Performance Analysis of Boosting Techniques for Classificationand Detection of Malicious Websites

W. Regis Anne [1], S. CarolinJeeva[2]
{wra.amcs@psgtech.ac.in[1], caroljeeva@gmail.com[2]}

[1] Assistant Professor, Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore, India., [2] Associate Professor, Department of Digital Sciences Karunya Institute of Technology and Sciences, Coimbatore, India.

Abstract. Phishing is a method of social engineering technique to deceive web users to capture sensitive information like user name and password in websites without the knowledge of the end user. The end user provides information about their personal and financial thinking it's the authenticated service provider. URL meaning the "Uniform Resource Locator" that identifies an address to a file in the server. The URLs can be categorized as benign or malicious. Malicious URLs are created for the purpose of attacking to create loss and poses great threat to the victims. Machine Learning approaches offer a wide range of algorithms to detect malicious websites. It considers the URL as a set of features of Lexical, Host based and Content features to train a model to classify it as malicious or benign. Boosting is a collection of algorithms that combine the weaklearning classifiers to build strong Classifiers. In this paper boosting algorithms are exploited to the study of URL detection as malicious or benign. Boosting algorithms such as LGBM, XGBoost and Gradient Boosting are used for predicting phishing URL is presented. Feature selection to identify the important features is performed. The selected features are then classified by Random Forest Classifier to give an accuracy of 99%.

Keywords: Malicious, Benign, Machine learning, Boosting, Cyber Security, LGBM, XGBoost and Gradient Boosting, Accuracy, Precision, Recall and Support.

1 Introduction

Phishing is a method of social engineering technique to deceive web users to capture sensitive information like user name and password in websites without the knowledge of the end user. The end user provides information about their personal and financial thinking it's the authenticated service provider. At present, the Internet is the daily way of life for everyone. Internet services are used to communicate and to perform mission critical system for various businesses. As a result, the cyber-crimes have augmented and thereby security companies are developing new techniques to protect their assets from the hackers. A phisher creates a fake webpage that resembles the legitimate webpage and thereby probe the user to enter the sensitive details like user name, password and it is transferred to the hacker's server. URL meaning the "Uniform Resource Locator" that identifies an address to a file in the server. The URLs can be categorized as benign or malicious. Malicious URLs are created for the purpose of attacking to create loss and poses great threat to the victims. Spamming, phishing, denial of service, malware, attack page and SQL injection are categories of malicious attack. Benign URLs are associated with webpage that does not cause a phishing attack.

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314506

Machine Learning approaches offer a wide range of algorithms to detect malicious websites. It considers the URL as a set of features of Lexical, Host based and Content features to train a model to classify it as malicious or benign. The lexical features include the features from the URL string. Host based features consist of the properties from the name of the host URL. Content features include the content of the downloaded webpage including the script and the elements of the webpage. Boosting Machine Learning technique is an ensemble method to build a strong classifier by combining weak classifiers. Multiple trees are built on different sets of training data and the predictions from several trees are combined by means of voting to provide better prediction with minimum error. Boosting ensures that its performance will not be worse than the best of the base learners. Boosting was first initiated by Freund and Schapire in the year 1997 [1] with AdaBoostclassifier. From the time when, it has been aestablished technique for resolving classification problems. In this paper, the ISCX-URL2016 dataset is considered. First the boosting algorithms are exploited to predict different classes of URLs. Second, the important features are selected using the boosting algorithms. Thirdly the identified features are classified using Random Forest classifier. The algorithms performance is measured using Accuracy, Precision, Recall and Support.

2. Literature Survey

Atharva Deshpande et. al [2] presented a model to detect the phishing website where the features related to address bar, anomaly, HTML and Javascript and domain based are extracted to determine the originality of the website. IshantTyangiet. al [3] proposed machine learning approach with a prediction accuracy of 98.4%. Sonowal et al. [4] proposed predictive black list-based technique to identify phishing websites. This approach detects the new phishing URLs using an matching algorithm. An approach [5] proposed where the features are collected by suggestion from the search engine Google, ranking of pages and URL patterns that are suspicious. A rule-based approach proposed by Mohammed et. al [6] where the relationship between the web content of the page and the URL of a page are extracted.

