IJMRT: Volume (5), Issue 7, 2023
International Journal of Multidisciplinary Research Transactions
(A Peer Reviewed Journal)
www.ijmrt.in

ISSN (Print):2663-2381
ISSN(Online):2663-4007

Malware and Phishing URL Detection using Machine Learning

Shapna Rani E[1], Anushree A[2]*, Guhan S[3], Nancy Pricilla R[4], Aishvariya B B[5]

[1]*Assistant Professor, Department of Computer Science Engineering, Saranathan College of Engineering, Tamil Nadu, India.*
[2,3,4,5]*Student, Department of Computer Science Engineering, Saranathan College of Engineering, Tamil Nadu, India.*

*Corresponding author

DoI: https://doi.org/10.5281/zenodo.7942437

Abstract

Phishing URLs and malicious software pose a serious risk to both individuals and organizations. These assaults may lead to financial loss, reputational harm, and data breaches. Security experts have created several tools and methods to identify and stop the propagation of malware and phishing URLs to tackle these threats. Using machine learning algorithms to identify malware and phishing URLs is one of the most efficient ways to do so. In order to find trends and traits linked to harmful websites and software, these algorithms analyze historical data. After being trained on this data, a model can then be used to scan emails and web pages for suspicious content.

Keywords: URL, Malicious URL Detection, Feature Extraction, Feature Selection, Machine Learning.

1. Introduction

Due to the increase in internet usage over the past several years, an increasing number of individuals are embracing the web as a platform for doing online business, sharing information, and e-commerce. Cybercrime is a new type of crime that emerged as the Internet expanded. Cybercriminals have a variety of methods for stealing information, and most of them utilize phishing.

Phishing, spear phishing, whaling, and email phishing are just a few of the different types of phishing. Phishing, a method of stealing passwords, was first documented in 1990. Attacks including phishing have increased recently. URL phishing is one such attack. A URL is a website address that identifies where a website is located on a network and how to access it. By accessing the URL, we establish a connection to the server's database, which has a webpage containing all the information related to the website and saves all its details. Malicious and benign URLs are classified into two categories. URL phishing uses malicious URLs, whereas benign URLs are secure and safe. The information on a fake website that a cybercriminal creates will be exactly the same as the information on the real website's absolute URL. Fraud will occur when the user enters their credentials since the URL will show up as an advertisement on other websites. Another method is to send a malicious URL to the user via email. When the user clicks on the URL, a nasty virus is downloaded, giving the cybercriminals access to the information they need to carry out their crimes. In order to identify between malicious and benign URLs, we must extract some characteristics from each. In order to identify harmful URLs, some of their attributes must first be extracted from them, and then these features must be compared to determine whether the URL is malicious or benign.

2. Proposed System

The proposed system for malware and phishing URL detection architecture, which uses machine learning algorithms is an important task for maintaining online security. Therefore, identifying and avoiding malicious URLs is essential for maintaining the safety of computer systems and shielding users from harm. Several techniques, including signature-based techniques, heuristic-based techniques, and machine learning-based techniques, have been proposed in recent years for identifying malicious URLs. Signature-based techniques look for matches by comparing URLs to a database of known malicious URLs. This method works well for identifying known dangerous URLs, but it might not work as well for threats that haven't been observed before. Heuristic-based techniques examine URL properties to find potential dangers. This technology can identify threats that had not yet been seen, making it more versatile than signature-based techniques, but it may also produce false positives. A dataset of known malicious and benign URLs is used to train algorithms in machine learning-based methods in order to discover patterns and features that distinguish between the two. This method can detect risks that were not previously known to exist and can also change as the threat environment does. The necessity to properly choose features and create suitable algorithms presents the fundamental difficulty in developing machine learning-based approaches for malicious URL detection. A variety of different methods, including neural networks and decision trees, can be used to assess these features, which can include domain-based features, content-based features, and network-based features.

