IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

International Journal of Safety and Security Engineering
Vol. 12, No. 3, June, 2022, pp. 381-386
Journal homepage: http://iieta.org/journals/ijsse

An Intelligent Prediction of Phishing URLs Using ML Algorithms

Lohith Ranganatha Reddy Kandula[1], T. Jaya Lakshmi[1*], Kalavathi Alla[2], Rohit Chivukula[3]

[1] Department of Computer Science and Engineering, SRM University, Guntur 522502, Andhra Pradesh, India
[2] Vasireddy Venkatadri Institute of Technology, Guntur 522508, Andhra Pradesh, India
[3] School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, United Kingdom

Corresponding Author Email: jayalakshmi.t@srmap.edu.in

https://doi.org/10.18280/ijsse.120312

Received: 23 December 2021
Accepted: 6 April 2022

Keywords:
phishing, data analytics, cyber security, analysis machine learning, malware analysis

ABSTRACT

History shows that, several cloned and fraudulent websites are developed in the World Wide Web to imitate legitimate websites, with the main motive of stealing sensitive important informational and economic resources from web surfers and financial organizations. This is a type of phishing attack, and it has cost the online networking community and all other stakeholders thousands of million Dollars. Hence, efficient counter measures are required to detect phishing URLs accurately. Machine learning algorithms are very popular for all types of data analysis and these algorithms are depicting good results in battling with phishing when we compare with other classic anti-phishing approaches, like cyber security awareness workshops, visualization approaches giving some legal countermeasures to these cyber-attacks. In this research work authors investigated different Machine Learning techniques applicability to identify phishing attacks and distinguishes their pros and cons. Specifically, various types of Machine Learning techniques are applied to reveal diverse approaches which can be used to handle anti-phishing approaches. In this work authors have experimentally compared large number of ML techniques on different phishing datasets by using various metrics. The main focus in this comparison is to showcase advantages and disadvantages of ML predictive models and their actual performance in identifying phishing attacks.

1. INTRODUCTION

Phishing is a method of fraudulent attack by the attacker, attacker tries to grab sensitive data by impersonating the user as an authorized resource. In such type of phishing attacks, whenever the victim opens a compromised URL that mimics like a reliable website [1]. Then the victim is asked to supply his login credentials, since the URL is a fake URL, all the sensitive and highly confidential information is routed to the attackers and the victim is hacked by the attacker. Sometimes, victim receives a message as if the message has come from very known contacts or organization [2]. The message encompasses a malicious software which directly targets the user computer or contains some links to direct eh victim to malicious websites, to trick the users to divulge their personal and financial information such as passcodes, account IDs, credit card and debit card details [3]. URL is constructed to address web pages. URL starts with a protocol to access the web site [4]. The fully qualified domain name finds the server which is hosting the web page. It contains a registered domain name and a suffix. The domain name must be constrained since it should be registered with domain name registrar. Host name contains a sub domain name and domain name [5]. Phisher has full control over the parts of a sub domain and can set his own values. The URL also contains path of the web page and its components. These two components can also be modified by the phisher. So, the sub domain name and path can be fully controlled by the phisher. Therefore, we call this part of the URL as PhisherURL in the rest of the research paper. These components of URL are shown in Figure 1.

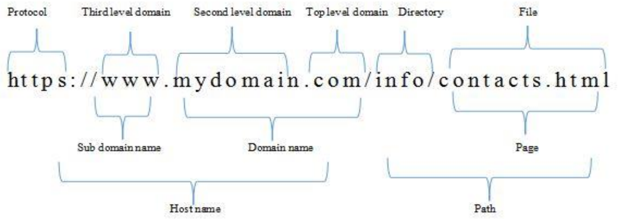


Figure 1. Various components of URL

Table 1. Sub parts of URL: http://amazon.com- secapps userid735892&limit.amazonapps.com/secapps/login.html

Description	Sub-part in the example URL
Protocol	http://
Domain Name	amazonapps.com
Path	userapps/login.html
Sub Domain1	com-secapps userid735892&limit
Sub Domain2	amazon

The attacker or Phisher can register any domain name which is not registered before. This can be set only once. But the attacker can change PhisherURL part at any point of time to generate a new web link. Due to this reason, many cyber warriors generally struggle to detect fake URLs. Once the domain is detected as fraudulent, then it is very easy to prevent this domain to be accessed by the users. Few Cyber

Intelligence companies detect and publish fake web sites and their IP addresses as black lists, in order to prevent these sites to be accessed by other users. The attacker should intelligently choose domain names to convince the users that it is a legitimate site and then he can set the PhisherURL in such a way that it should not be easily detected by the users. The subparts of a sample URL is given in Table 1.