Mustafa et. al [7] proposed an approach to extract the URL features and subset-based feature selected is carried out. A light weight-based approach is proposed in detecting phishing URL by mohammed et.al [8]. An approach based on a weighted token in URL is proposed [9] by Tan et. al. Identity keyword phrases are extracted as signatures from the webpage. The experiment results achieved 99.20% true positive and 92.20% true negative.

An approach that uses search engine proposed by Huh et. al [10] for phishing detection but in a lighter way. The full URL string is given to search engine thereby reducing the keyword extraction method followed in the above-mentioned search engine-based approach. The proposed technique by Jain A.K and Gupta [11] uses automatic update of legitimate websites and warns the online user for the availability of the. The legitimacy of the webpage is accessed based on 1) IP address and domain matching module, 2) Hyperlinks features from source code. An approach by lee et.al [12] proposed where the features are gathered by Google's results, ranking of the webpage and wary URL features. Liqun et al. [13] proposed NIOSELM approach [3] that extracts surface level feature, topological and inheritance of the websites are used in detecting phishing webpages. CANTINA+ by Xiang et al. [14] encompasses the Document Object Model, third party and Google search engines to identify phishing web pages using machine learning. Zhang et al. [15] presented a method that examines the source code of a web site and makes use of Term Frequency and Inverse Document Frequencyto locate the maximum rating keywords. The key phrases acquired are given as input to the search engine to detect whether or not the URL fits with N top seek end result and is considered as legitimate.

3. Boosting Algorithms for Classification

Boosting Machine Learning technique is an ensemble method to build a strong classifier by combining weak classifiers. Multiple trees are built on different sets of training data and the predictions from several trees are combined by means of voting to provide better prediction with minimum error. In this paper, we will explore Boosting Machine Learning classifiers namely Light Gradient Boosted Machine (LGBM),eXtreme Gradient Boosting (XGBoost) and Gradient Boosting Machines.In this paper, we classify the raw URLs into different class types such as benign or spam, phishing, malware or defacement URL.

3.1 Light Gradient Boosted Machine Classifier

The LGBM classifier selects the features automatically and boosts examples with larger gradients resulting in faster training and accurate predictions. In the architecture of LGBM classifier, the tree grows leaf wise and chooses the leaf whose delta loss is maximum. The modelling rate controls the growth of selecting the gradients according to the learning graph. The learning rate is 0.15. The accuracy is improvised by increasing the number of leaves and building deeper trees. The only disadvantage of this method is it leads to overfitting.

3.2 **eXtreme Gradient Boosting Classifier**

The eXtreme Gradient Boosting (XGBoost) classifier is derived from the Gradient boosting model. The gradient of the loss function is calculated using the “Eq. 1”

$$r_{im} = -\alpha \left[\frac{\delta L\big(y_i, F(x_i)\big)}{\delta F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad (1)$$

wherer_{im} is the pseudo residuals, L is the loss function, m is the iteration number, α denotes the learning rate and F(x) gives the predictions from the model.Pruning and regularization can be done to optimize the algorithm.

3.3 Gradient Boosting Classifier

In Gradient Boosting classifier many weak classifiers are generated and are ensembled to form a strong classifier.First, the data is made to learn with the set weight value and weak classifiers are built sequentially. The residual error of the current model is given as input to the next classifier and the tree is built additively.Then the weights are modified and the models are trained to give accurate prediction of the responsive variable. The model is trained to minimize the error in the classifier. The learning rate and the number of weak learners can be controlled to define the model accuracy. The model can be cross-validated to optimize the correctness and accuracy.The growth function $H(x)$ that bounds the generalization error is given in “Eq. 2”

$$H(x) = sign\left(\sum_{t=1}^{T} \propto_t h_t(x)\right) \qquad (2)$$

where$x$ is equal to sign of the weighted sum of the outputs $h_t(x)$ of T weak classifiers with the weights equal to $(\propto_t)$. Pruning and regularization can be done to optimize the algorithm.