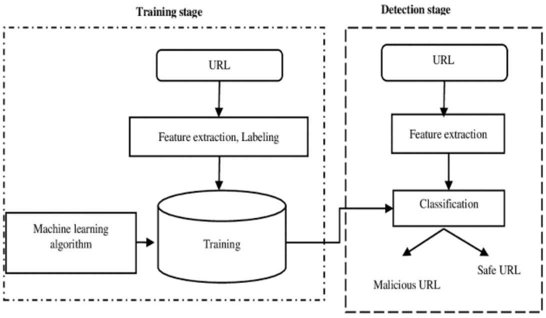


Figure.1. Block Diagram

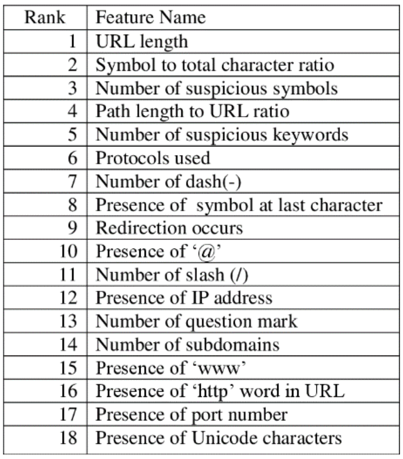
2.1. Collection of Dataset

The dataset used in this system has been fetched from Kaggle. The dataset contains 11000+ data which are to be trained and analyzed.

2.2. Pre-Processing

Only the lexical features that were extracted from the URL are the focus of this paper. We simply need to access application header data online for this. Because we need to access the packet payload in order to retrieve web content, features taken from web page content are not taken into account. However, because the network's packet payload is so large, it is impossible to process and search such a volume of data in real-time or offline. Because of this, we only focus on URLs whose size is significantly smaller than that of web pages. URLs are used to retrieve various characteristics that other researchers have employed. To choose only the crucial traits from them, a feature selection technique is used. SVM is utilized in this method to analyze performances using all possible feature combinations. Each feature's ranking is assessed based on how well it can correctly categorize URLs. Table I displays the attributes along with their rankings. The length of the URLs ranks highest among all the characteristics, as indicated in Table I.

Table.1. URLs Ranks



Rank	Feature Name
1	URL length
2	Symbol to total character ratio
3	Number of suspicious symbols
4	Path length to URL ratio
5	Number of suspicious keywords
6	Protocols used
7	Number of dash(-)
8	Presence of symbol at last character
9	Redirection occurs
10	Presence of '@'
11	Number of slash (/)
12	Presence of IP address
13	Number of question mark
14	Number of subdomains
15	Presence of 'www'
16	Presence of 'http' word in URL
17	Presence of port number
18	Presence of Unicode characters

2.3 Performance Metrics

Parameters used to gauge how well the aforementioned models performed:

True Positive (TP): The proportion of phishing URLs that the classifier accurately identified as such.

True Negative (TN): The proportion of non-phishing URLs that the classifier properly identified as such.

False Positive (FP): The proportion of legitimate URLs that the classifier mistakenly identifies as phishing URLs.

False Negative (FN): The proportion of phishing URLs that the classifier correctly identified as non-phishing URLs.

The aforementioned factors are used to evaluate TPR, TNR, FPR, and FNR in percentage (%). Using the following formulae, accuracy, recall (which is equivalent to TPR), precision, and F-Score are assessed.

$$\text{Accuracy, ACC} = \frac{(\text{TP+TN})}{(\text{TP+TN+FP+FN})} * 100\% \quad \text{-----------------------------------} \ (1)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} * 100\% \quad \text{-----------------------------------} \ (2)$$

$$\text{Precision} = \frac{TP}{(TP+FP)} * 100\% \quad \text{-----------------------------------} \ (3)$$

$$\text{F-Score} = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad \text{-----------------------------------} \ (4)$$