Even though the URL is amazonapps.com, the phisher constructed the domain like amazon.com by adding PhisherURL to it. when users observe amazon.com in the beginning of the URL, by seeing these users simply trust the web site and supply their sensitive information in the form of user id and password to the fraudulent users. This one is the most frequently used attack by the phishers. The other type or procedures that are also used by most of the attackers is typo squatting and cybersquatting. Cybersquatting is a process of trafficking in, registering [6]. The cyber squatter may sell the domain name to a person or an organization who owns the trademark. For example, if our company name is freeshopping.com, the cyber squatter creates domains like freeshopping.biz, freeshopping.net, freeshopping.org to use for the fraudulent purposes. This process is also known as domain squatting. On the other hand, in typo squatting which is also represented as URL hijacking which mainly focuses on typographical mistakes done by web surfers. Such kind of URLs looks like trusted domains. Few examples of typo squatting are yawhoo.com, amzoon.com, goggle.com, microwsoft.com, appie.com, yutube.com [7]. Phishing web pages can be identified using domain-based features, content, page ranking and URL based features by applying machine learning algorithms. This paper portrays different statistics on phishing. As per the phishing statistics of 2020 data collected from a Verizon based organization DBIR (A Data Breach Investigation Report Organization) states that almost 30% of the sensitive data was breached due to phishing [8]. Other statistics based on SonicWall's cyber-attack report, most of the cyber attackers are overwhelmed with these phishing attacks. Knowbe4 2020 report shows that 39% of the uneducated web users are failed in identifying the phishing attacks. Confess report portrays that 90% of the threats were found in most of the secured Email gateways. Practically 75% of the phishing web pages normally use HTTPS protocol [9]. Freshly in a short span of seven days, Google search engine has blocked approximately 18 million emails per day and 250 million spam messages related to covid 19 by suspecting these as smishing (SMS phishing) and Email phishing [5]. Hence it is very necessary to conduct more analysis and research in this cyber security. It is also highly beneficial to explore and educate the people on cyber-squatting attacks.

2. RELATED WORKS

Numerous numbers of research papers on phishing detection are reviewed to analyze the performance of various phishing detection algorithms [4, 10-14]. Machine Learning techniques are proven to be more powerful in many data analysis applications like medical diagnosis, market price analysis, weather forecasting and even in cyber analytics also [15]. This is since Machine Learning techniques can analyze the performance even from the large datasets also [16]. Lately, there are several studies conducted to acquire automated rubrics to distinguish trusted and imitated web sites using statistical analysis [17]. For instance, authors in the research paper [6] characterized various intelligently derived rules to classify the different web site features by using frequency analysis on websites based on the statistics posted by yahoo and Phishtank. More progressions came in framing rules to take decisions using computational intelligence methods on a large phishing data set collected from various data repositories [3]. Phishing was explored using decision tress, support vector machines, Random Forest Trees and Naïve Bayes [18]. Phishing detection can also be done by using learning on features of email received by the user. This feature was designed under anti-phishing approach [19]. This method is conducted on a set of 920 phishing and 720 harm cases. Results shown that there are IP URLs, HTML tags, Java Script features and number of links involved in the message component of the Email. Authors also focused that PILFER improved the clustering of different messages by joining several features identified in the "Spam filter output". To improve false positives and false negatives, authors have used Random Forest classification algorithm on approximately 2000 sample messages. In another attempt authors categorized features into different criteria and then uploaded these features into WEKA system [3]. Various experimented conducted on it using different classification algorithms against 2000 instances which are downloaded from Phishtank [7]. The evaluation parameters detect the pertinency of these features is classification accuracy. Results showcased that 83% phishing domains were detected using decision tree algorithm. Authors analyzed on the significant features that have are used comprehensively and successfully in preventing phishing sites [19]. Additionally, new measures are proposed, to update different features of phishing sites.