4. Feature Selection and Prediction using Random Forest classifier

The accuracy of the classification of the boosting algorithms can be improvised by means of Feature Selection. Also known as Attribute Selection, it identifies the appropriate parameters that optimize the classification to URLs. Feature selection is performed by the boosting algorithm where each algorithm calculates the feature importance. For each feature, the importance value calculated by each algorithm is combined to find the mean. The feature importance is ordered and the top features are considered as input for the Random Forest classifier. The dimensionally reduced dataset and Random Forest classifier yields better accuracy.

Random Forest is a meta estimator that creates multiple decision trees on different samples of the dataset. The Gini index or Entropy is used to branch on a node to create the tree. This is done by calculation of impurity on classification due to randomness of selection of samples. The Gini index is used to conclude how the nodes are added to the branch of the decision tree and is given in “Eq. 3”. Then finally voting is done to combine all the trees to do final prediction by improving accuracy and to prevent overfitting.

$$Gini = 1 - \sum_{i=1}^{C}(P_i)(3)$$

where P_i denotes the relative frequency of the class and C denotes the total number of classes. The entropy is given in “Eq. 4”

$$Entropy = \sum_{i=1}^{C} -P_i * \log_2(p_i) \quad (4)$$

whereP_i denotes the relative frequency of the class and C denotes the total number of classes.

5.Performance Metrics

Precision is the proportion between the true positive and real positive values i.e.,the number of URLs that have been identified as phished to that of the really phished URL in the dataset. Recall is the proportion of true positive values to the sum of true positive values and true negative values. F1 score has given the stability or weighted average between Precision and Recall. Support is the total number of genuineexistences of the class in the specified dataset. Improper support will lead to inconsistencies of the classifier and would need to reconstruct the training and testing data. Support is static across different models.

6. Experimental setup

In this paper,ISCX-URL2016 [16] dataset is considered that contains more than 1,10,000 URLs . This dataset contains over 35,300 benign URLs, 12,000 spam URLs, 10,000 phishing URLs, 11,500 malware URLs and 45,450 defacement URL. Exploratory data analysis is done on the dataset and the data is cleaned and finally the cleaned dataset contains 36707 URLs and their 80 features. The distributions of values of different features are given in Fig 1.