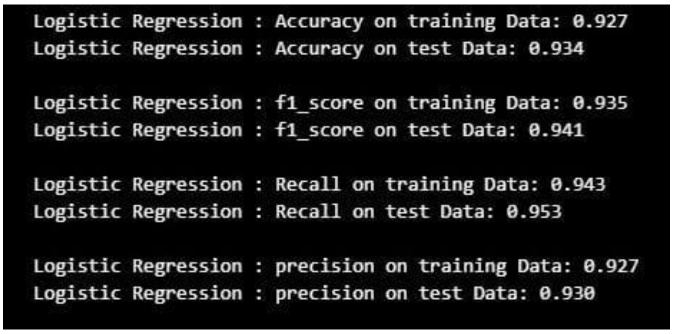
2.4. Data Visualization

Security professionals can learn more about the behavior of malware and phishing URLs through the use of data visualization tools. They can spot data abnormalities, trends, and patterns that might be signs of an assault. Hidden links between data points can be found via data visualization, which may not be possible with more conventional data analysis techniques. The capacity to immediately spot outliers and anomalies is one of the main advantages of data visualization in malware and phishing URL identification. Network graphs, heat maps, and other visualization tools can be used to draw attention to the unusual activity that could be a sign of malware or phishing activities. This can make it easier for security professionals to react to suspected threats promptly and limit future harm.

3. Machine Learning Algorithms

3.1. Logistic Regression

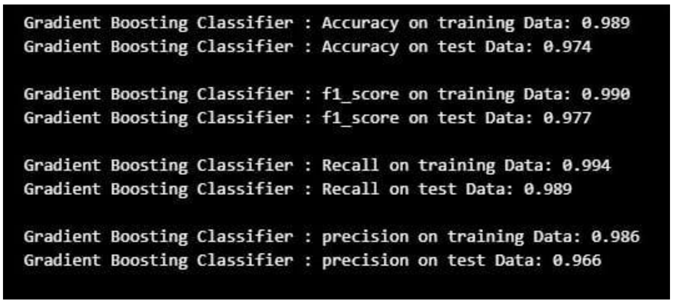
According to a collection of input features (such as URL length, the number of subdomains, etc.), logistic regression models the likelihood that a URL belongs to a specific class (such as malware or benign), depending on that URL's characteristics. The logistic regression algorithm maps the input features to the output probability using a sigmoid function. A value between 0 and 1 is generated by the sigmoid function, and this value can be understood as the likelihood that the URL belongs to the positive class (such as malware). The algorithm then calculates a cost function, which gauges the discrepancy between the anticipated probability and the actual label, using the input attributes and output probability. By altering the weights of the input features, logistic regression seeks to minimize the cost function.



```
Logistic Regression : Accuracy on training Data: 0.927
Logistic Regression : Accuracy on test Data: 0.934

Logistic Regression : f1_score on training Data: 0.935
Logistic Regression : f1_score on test Data: 0.941

Logistic Regression : Recall on training Data: 0.943
Logistic Regression : Recall on test Data: 0.953

Logistic Regression : precision on training Data: 0.927
Logistic Regression : precision on test Data: 0.930
```