3. MACHINE LEARNING ALGORITHMS

There are several lexical features to detect the phishing URLs. These features are grouped as URL based, page based, Domain based and Content features [20]. URLs have some distinct features like Digit Count, URL length, number of sub domains, is the URL typo squatted or not. The second classification is based on Domain based, and its features verify whether the domain or its IP address is blacklisted or not? Age of the Domain, is the registrant's name hidden or not? The third classification contains page-based features like page rank, web traffic, average page views, Average Page view duration [21]. Lastly to obtain content-based features one has to scan the target domain. With this scanning all page contents are processed, so that one can easily find whether the target site is exposed to phishing attacks. Some of the processed data about page features is page titles, meta tags, concealed texts, different type of text in the body of the mail and images. In this research paper authors have tested with 30 different phishing URL detection features on a large dataset which is downloaded from Kaggle. The performance is analyzed using Decision Tree, Random Forest, Support Vector Machine and XGBoost algorithms.

3.1 Decision tree classifier

A supervised Machine Learning algorithm that can be used for classification and regression. But preferably this algorithm is used for classification related problems only [1]. Decision Tree is a tree structured classification algorithm which represents the features of input dataset using internal node and branches designates decision rules. Each leaf node of the tree denotes the outcome.

Typically Decision Tree starts with a single node which branches into possible outcomes. These possible outcomes lead to additional nodes, which branch off into other possibilities. It contains three types of nodes: chance nodes, decision nodes and end nodes. Chance node is represented by a circle which shows the possibilities of certain results. A decision node is represented by a square, which shows a decision to be made, and finally an end node shows the final outcome of a decision path. The structure of Decision Tree is given in Figure 2.

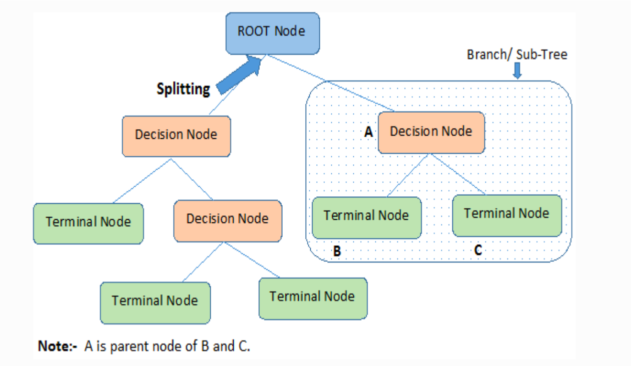


Figure 2. Structure of decision tree

During the implementation algorithm selects various attributes based on the two most popular attribute selection measures. If the data set contains N attributes, then it will be very difficult to select which attribute should be selected as an internal node at different levels of the tree. Random selection will not solve the issue. If random approach is followed, we may get bad results with low accuracy. So to solve this attribute selection problem, authors used information gain, entropy and GINI index as criteria to decide internal node. These criteria will calculate values for every attribute. The values are then sorted and placed in the tree by following the order, i.e., the attribute with the highest value (in case of information gain) is placed at the root. When we use Information Gain as a measure, we assume attributes to be categorical and for the Gini Index, attributes are assumed to be continuous. Entropy is a measure of randomness in the information is being processed. The higher the entropy, the harder is to draw conclusions from the given information. Flipping a coin is an example of an action that provides information that is random. Information Gain, Entropy and GININ Index are calculated using the following formulae.

Information Gain = Entropy before splitting - Entropy after splitting.

$$Entropy(P) = -\sum_{i=1}^{n} p_i \, log_2(p_i)$$

$$Gini(P) = \sum_{i=1}^{n} p_i \,(1 - p_i) = 1 - \sum_{i-1}^{n} {p_i}^2$$

3.2 Random forest classifier

Another supervised Machine Learning Algorithm which is also used for both classification and Regression analysis [16]. It is composed of different decision trees, each with sample nodes, but uses different data that leads to different leaves. It merges the decisions of multiple decision tress in order to solve the problem, which represents the average of all these decision Trees. When using Random Forest Classifier algorithm, to solve regression problems MSE (Mean Squared Error) should be used to how the data branches from each node.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (f_i - y_i)^2$$

where, N is the number of data points, f_i is the value returned by the model and y_i is the actual value for data point i. The general structure of random forest classifier is shown in Figure 3.