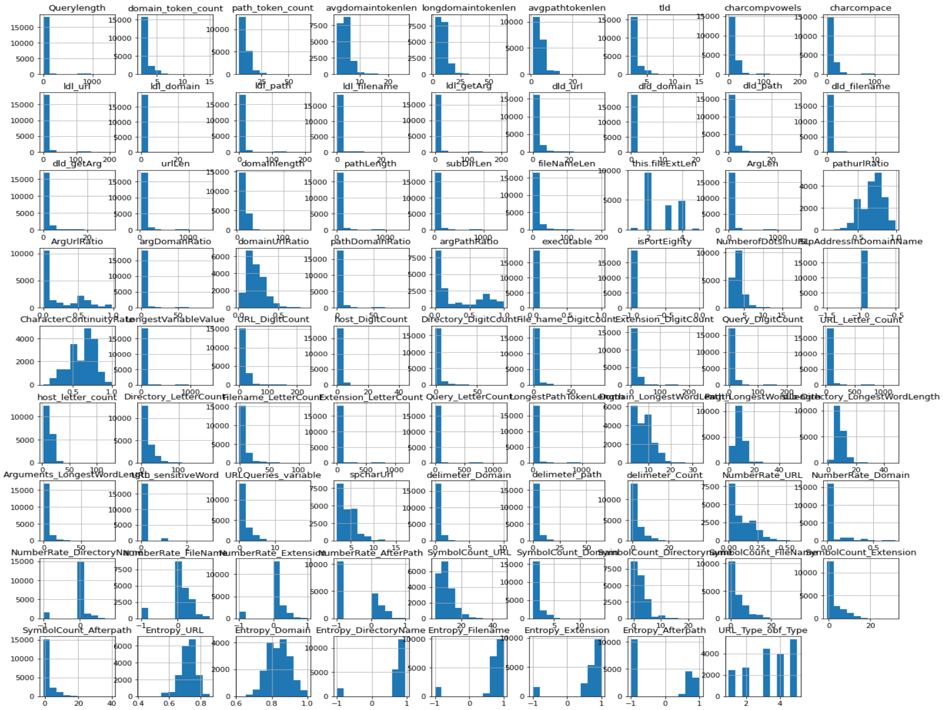

Figure1. Histogram Plot of different values of the 80 different features

Some of the features considered are Querylength, domain_token_count , path_token_count, avgdomaintokenlen, longdomaintokenlen , avgpathtokenlen, LongestPathTokenLength, URLQueries_variable, SymbolCount_Domain, Entropy_DirectoryName and URL_Type_obf_Type etc.,The feature URL_Type_obf_Type contains the different classes defacement, malware, phishing, spam and benign. The Malicious URLs include defacement, malware, phishing and spam classes. The final dataset after cleaning contains 7930 defacement URLs, 7781 benign URLs, 6712 malware URLs, 7586 phishing URLs and 6698 spam URLs. After cleaning the data the dataset contains 2477 defacement URLs, 2709 benign URLs, 4440 malware URLs, 4014 phishing URLs and 5342 spam URLs. This sums upto18982 URLs. The dataset is divided into training set and testing set. The training set contains 15185 URLs and the testing set contains 3797 URLs. The dataset is trained on the training set and its validated on the training set.

7.Results and Discussion

The performance of the algorithms is discussed in this section.The LGBM classifier performance metrics is given in Table 1 with accuracy of 98 %.The value of precision is 0.98 meaning that the LGBM classifier has correctly identified 98% of the maliciousor benign URL correctly. Recall value of 0.98 shows that the LGBM classifier has correctly identified a malicious URL. F1 score has given the stability or weighted average between Precision and Recall with a value of 0.98.Support values for different classes prove the existence of the class in the training dataset has been identified correctly by the classifier.

Table 1. Performance metrics of LGBM Classifier

Class	Precision	Recall	F1-Score	Support
Defacement URL	0.99	0.99	0.99	495
Benign URL	0.97	0.97	0.97	542
Malware URL	0.98	0.98	0.98	888
Phishing URL	0.96	0.96	0.96	803
Spam URL	0.99	0.99	0.99	109
Macro average of the URL Classes	0.98	0.98	0.98	3797
Weighted average of the URL Classes	0.98	0.98	0.98	3797

The XGBoost classifier performance metrics is given in Table 2 with accuracy of 94 %.The value of precision is 0.94 meaning that the XGBoost classifier has correctly identified 94% of the malicious or benign URL correctly. Recall value of 0.94 shows that the XGBoost classifier has correctly identified 94% of the malicious URLs. F1 score has given the stability or weighted average between Precision and Recall with a value of 0.94.

Table 2. Performance metrics of XGBoost Classifier

Class	Precision	Recall	F1-Score	Support
Defacement URL	0.97	0.96	0.96	495
Benign URL	0.92	0.95	0.93	542
Malware URL	0.95	0.93	0.94	888
Phishing URL	0.89	0.9	0.9	803
Spam URL	0.98	0.97	0.98	1069
Macro average of the URL Classes	0.94	0.94	0.94	3797
Weighted average of the URL Classes	0.94	0.94	0.94	3797

The Gradient Boosting classifier performance metrics is given in Table 3 with accuracy of 94.2 %.The value of precision is 0.94 meaning that the Gradient Boosting classifier has correctly identified 94% of the malicious or benign URL correctly. Recall value of 0.94 shows that the Gradient Boosting classifier has correctly identified 94% of the malicious URLs. F1 score has given the stability or weighted average between Precision and Recall with a value of 0.94.

Table 3. Performance metrics of Gradient Boost Classifier.

Class	Precision	Recall	F1-Score	Support
Defacement URL	0.95	0.96	0.96	495
Benign URL	0.93	0.94	0.94	542
Malware URL	0.95	0.93	0.94	888
Phishing URL	0.88	0.9	0.89	803
Spam URL	0.98	0.97	0.98	1069
Macro average of the URL Classes	0.94	0.94	0.94	3797
Weighted average of the URL Classes	0.94	0.94	0.94	3797

The accuracy of the different Boosting Algorithm is given in Fig2. The LGBM classifier outperforms the other entire two algorithms to classify URLs into different classes with an accuracy of 98%.