3.2. Gradient Boost Classifier (GBC)

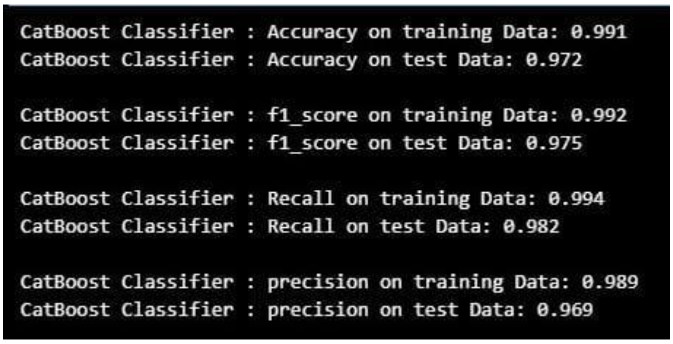
In order to build a more robust model that can reliably categorize URLs as dangerous or benign, Gradient Boosting Classifier (GBC) combines numerous decision trees. A portion of the training data is used by GBC in the first iteration to build a single decision tree. The training data is used by this decision tree to produce predictions, and the next iteration gives more weight to the misclassified data points. A new decision tree is generated using the updated weights in each succeeding iteration, and the process is repeated up to a predetermined number of trees are created or the model's accuracy reaches a plateau. The GBC method combines the predictions from each decision tree to produce the final prediction after they have all been produced. The forecasts from each decision tree are combined using a weighted total, with the weights depending on the accuracy of each tree. The GBC algorithm's final prediction, which is used to categorize the URL as malicious or benign, is the result. Last but not least, the GBC algorithm creates several decision trees, modifies the training data weights based on the classification accuracy, and combines the predictions of the decision trees to provide a final prediction. By using this technique, the GBC algorithm is able to develop a robust and precise model for identifying malware and phishing URLs.



```
Gradient Boosting Classifier : Accuracy on training Data: 0.989
Gradient Boosting Classifier : Accuracy on test Data: 0.974

Gradient Boosting Classifier : f1_score on training Data: 0.990
Gradient Boosting Classifier : f1_score on test Data: 0.977

Gradient Boosting Classifier : Recall on training Data: 0.994
Gradient Boosting Classifier : Recall on test Data: 0.989

Gradient Boosting Classifier : precision on training Data: 0.986
Gradient Boosting Classifier : precision on test Data: 0.966
```

3.3. Catboost Classifier

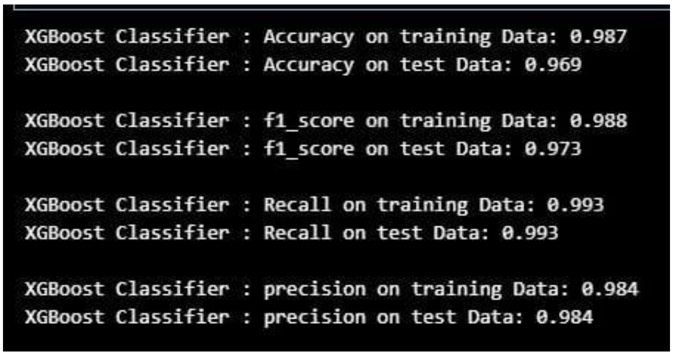
CatBoost is a gradient-boosting approach for machine learning that uses decision trees as its base models. The CatBoost method can automatically handle missing data and is built to handle category features. In order to use CatBoost for malware and phishing URL detection, you would first gather a dataset of URLs that were either classified as malware or not, or as phishing or not. The data would next undergo preprocessing to extract features like the URL's length, the existence of particular keywords, and the number of subdomains. correlations and patterns connected to phishing or malware. Finally, you would assess the classifier's performance on the test set to determine how well it generalizes to fresh data. Metrics like accuracy, precision, recall, and F1 score can be used to assess the classifier's performance. All things considered, the CatBoost classifier is a potent machine-learning method that may be used to identify malicious software and phishing URLs. With the right training and evaluation, it can achieve good accuracy and generalization performance and is particularly well-suited for handling categorical characteristics and missing data.



```
CatBoost Classifier : Accuracy on training Data: 0.991
CatBoost Classifier : Accuracy on test Data: 0.972

CatBoost Classifier : f1_score on training Data: 0.992
CatBoost Classifier : f1_score on test Data: 0.975

CatBoost Classifier : Recall on training Data: 0.994
CatBoost Classifier : Recall on test Data: 0.982

CatBoost Classifier : precision on training Data: 0.989
CatBoost Classifier : precision on test Data: 0.969
```

