Predicted negative class
Actual positive class	TP	FN
Actual negative class	FP	TN

Table 1 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.

$$ACC = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (1)$$

$$Prec = \frac{TP}{(TP + FP)} \quad (2)$$

$$Rec = \frac{TP}{(TP + FN)} \quad (3) F - score =$$

$$\frac{2 * (Prec * Rec)}{(Rec + Prec)} \quad (4)$$

$$MCC = \frac{(TP * TN) - (FP * FN)}{Sqrt (TP + FP)(TP + FN)(TN + FP)(TN + FN)} \quad (5)$$

**5. Classification Model**

Phishing detection is a supervised learning problem where we use the training data xi to predict a target variable yi. The inputs to the

phishing detection model are usually pairs of training instances  $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$  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). We want to determine some useful parameters of the model using the available dataset so that at any given instance, the model can use those parameters 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 substantially [21]. XGBOOST is tree based model that aggregates trees using the boosting technique. In XGBOOST we used the training data  $x_i$  to predict the target variable  $y_i$  iteratively until the parameters of the model are optimized. Mathematically, the proposed phishing detection model can be represented as follows:

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

$$\hat{y}_i = \sum_k^K f_k(x), f_k \in F \tag{6}$$

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 on each iteration in 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

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{7}$$

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 ( $l(\Theta)$ ) encourages predictive models as well as optimizing regularization ( $\Omega(\Theta)$ ) encourages simpler model to have smaller variance in future predictions, making prediction stable [22]. The closed form of the objective is given below:

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \tag{8}$$

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. Similarly, our optimization objective at step  $t$  of the training process can be formulated as:

$$obj^t = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \Omega(f_k) \tag{9}$$

Optimization objective using square loss can be written as:

$$l = (y_i - \hat{y}_i^{(t)})^2, \text{ but } \hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

$$obj^{(t)} = \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) \tag{10}$$

While Using Taylor expansion, Objective, with constants removed, the new form of optimizing goal is:

$$obj^{(t)} = \sum_{i=1}^n \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{11}$$

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 (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 [22].  $\Omega(f_t)$  is a regularization term which describe the complexity of the tree  $f_t$  and is defined in the XGBOOST algorithm as:

$$\Omega(f_t) = \gamma T + \frac{1}{2} \sum_{j=1}^T \omega_j^2 \tag{12}$$

Where  $T$  is the number of leaves of tree  $f_t$  and is the leaf weights (i.e the predicted values at the leaf nodes).

And  $\gamma$  and  $\lambda$  are constants,  $\gamma$  and  $\lambda$  are the Lagrangian multipliers and can be tuned for accuracy, that is user defined parameters.

Given the description of XGBOOST, it is time to differentiate the model with Random Forest; 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 compliment 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 over fitting [22].

**6. Dataset Description.**

In order to assess and compare the predictive performance of the proposed model, we adopted a recently created phishing websites dataset from UCI machine learning repository. This dataset was created by Mohammed, 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 pre-classified as benign (non phishing) and phishing websites with 30 features. Each website is converted into a vector  $x = (x_1, x_2, \dots, x_n)$  where  $x_i$  are the values corresponding to specific feature (variable) of a particular website. Features in dataset are divided into four categories. The first category (f0 -f11) is the address bar based features, the second category (f12 -f17) are abnormal based features. The third category (f18 -f22) is html and JavaScript based features and the last category

(f23-f29) is domain based features. In the value range column, a value of -1 means benign, 0 means suspicious and 1 means phishing.

Table below shows the descriptions of features contained in phishing websites dataset. The table contains 30 features associated with their descriptions.

S/N	Features	Feature Notation	Value range
F0	Having IP Address	Has_ip	{-1, 1}
F1	URL Length	url_length	{-1,0, 1}
F2	Using URL Shortening Service	Short_service	{-1, 1}
F3	URL having the @ symbol	Has_@_symbol	{-1,1}
F4	URL has redirect symbol	Double_slash_redirect	{-1,1}
F5	Prefix or suffix to domain	Pref_suf	{-1,1}
F6	Having subdomains	Has_subdomain	{-1,1}
F7	Using HTTPS with trusted certificate	Ssl_state	{-1,1}
F8	Domain registration length	Long_domain	{-1,1}
F9	Favicon	Favicon	{-1, 1}
F10	Use of non standard port	Nonst_port	{-1, 1}
F11	HTTPS is Domain part	https_token	{-1, 1}
F12	External object URL	External_request	{-1, 1}
F13	Anchor URL refer to another domain	Anchor_url	{-1, 0, 1}
F14	pppLinks in meta, script, link tags	Links_tag	{-1, 0, 1}
F15	Server from handler	SFH	{-1, 1}
F16	Submitting to email	Submit_email	{-1, 1}
F17	Abnormal URL	Abnormal_url	{-1, 1}
F18	Website forwarding	Redirect	{0, 1}
F19	Status bar customization	Mouseover	{-1, 1}
F20	Disabling right click	Right_click	{-1, 1}
F21	Use of pop up window	Popup	{-1, 1}
F22	Iframe redirect	Iframe	{-1, 1}
F23	Domain age	Domain_age	{-1, 1}
F24	DNS record	Dns_record	{-1, 1}
F25	Website traffic	Website_traffic	{-1, 0, 1}
F26	Page rank value	Page_rank	{-1, 1}
F27	Google indexed	Google index	{-1, 1}
F28	Links pointing to websites	Links_to_page	{-1, 1}
F29	Statistical Report	Stats_report	{-1, 1}
	Result	Result	{-1, 1}

Table 2 Description of phishing websites dataset (Sources: UCI machine learning repository, 2015).