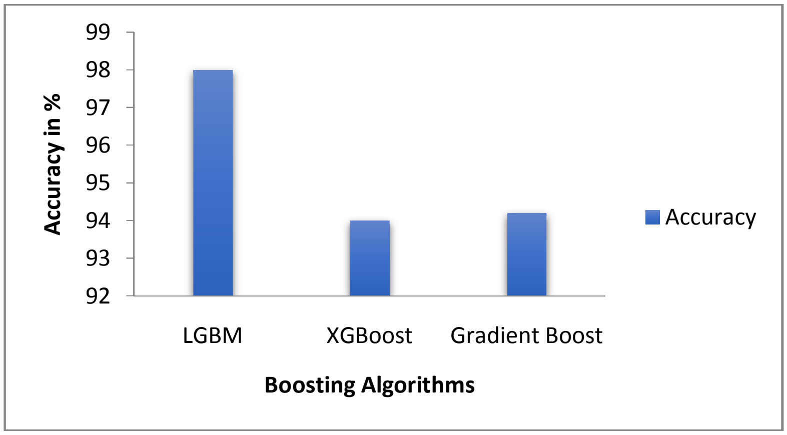

Figure 2. Accuracy of different Boosting Algorithms

The confusion matrix of the LGBM Classifier is given in Fig 3. The confusion matrix displays that the defacement URL is identified as defacement correctly and similarly all other classes are also identified correctly with the spam URL identified perfectly.

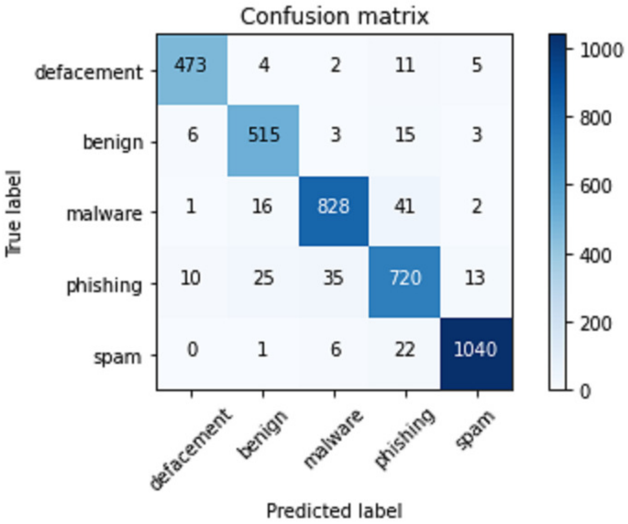

Figure 3. Confusion Matrix of LGBM Classifier

To still improve the accuracy the features that contribute to the detection of malicious and benign URLs can be identified and then the model can be trained using these features. The important features from the dataset are identified using the boosting algorithms. The Table 4 depicts some of the features and the importance scores calculated for all the features. The mean of the importance score calculated from all the three boosting algorithm for each feature is calculated and the top 20 features are identified.

Table 4. Feature Importance of features from the Boosting Algorithms

S.No	**Features**	**Gradient Boost Feature Importances Score**	**XG Boost Feature Importances Score**	**LGBM Feature Importances Score**	**Mean Score**
0	Querylength	0.012149	0.004101	71	23.6721
1	domain_token_count	0.027736	0.040049	348	116.023
2	path_token_count	0.008475	0.005676	89	29.6714
3	avgdomaintokenlen	0.045696	0.019968	650	216.689
4	longdomaintokenlen	0.011145	0.007833	446	148.673
5	avgpathtokenlen	0.019359	0.013348	402	134.011
6	tld	0.024856	0.000142	0.0001	0.00829
7	charcompvowels	0.011278	0.019651	215	71.677
8	charcompace	0.005528	0.001946	167	55.6692
9	ldl_url	0.006866	0.011979	204	68.0063

The Top features ordered by mean value of importance are represented in Fig 4.The defacement, malware, phishing and spam classes are combined to form the Malicious class. The modified dataset contains 13640 Malicious URLs and 5342 benign URLs. This is modelled as binary classification problems. The top features are considered and the dataset is modified. The Random forest classifier model is trained with the modified dataset to predict if it is a malicious or benign URL. The performance metrics is given in Table 5 with the accuracy of 99%. The accuracy is 99% because the features that contribute to finding an URL is malicious or benign is considered for training the model. Other features are excluded. The value of precision is 0.99 meaning that the Random Forest classifier has correctly identified 99% of the malicious or benign URL correctly. Recall value of 0.99 shows that the Random Forest classifier has correctly identified 99% of the malicious URLs. F1 score has given the stability or weighted average between Precision and Recall with a value of 0.99. The support factor shows that the classifier has correctly identified the benign and malicious URL existence in the dataset.