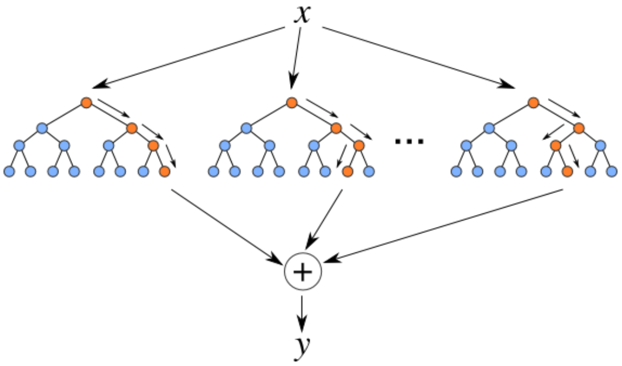


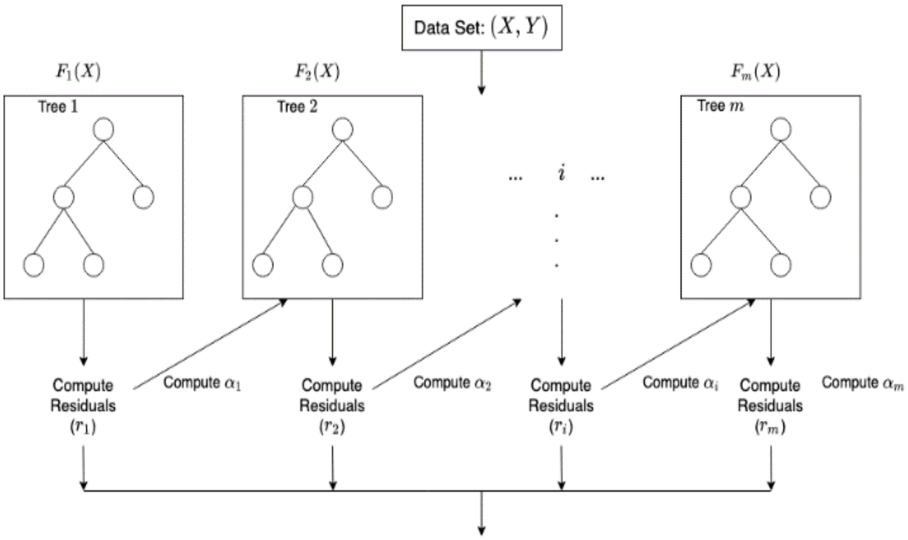
Figure 3. Random forest classifier

In the above example, three individual decision trees which together forms a Random Forest. This can be used for ensemble learning. This learning algorithm is used to solve complex problem using multiple classifiers. Helps to create more accurate results by using multiple models to come to its conclusion. This process improves the performance of the algorithm [6]. The large number of trees generated in the forest leads to high accuracy [22] since it is looking at the results of many different decision trees and finding an average. The single decision tree is very sensitive to data variations. It can easily overfit to noise in the data. When we add tress to the Random Forest then the tendency to overfitting will decrease.

Step 1: select K no of Random data sets from training set
Step 2: Construct the decision trees with selected data
Step 3: Choose the number N to construct the decision trees
Step 4: Repeat step 1 and 2
Step 5: For all new data, find the forecasts of each newly constructed tree.

3.3 Support vector machine algorithm

Another model of supervised learning algorithm in Machine learning which supports both classification and regression analysis [22]. The goal of this algorithm is to draw the best decision boundary that segregates the n dimensional space into different classes. SVM works relatively well when there is a clear margin of separation between classes. This algorithm is more effective in high dimensional spaces. It is effective in some cases, where the number of dimensions is greater than the number of samples.



$$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$$

where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

Figure 4. Support vector machine algorithm

3.4 XGBoost algorithm

XGBoost algorithm is a decision tree-based ensemble Learning Algorithm which uses Extreme Gradient Boosting Framework [4]. This algorithm uses a parallel tree boosting to solve many data science problems. For very small to medium size datasets decision tree classifier is good. But for larger data sets XGBoost performs very well and also minimizes the errors in sequential models [23]. It attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models. When we use Gradient Boosting for regression, the weak learners are regression trees, and each regression tree maps input data point to one of its leaves that contains a continuous score. It minimizes a regularized (L1,L2) objective function that combines a convex loss function(calculated on the difference between target and predicted outputs) and a penalty from a model complexity. The training process works iteratively by adding new trees which predict residual or errors of prior trees thar are then combined with previous trees to take a final decision. So it is called Gradient boosting, since it is minimising the loss when adding new models. A brief illustration of how this algorithm works is shown in Figure 4.