Figure 3.2 below is a plot of each feature of our datasets and their importance score value. The larger the score value, the more important the

feature has in the predictive performance of a model. Based on the importance value, we selected a subset  $X_s$  of  $X$  (entire feature set) that can predict  $y$  (response variable) with the best performance. Feature importance in tree ensemble models is given by how frequently a feature has appeared in the three model, we performed feature selection using XGBOOST built in function to plot features ordered by their importance on a random subset of 3,000 instances to prevent sampling bias. Feature selection has huge influence in the performance of model; it helps in reducing training time, preventing overfitting, reducing computational cost and above all improving the performance of a model [23]. Figure 3 shows the description of features contained in phishing websites dataset.

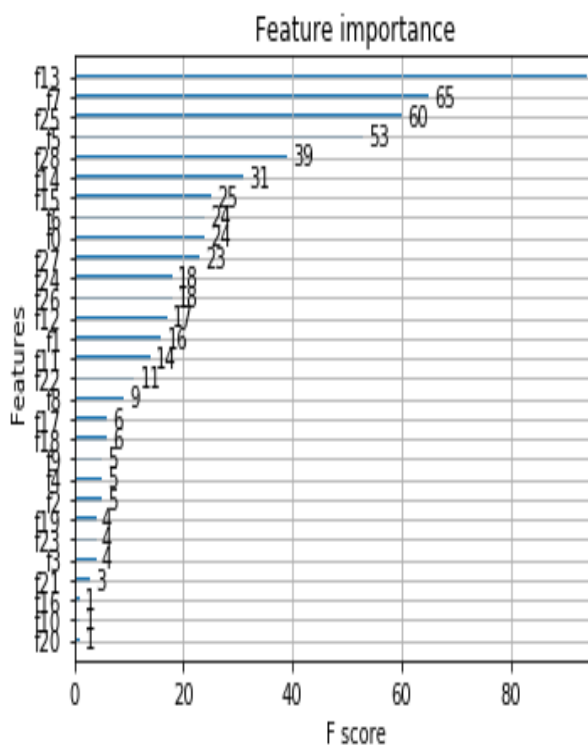


Figure 3 Features against its importance in a phishing classification.

Based on the above figure we evaluated and analyzed on the important features among phishing dataset created by Mohammed, Thabtah and McChushy at the University of Huddesfied, United Kingdom and denoted to UCI machine learning

repository in 2015. The plot show that some features are obviously not important in as a result hampers on the accuracy of the detection. We termed the least important features, the useless features and removing them and using only the important ones yields an increase in accuracy and faster detection process.

**7. Comparative Analysis of Features**

The plot below shows the comparative accuracy gotten when a plot of full features are used to show the performance of the algorithm and also when only few selected features are used. The chard reveals that higher accuracy is gotten from using only the few selected features with accuracy of 97.41% while a full features gives the accuracy of 97.29%. in our former work[1], we showed that using full features with this algorithm yield a better and higher than known method. But this current work yields even far better accuracy with fewer features using same algorithms. So therefore the comparative analysis chart between using full features and using few selected features are as shown bellow.

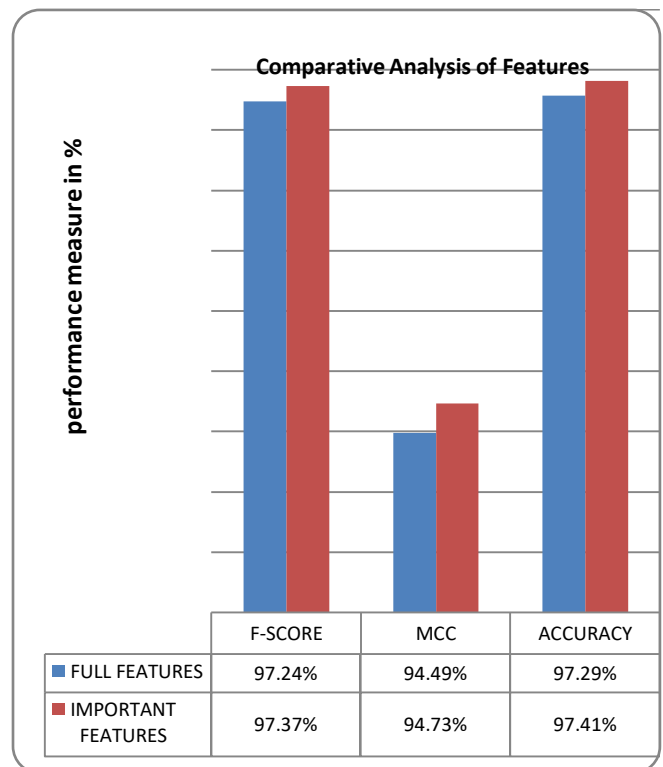


Figure 4 Comparative Analyses of Features

The table below shows the F-score, MCC and the Accuracy for using full features and using only the selected important features. The full features has an accuracy of 97.29 while the important features has the accuracy of 97.41