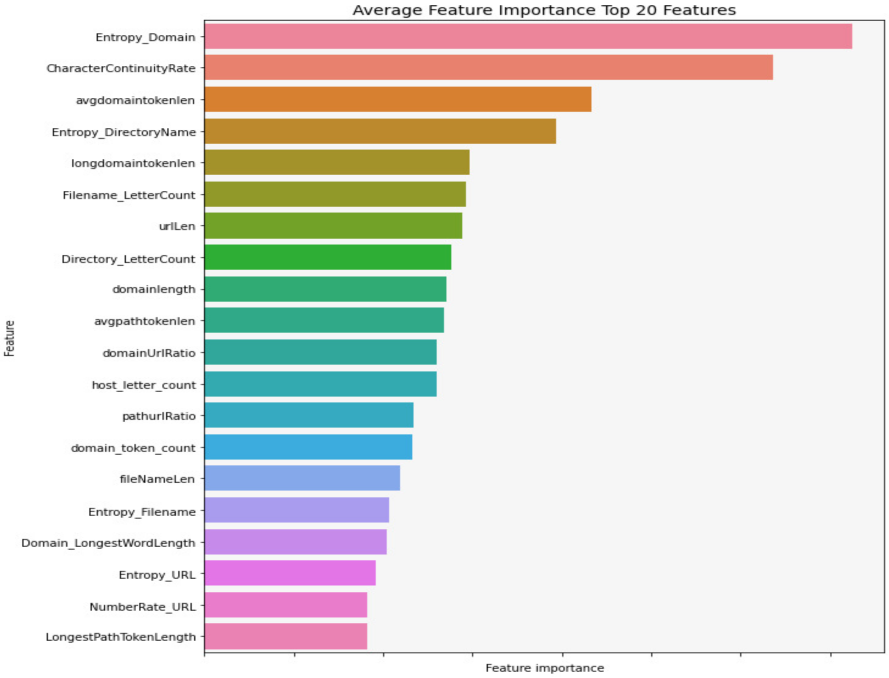

Figure 4. Feature Selection of top 20 features and their importance value

Table 5. Performance metrics of Random Forest Classifier.

Class	Precision	Recall	F1-Score	Support
Benign URL	0.99	0.98	0.99	1048
Malicious URL	0.99	1	0.99	2749
Macro average of the URL Classes	0.99	0.99	0.99	3797
Weighted average of the URL Classes	0.99	0.99	0.99	3797

The confusion matrix is given in Fig 5.The True positive value is 1025 which depicts that the 1025 URL predicted as benign as benign URLs. The True negative value is 2743 which depicts that the 2743 URL is predicted as malicious and it is malicious URL. The wrong prediction is 23 predicted as malicious but is benign and 6 is predicted as benign but labelled as malicious. The feature selection followed by the classifier performs well and gives the accuracy of 99%.

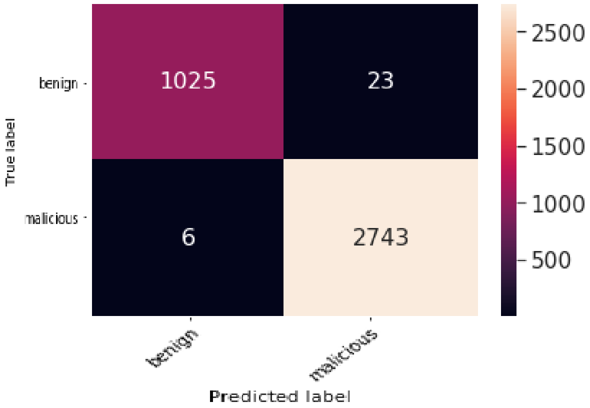

Figure 5. Confusion Matrix of the Random Forest Classifier

8. Conclusion

In this paper, boosting algorithms are exploited to the study of URL detection as malicious or benign. Boosting algorithms such as LGBM, XGBoost and Gradient Boosting for predicting phishing URL is presented. The LGBM performs better than XGBoost and Gradient Boosting. Features are analysed to identify important features that facilitates to detect malicious and benign URLs. Then the Random classifier model is trained with the features selected and the model is trained to give an accuracy of 99%. For future enhancements, deep learning models with larger datasets can be explored with dynamic URL features.