4. METHODOLOGY

4.1 Dataset collection and processing

To detect phishing, most of the researchers use datasets developed by themselves. By using such kind of datasets, it is very difficult to assess and compare the performance of data set models with other models developed with different algorithms. In this research we have used a dataset downloaded recently from Kaggle which consists of approximately 11,550 website names. These websites are pre classified as legitimate websites (non phishing URLs) and Phishing websites which are not legitimate by testing each URL with 30 different features. Out of which 5423 URLs are legitimate means trusted web sites, and the remaining 6127 URLs are Phishing URLs. The input data set is preprocessed using correlation detection. The 30 different adopted features are listed below.

Table 2. Features used to predict phishing URLs

S.No	Feature Type	Value Range	S.No	Feature Type	Value Range
1	index	{-1,1}	16	Links_in_each_tags	{-1,1}
2	IP_address	{-1,1}	17	Anchor_tag_URL	{-1,0,1}
3	Length of URLs	{-1,1}	18	Redirection	{0,1}
4	URL shortening service	{-1,1}	19	Abnormal_URLs	{-1,1}
5	Redirection using double slashes	{-1,1}	20	on_mouseovers	{-1,1}
6	Using @symbol	{-1,1}	21	Statistical_reports	{-1,1}
7	Prefix and Suffixes of the domain	{-1,1}	22	Page links	{-1,0,1}
8	Based on the sub domain part	{-1,1}	23	Google_Index_values	{-1,1}
9	Final state value in SSL	{-1,1}	24	Page_Rank	{-1,1}
10	registeration_length_of_domain	{-1,1}	25	web_traffics	{-1,0,1}
11	Favicon	{-1,1}	26	DNS_Record	{-1,1}
12	port	{-1,1}	27	age_of_domain_page	{-1,1}
13	HTTPS_token_Value	{-1,1}	28	Iframe	{-1,1}
14	URL_Request	{-1,1}	29	pop_up_window	{-1,1}
15	Submitting_to_email	{-1,1}	30	Right_click	{-1,1}

4.2 Feature selection and ranking

Features that are used to train the Machine Learning models have great impact on the performance. In this work authors have adopted 30 different features to train the model and are listed in Table 2. Features that are not relevant to the model will affect the performance of the model. There are several techniques that can be used for feature selection and ranking like univariate selection, principal component analysis and recursive feature elimination.

5. EXPERIMENTAL RESULTS

To measure and compare the efficiency of the Machine Learning algorithms in predicting phishing URLs, authors have used the following parameters: precision, accuracy, recall, macro average and weighted average of the different models.

Precision identifies that how many numbers of websites are properly predicted as phishing websites.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall measures the number of phishing websites which are correctly identified by the model. This is also known as sensitivity.

$$\text{TPR} = \text{sensitivity} = \text{Recall} = \frac{TP}{TP + FN} = 1 - FPR$$

Accuracy finds the percentage of URLs that are correctly predicted.

$$\text{Accuracy} = \frac{\#Correct\ predictions}{\#\ Total\ Predictions}$$

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

F1Score is the Harmonic average of recall and precision parameters which takes values between 0 and 1.