3.4. Extreme Gradient Boosting (XGBOOST)

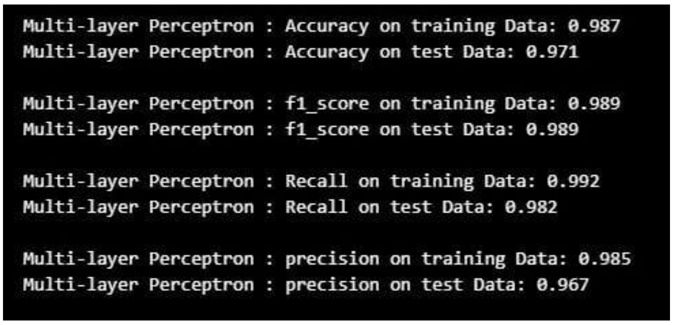
A decision tree-based ensemble learning technique called XGBoost (eXtreme Gradient Boosting) can handle complex datasets and boost prediction accuracy. In order for XGBoost to function, a sizable dataset of known fraudulent and phishing URLs must first be trained on. It then creates a series of decision trees using this training data, each of which predicts whether or not a given URL is dangerous. In order to prioritize the data points that are more challenging to categorize, XGBoost gives the data points weights during the training phase. Gradient boosting is also used to enhance prediction accuracy over time, with each iteration of the algorithm modifying the weights and perfecting the model. After being trained, the XGBoost classifier may be used to categorize new URLs as harmful or non-malicious based on the characteristics that it has come to identify with each class. Analysts may better comprehend the characteristics of malware and phishing URLs by using the algorithm, which can also discover the key criteria that lead to the classification of URLs. Overall, XGBoost is a strong and well-liked classifier for phishing and malware URL detection since it can manage complex datasets and increase prediction accuracy.



```
XGBoost Classifier : Accuracy on training Data: 0.987
XGBoost Classifier : Accuracy on test Data: 0.969

XGBoost Classifier : f1_score on training Data: 0.988
XGBoost Classifier : f1_score on test Data: 0.973

XGBoost Classifier : Recall on training Data: 0.993
XGBoost Classifier : Recall on test Data: 0.993

XGBoost Classifier : precision on training Data: 0.984
XGBoost Classifier : precision on test Data: 0.984
```

3.5. Multi – Layer Perceptron (MLP)

The detection of malware and phishing URLs can be done using a multi-layer perceptron (MLP) classifier, a particular kind of neural network. A variety of input features, including the length of the URL, the existence of particular keywords, and the number of subdomains, are fed into the MLP classifier and then passed via a number of hidden layers. The input features are processed by a group of neurons in each hidden layer, which then passes on its output to the subsequent layer. The output layer, the last layer in the MLP classifier, generates a binary classification output that indicates whether the URL is harmful or benign. In order to reduce the discrepancy between the anticipated output and the actual output, the MLP classifier modifies the weights and biases of the neurons in each layer during training

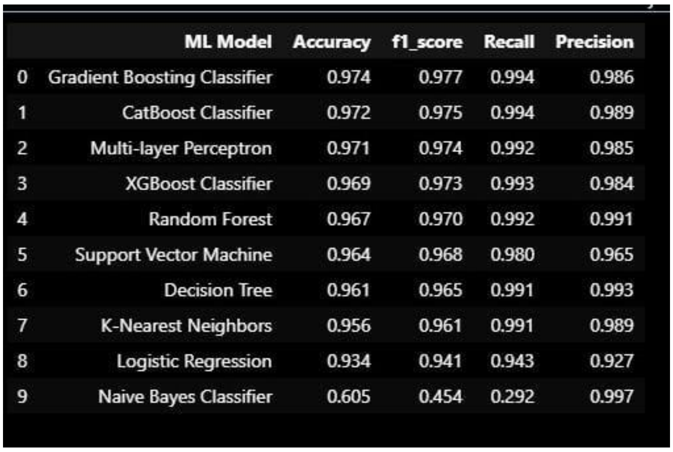


```
Multi-layer Perceptron : Accuracy on training Data: 0.987
Multi-layer Perceptron : Accuracy on test Data: 0.971

Multi-layer Perceptron : f1_score on training Data: 0.989
Multi-layer Perceptron : f1_score on test Data: 0.989

Multi-layer Perceptron : Recall on training Data: 0.992
Multi-layer Perceptron : Recall on test Data: 0.982

Multi-layer Perceptron : precision on training Data: 0.985
Multi-layer Perceptron : precision on test Data: 0.967
```