Features	F-Score	MCC	Accuracy
Full features	97.24	94.49	97.29
Important features	97.37	94.73	97.41

Table Full Features and Important Features Result

## 8. CONCLUSION

In this research, we have clearly shown and demonstrated the boost in accuracy of phishing detection by a careful selection of only some important features from 30 used features using XGBOOST algorithm. This research contributes to knowledge by boosting the accuracy that outperforms other widely known algorithms like: PNN and RFetc trained on the same dataset with the same evaluation criteria for fairness; the most widely used algorithms use 30 features which is standardized for detecting phishing for detection but in this work we have demonstrated that some features are absolutely useless and hampers on the accuracy of the detection and also slows down the detection process. The method helps us weed out those useless features and only uses the important features to boost the accuracy of detection.

## 9. REFERENCES

[1] Hajara Musa, A.YGital, F.U Zambuk, Abubakar Umar, Aishatu yahya umar, and

jamilu waziri, A comparative Analysis of phishing websites detection using XGBOOST Algorithm, 2019 Journal of Theoretical and Applied Information Technology, Vol. 97, No 5

[2] Mona Ghotai Alkhozai and Omar Abdullah Batar\_, Phishing websites detection based on phishing characteristics in the webpage source code, International Journal of Information and Communication Technology Research 1 (2011), no. 6.

[3] Choon Lin Tan, Kang Leng Chiew, et al., Phishing website detection using url-assisted brand name weighting system, Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on, IEEE, 2014, pp. 054-059.

[4] Amandeep Singh and Somanath Tripathy, Tabsol: An efficient framework to defend tab nabbing, Information Technology (ICIT), 2014 International Conference on, IEEE, 2014, pp. 173-178.

[5] Enrico Sorio, Alberto Bartoli, and Eric Medvet, Detection of hidden fraudulent urls within trusted sites using lexical features, Availability, Reliability and Security (ARES), 2013 Eighth International Conference on, IEEE, 2013, pp. 242-247.

[6] Ram B Basnet, Andrew H Sung, and Quingzhong Liu, Rule-based phishing attack detection, International Conference on Security and Management (SAM 2011), Las Vegas, NV, 2011.

[7] Feature selection for improved phishing detection, Advanced Research in Applied Artificial Intelligence, Springer, 2012, pp. 252-261.

[8] V Santhana Lakshmi and MS Vijaya, Efficient prediction of phishing websites using supervised learning algorithms, Procedia Engineering 30 (2012), 798{805.

[9] Weibo Chu, Bin B Zhu, Feng Xue, Xiaohong Guan, and Zhongmin Cai, Protect sensitive sites

from phishing attacks using features extractable from inaccessible phishing urls, Communications (ICC), 2013 IEEE International Conference on, IEEE, 2013, pp. 1990-1994.

[10] Michael Darling, Greg Heileman, Gilad Gressel, Aravind Ashok, and Prabaharan Poornachandran, A lexical approach for classifying malicious urls, High Performance Computing and Simulation (HPCS), 2015 International Conference on, IEEE, 2015, pp. 195-202.

[11] Mahmoud Khonji, Andrew Jones, and Youssef Iraqi, A novel phishing classification based on url features, 2011 IEEE GCC Conference and Exhibition (GCC), 2011.

[12] Toolan, F., and Carthy, J. (2009). Phishing detection using classifier ensembles. In eCrime Researchers Summit, 2009. eCRIME'09. IEEE. pp. 1-9.

[13] Almomani, A., Gupta, B.B., Wan, T. and Altaher, A. (2013), Phishing Dynamic Evolving Neural Fuzzy Framework for Online Detection Zero-Day Phishing Email. Indian J. Sci. Technol. 6. (1) 3960–3964.

[14] Huajun Huang, Liang Qian, and Yaojun Wang, A svm-base technique to detect phishing urls, Information Technology Journal 11 (2012), no. 7, 921.

[15] Gupta, B.B., Tewari, A., Jain, A.K. and Agrawal, D.P. (2016) Fighting against phishing attacks: state of the art and future challenges. Neural Comput. Appl., First Online, 1–26.

[16] Nuttapong Sanglerdsinlapachai and Arnon Rungsawang, Using domain top-page similarity feature in machine learning-based web phishing detection, Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on, IEEE, 2010, pp. 187-190.

[17] Anh Le, Athina Markopoulou, and Michalis Faloutsos, Phishdef: Url names say it all, INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 191-195.

[18] 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.

[19] Thabtah, F., and Abdelhamid, N. "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.

[20] 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.

[21] 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.

[22] 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.

[23] Brownlee, J. (2016, March). Machine learning algorithms. Retrieved from <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/> 2019/09/14.