$$F_1 Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

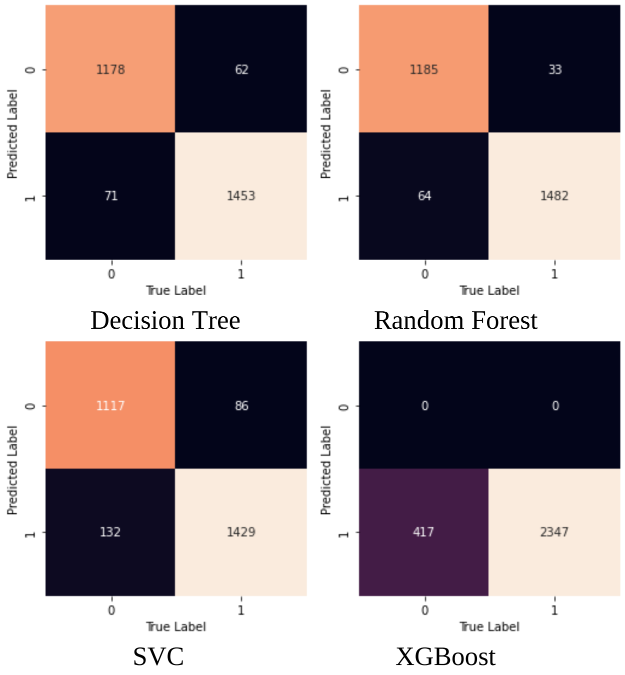
Micro average averages the unweighted mean per each label, where weighted average averages the support-weighted mean per label. The range of micro average values for the input data set is 0.84 to 0.96.

All measurements that are used here are functions of the confusion matrix. The general structure of confusion matrix is given in Table 3, used to measure performance of chosen classification model.

Table 3. Confusion matrix

	Predicted positive Class	Predicted Negative Class
Original Positive Class	True Positive (TP)	False Negative (FN)
Original Negative Class	False Positive (FP)	True Negative (TN)

where, TP is a true positive where the model correctly predicts the URL as phishing URL TN is a true negative which wrongly classified the URL as benign. FP is a class where a website is wrongly classified as phishing website, and finally FN is a class which wrongly classified the URL as benign URL. All proposed algorithms are implemented in Python. Confusion matrix of each algorithm is given in Figure 5.



Decision Tree Random Forest

SVC XGBoost

Figure 5. Confusion matrix of ML algorithms

The experiment is carried out on HP machine with 64bit operating system with 16GB RAM. After testing the dataset using 30 different features, the results are tabulated in Table 4 and the corresponding chart is given in Figure 6.

Table 4. Performance analysis of ML algorithms

	Decision Tree	Random Forest	SVM	XGBoost
Precision	0.96	0.95	0.89	0.96
Recall	0.95	0.97	0.93	0.97
f1-score	0.95	0.96	0.91	0.97
accuracy	0.95	0.97	0.92	0.97
Macro avg	0.95	0.97	0.92	0.95
weighted avg	0.95	0.97	0.92	0.95

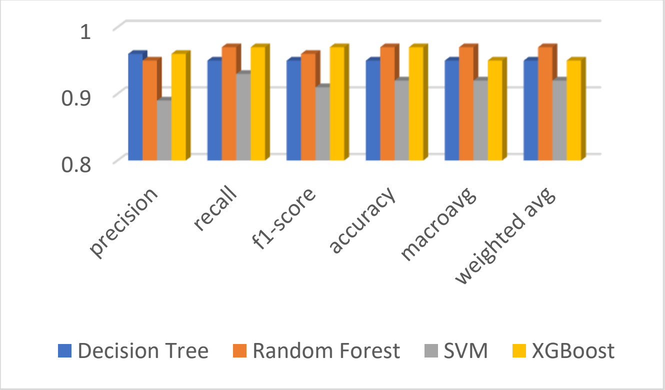


Figure 6. Graphical representation of performance analysis where y-axis represents the score range for each measure

6. CONCLUSION

In this research work, we have taken a dataset with 11550 URLs and is tested against 4 most popular Machine Learning algorithms Decision Tree, Random Forest, SVM and XGBoost and their performance is analyzed with respect to the six different parameters precision, recall, f1-score, accuracy, macro average and weighted average. Radom Forest and XGBoost algorithms have shown most promising results. Their accuracy is very high compared with the other algorithms.

REFERENCES

[1] Mohammad, R., Thabtah, F., McCluskey, L. (2015). Tutorial and critical analysis of phishing websites methods. Computer Science Review, 17: 1-24. https://doi.org/10.1016/j.cosrev.2015.04.001

[2] Gupta, B.B., Tewari, A., Jain, A.K., Agrawal, D.P. (2017). Fighting against phishing attacks: State of the art and future challenges. Neural Computing and Applications, 28(12): 3629-3654. https://doi.org/10.1007/s00521-016-2275-y

[3] http://toolbar.netcraft.com/, accessed on 12 March 2022.