4. Model Comparison

The development of a powerful malware and phishing URL detection system requires a comparison of several machine learning models. A number of methods, such as accuracy, precision, recall, F1 score, and ROC curve analysis, can be used to compare models.

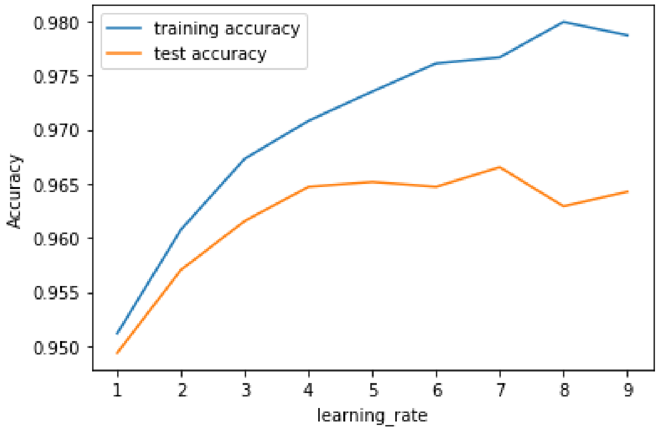
Precision is a measure of how frequently the model predicts a positive class when it is truly a positive class, whereas accuracy is a measure of how frequently the model predicts the class of a URL accurately. A recall is a metric for how frequently the model predicts a positive class accurately when it is in fact positive. A score that combines recall and precision is known as the F1 score. ROC curve analysis is a visual method for contrasting the effectiveness of various models. For various threshold settings, the ROC curve shows the true positive rate against the false positive rate. The effectiveness of the model can be evaluated using the area under the ROC curve (AUC). Better performance is indicated by a higher AUC. Precision is a measure of how frequently the model predicts a positive class when it is truly a positive class, whereas accuracy is a measure of how frequently the model predicts the class of a URL accurately. A recall is a gauge of how frequently the model predicts a favorable class when it is used.

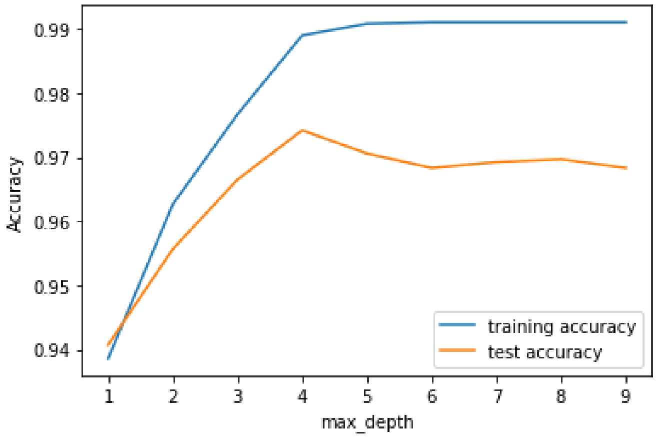


	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Multi-layer Perceptron	0.971	0.974	0.992	0.985
3	XGBoost Classifier	0.969	0.973	0.993	0.984
4	Random Forest	0.967	0.970	0.992	0.991
5	Support Vector Machine	0.964	0.968	0.980	0.965
6	Decision Tree	0.961	0.965	0.991	0.993
7	K-Nearest Neighbors	0.956	0.961	0.991	0.989
8	Logistic Regression	0.934	0.941	0.943	0.927
9	Naive Bayes Classifier	0.605	0.454	0.292	0.997

5. Result and Discussion

Gradient Boosting Classifier currently classifies the URL up to 97.4% of respective classes and hence reduces the chances of malicious attachments. And in the Phishing dataset some features like "HTTPS", "AnchorURL", and "WebsiteTraffic" have more importance to classify URLs as phishing URLs or not. And using Jupyter Notebook helped us to learn a lot about the features affecting the models to detect whether a URL is safe or not, also helped us to learn how to tune models and how they affect the model performance.