References

[1] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. In Machine Learning: Proceedings of the Fifteenth International Conference, 1998.

[2] Atharva Deshpande , OmkarPedamkar , Nachiket Chaudhary , Dr. SwapnaBorde, 2021, Detection of Phishing Websites using Machine Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 05.

[3] IshantTyagi, Jatin Shad, Shubham Sharma, Siddharth Gaur, Gagandeep Kaur, ‘ A Novel Machine Learning Approach to Detect Phishing Websites’, IEEE conference, 2018, DoI: 10.1109/SPIN.2018.8474040.

[4] Sonowal, G., Kuppusamy, K., (2016) Masphid: A model to assist screen reader users for detecting phishing sites using aural and visual similarity measures. In: Proceedings of the International Conference on Informatics and Analytics pp.87

[5] Huh. J., H. Kim (2011) ‘Phishing detection with popular search engines: simple and effective’, 4th Canada- France MITACS Workshop on Foundations and Practice of Security, Paris, France May 12-13, Springer Verlag , pp. 194-207.

[6] MahmoodMoghimi, Ali YazdainVarjani (2016) ‘New rule based phishing detection method’ Expert Systems with Applications Vol. 53 pp. 231-242.

[7] Mustafa Aydin and Nazife Baykal, "Feature Extraction and Classification Phishing Websites Based on URL", IEEE International Conference on Communications and Network Security (CNS), pp. 769-770, 2015.

[8] Mohammad Saiful Islam Mamun, Mohammad Ahmad Rathore, ArashHabibiLashkari, Natalia Stakhanova and Ali A. Ghorbani, "Detecting Malicious URLs Using Lexical Analysis", Network and System Security, Springer International Publishing, P467--482, 2016.

[9] Tan, C.L., Chiew, K.L., and San Nah Sze (2017) ‘Phishing webpage detection using weighted url tokens for identity keywords retrieval. In: 9th International Conference on Robotic, Vision, Signal Processing and Power Applications, Springer, pp. 133–139.

[10] Huh. J., H. Kim (2011) ‘Phishing detection with popular search engines: simple and effective’, 4th Canada- France MITACS Workshop on Foundations and Practice of Security, Paris, France May 12-13, Springer Verlag , pp. 194-207

[11] Jain, A.K., Gupta, B., (2016) ‘A novel approach to protect against phishing attacks at client side using auto-updated white-list’, EURASIP Journal on Information Security, pp.1–11.

[12] Lee, J. L., Kim, D. H., Chang-Hoon, Lee, (2015) Heuristic-based approach for phishing site detection using url features. in: Third International Conference On Advances in Computing, Electronics and Electrical Technology - CEET.

[13] Liqun Yang, Jiawei Zhang, Xiaozhe Wang, zhi Li, Zhoujin Li, Yueying He (2021) ‘ An improved ELM based and data preprocessing integrated approach for phishing detection considering comprehensive features’, in the Expert Systems with Applications, vol. 165.

[14] Xiang G., Hong J., carolyn p. Rose, Lorrie Cranor (2011) ‘CANTINA +: A feature-rich machine learning framework for detecting phishing web sites’, Article 21 ACM Transactions on Information and System Security, Vol. 5, pp. 1-32.

[15] Zhang Yue, Hong Jason I., Cranor Lorrie F., (2007) ‘CANTINA: A content based approach to detecting phishing web sites’, In: Proceeding of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, pp.639-648.

[16] Mohammad Saiful Islam Mamun, Mohammad Ahmad Rathore, ArashHabibiLashkari, Natalia Stakhanova and Ali A. Ghorbani,, (2016) ‘Detecting Malicious URLs Using Lexical Analysis’, Network and System Security, Springer International Publishing, pp.467-482.