[4] Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B., Si, Y. (2018). A data-driven design for fault detection of wind turbines using random forests and XGboost. IEEE Access, 6: 21020-21031. https://doi.org/10.1109/ACCESS.2018.2818678

[5] Fette, I., Sadeh, N., Tomasic, A. (2007). Learning to detect phishing emails. WWW '07: Proceedings of the 16th International Conference on World Wide Web, pp. 649-656. https://doi.org/10.1145/1242572.1242660

[6] Akinyelu, A.A., Adewumi, A.O. (2014). Classification of phishing email using random forest machine learning technique. Journal of Applied Mathematics, 2014: 1-6. https://doi.org/10.1155/2014/425731

[7] Sonowal, G. (2022). What Does a Phishing URL Look Like? In: Phishing and Communication Channels. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7744-7_4

[8] Aburrous, M.., Hossain, M., Dahal, K., Thabtah, F. (2010). Experimental case studies for investigating E-banking phishing techniques and attack strategies. Journal of Cognitive Computation, Springer Verlag, 2(3): 242-253. https://doi.org/10.1007/s12559-010-9042-7

[9] Salzberg, S.L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Machine Learning, 16: 235-240. https://doi.org/10.1007/BF00993309

[10] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S. (2007). A comparison of machine learning techniques for phishing detection. eCrime '07: Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, pp. 60-69. https://doi.org/10.1145/1299015.1299021

[11] Sahingoz, O.K., Buber, E., Demir, O., Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117: 345-357. https://doi.org/10.1016/j.eswa.2018.09.029

[12] Jain, A.K., Gupta, B.B. (2019). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing, 10(5): 2015-2028. https://doi.org/10.1007/s12652-018-0798-z

[13] Gandotra, E., Gupta, D. (2021). An efficient approach for phishing detection using machine learning. In: Giri, K.J., Parah, S.A., Bashir, R., Muhammad, K. (eds) Multimedia Security. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-8711-5_12

[14] Abdelhamid, N., Thabtah, F., Abdel-jaber, H. (2017). Phishing detection: A recent intelligent machine learning comparison based on models content and features. 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 72-77. https://doi.org/10.1109/ISI.2017.8004877

[15] Khonji, M., Iraqi, Y., Jones, A. (2013). Phishing detection: A literature survey. IEEE Communications Surveys & Tutorials, 15(4): 2091-2121. https://doi.org/10.1109/SURV.2013.032213.00009

[16] Bright, M. (2011). MillerSmiles. Online phishing Scams Available at: http://www.millersmiles.co.uk/, accessed on 12 March 2022.

[17] Tan, C.L., Chiew, K.L., Sze, S.N. (2017). Phishing webpage detection using weighted URL tokens for identity keywords retrieval. In: Ibrahim, H., Iqbal, S., Teoh, S., Mustaffa, M. (eds) 9th International Conference on Robotic, Vision, Signal Processing and Power Applications. Lecture Notes in Electrical Engineering, vol 398. Springer, Singapore. https://doi.org/10.1007/978-981-10-1721-6_15

[18] Bouckaert, R.R. (2004). Bayesian network classifiers in Weka. Working paper series. University of Waikato, Department of Computer Science. No. 14/2004. Hamilton, New Zealand: University of Waikato.

[19] Gaines, B.R., Compton, P. (1995). Induction of ripple-down rules applied to modeling large databases. Journal of Intelligent Information Systems, 5(3): 211-228. https://doi.org/10.1007/BF00962234

[20] Abdelhamid, N., Ayesh, A., Thabtah, F. (2014). Phishing detection based Associative Classification data mining. Expert Systems with Applications, 41(13): 5948-5959. https://doi.org/10.1016/j.eswa.2014.03.019

[21] Basnet, R., Mukkamala, S., Sung, A.H. (2008). Detection of phishing attacks: A machine learning approach. In: Prasad, B. (eds) Soft Computing Applications in Industry. Studies in Fuzziness and Soft Computing, vol 226. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-77465-5_19

[22] Toolan, F., Carthy, J. (2019). Phishing detection using classifier ensembles. 2009 eCrime Researchers Summit, pp. 1-9. https://doi.org/10.1109/ECRIME.2009.5342607

[23] Kaytan, M., Hanbay, D. (2017). Effective classification of phishing web pages based on new rules by using extreme learning machines. Computer Science, 2(1): 15-36.