6. Conclusion

One effective machine-learning approach for spotting malware and phishing URLs is gradient boosting. A gradient boosting model can be trained to accurately classify URLs as safe or dangerous by extracting features from a huge dataset of labeled URLs. In order to generate a powerful ensemble classifier, the gradient boosting algorithm combines numerous weak classifiers. Decision trees are employed as the weak classifiers, and the model is incrementally improved by the addition of new trees that concentrate on the incorrectly categorized samples. As a result, the model is very accurate, strong, and able to handle large datasets and generalize well to new data. To achieve the highest results, it is crucial to appropriately preprocess the data, extract pertinent features, and carefully adjust the model's hyperparameters while creating a system for malware and phishing URL detection using gradient boosting. In order to keep the model accurate and successful at identifying new and evolving threats, it must also undergo ongoing monitoring and updating.

REFERENCES

[1]. Alsaedi, M.; Ghaleb, F.A.; Saeed, F.; Ahmad, J.; Alasli, M. (2022) ‘Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning’, Cybersecurity Issues in Smart Grids and Future Power Systems, Vol 22, pp. 33-73.

[2]. C. Singh and Meenu (2020) ‘Phishing Website Detection Based on Machine Learning: A Survey’, 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Vol 22, pp. 398-404.

[3]. Gururaj Harinahalli Lokesh & Goutham BoreGowda (2021) ‘Phishing website detection based on effective machine learning approach’, Journal of Cyber Security Technology, Vol 5:1, pp. 1-14.

[4]. Hong, J., Kim, T., Liu, J., Park, N., Kim, SW. (2020). ‘Phishing URL Detection with Lexical Features and Blacklisted Domains. In: Jajodia, S., Cybenko, G., Subrahmanian, V., Swarup, V., Wang, C., Wellman, M. (eds) Adaptive Autonomous Secure Cyber Systems, Vol. 54, No. 1, pp. 55-68.

[5]. J. Acharya, A. Chaudhary, A. Chhabria and S. Jangale (2021), "Detecting Malware, Malicious URLs and Virus Using Machine Learning and Signature Matching," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-5.

[6]. Jang, M., Song, J. & Kim, M. (2022) ‘A Study on the Detection Method for Malicious URLs’ Based on a Number of Search Results Matching the Internet Search Engines Combining Machine Learning. J. Electr. Eng. Technol. Vol 17, pp. 617–626.

[7]. J. H. Ateeq and M. Moreb (2021) "Detecting Malicious URL using Neural Network," 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, Vol. 4, No. 2, pp. 1-8.

[8]. Masurkar, S., Dalal, V. (2021) ‘Enhanced Lightweight Model for Detection of Phishing URL Using Machine Learning’ In Fong, S., Dey, N., Joshi, A. (eds) ICT Analysis and Applications. Lecture Notes in Networks and Systems, vol 154, No. 1, pp. 55-68.

[9]. Preeti, Nandal, R., Joshi, K. (2021) “Phishing URL Detection Using Machine Learning.” In: Hura, G.S., Singh, A.K., Siong Hoe, L. (eds) Advances in Communication and Computational Technology. ICACCT 2019. Lecture Notes in Electrical Engineering, vol 668, No.5, pp.1514-1527.

[10]. R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy and T. Reddy Gadekal (2021), "Malicious URL Detection using Logistic Regression," 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain,2021, pp. 1-6.