Malicious URL Detection and Classification Analysis using Machine Learning Models

Conference Paper · January 2023

DOI: 10.1109/IDCIoT56793.2023.10053422

3 authors, including:

Mohana Mohana
R. V. College of Engineering
74 PUBLICATIONS **1,415** CITATIONS

Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT 2023)
IEEE Xplore Part Number: CFP23CV1-ART; ISBN: 978-1-6654-7451-1

2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) | 978-1-6654-7451-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/IDCIoT56793.2023.10053422

Malicious URL Detection and Classification Analysis using Machine Learning Models

Upendra Shetty D R
Electronics and Telecommunication Engineering
RV College of Engineering®
Bengaluru, Karnataka, India

Anusha Patil
Electronics and Telecommunication Engineering
RV College of Engineering®
Bengaluru, Karnataka, India

Mohana
Computer Science & Engineering (Cyber Security)
RV College of Engineering®
Bengaluru, Karnataka, India.

***Abstract*— One of most frequent cybersecurity vulnerabilities is malicious websites or malicious uniform resource location (URL). Each year, people are losing billions of rupees by hosting gratuitous material (spam, malware, unsuitable adverts, spoofing etc.) and tempting naïve visitors to fall for scams. Email, adverts, web searches, or connections from other websites can all encourage people to visit these websites. Users click on the malicious URL in each instance, a trustworthy system that can categorize and identify dangerous URLs is needed due to rise in phishing, spamming, and malware occurrences. Due to the enormous amount of data, changing patterns and technologies, as well as the complex relationships between characteristics, non-availability of training data, non-linearity and the presence of outliers made classification challenging. In the proposed work, malicious URLs are detected for various applications. Dataset has been categorized into four types i.e., Phishing, Benign, Defacement and Malware. Totally 6,51,191 URLs have been used for proposed implementation. Three machine learning algorithms such as random forest, LightGBM and XGBoost were implemented to detect and classify malicious URLs.**

***Keywords*—URL Detection, Cybersecurity, Machine Learning, URL classification, Phishing, Benign, Defacement, Malware.**

I. INTRODUCTION

The development of online enterprises including e-commerce, social networking, and e-banking is significantly influenced during Covid-19. Unfortunately, sophisticated user exploitation techniques develop along with technological advancements. These attacks typically make use of rogue websites that steal different types of sensitive data that a hacker could utilize. Every URL has two distinguishing parts in it, i.e., identity and the resource name. For example, http.//upsssc.gov.in protocol identification is http, and the resource name is upsssc.gov.in. Figure 1 shows the typical example of URL and identification.

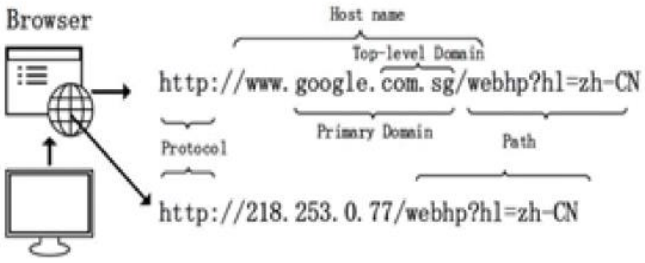

Fig.1. structure or format of URL [7]

II. LITERATURE SURVEY

Rogue websites, often known as unsafe URLs, pose a severe threat to cybersecurity. Malicious URLs host unwanted information, deceive gullible site visitors into falling for scams and result in billions of dollars in losses every year [1][2]. Such hazards must be identified and countered as soon as they appear. Blacklists have frequently been the main tool used in this detection. Blacklists, however, are incomplete and unable to identify freshly created hazardous URLs. More people have recently shown an increased interest in machine learning techniques as a means of expanding the reach of harmful URL detectors [4][5]. Using research on numerous facets of this issue, the dataset has to be categorized, defined, and provide a formal definition of the machine learning task of finding hazardous URLs.

In order to take advantage of a user's weakness, URLs have frequently been utilized and abused. The categorization of any URL as benign or harmful is the main objective of proposed implementation. It also contrasts the outcomes of machine learning classification methods such as random forest classifier, light GBM classifier, and XGboost classifier used to categorize URLs into Phishing, Benign, Defacement, and Malware[13] [14] [15].

III. PROPOSED MODELS

Collection of data-A tagged dataset of harmful, benign, defacement and malware URLs are acquired from the Kaggle repository. The dataset considered does not contain any null or empty cells.

Cleaning of data- Pre-processing includes managing missing data as well as the extraction of additional characteristics, normalization, encoding of categorical values, and standardization of values.

Model training- Using a variety of machine learning methods, such as random forest classifier, Light GBM classifier, and XGboost classifier, 80 percent of the data is utilized to train the model using the Sklearn python library.

Model validation and optimization- remaining 20% of the data used for validation. Hyperparameters are changed in order to increase the sensitivity, F1 score, recall, and accuracy.

Comparison of models- Based on assessment measures, the machine learning classification methods are contrasted.

IV. DESIGN AND IMPLEMENTATION

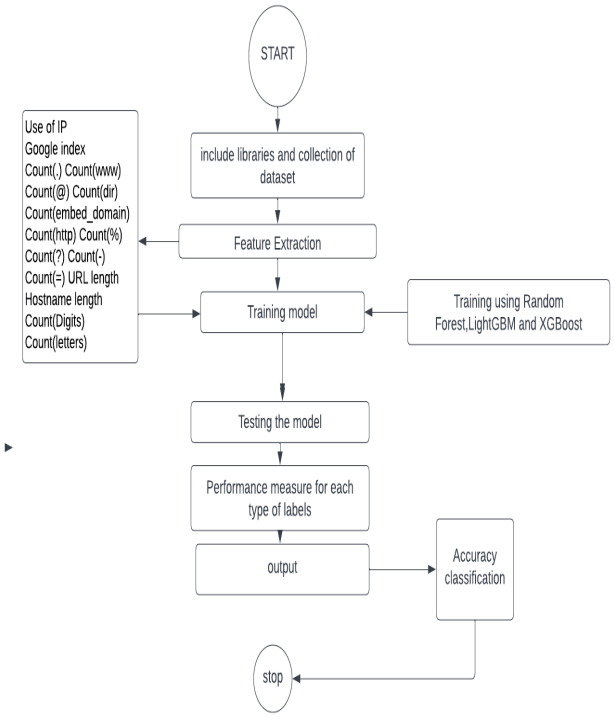

Fig.2. Flowchart of design and implementation

Figure 2 shows the flowchart of design and implementation, starts with including various libraries required and the collection of datasets. The obtained dataset extracted the various features to enhance the performance of the chosen model and also to help the model predict output easily. Feature extraction also includes the dimensionality reduction which means that obtaining the required set of columns as not all the columns of the dataset required to predict the malicious URLs [6] [12]. After getting the extracted features the model has to be trained with the mentioned classifiers. Also, it has to be tested for the dataset using 80:20 split and measured the performance using various measures and then it has compared with various models to find better model.

A. Classification Techniques

Classification is the supervised ML process in which the dataset is fed into a model using labels. The model will get prior information about its training data. Both organized and unstructured data are possible. Preprocessing, training the model, and classifying the data are all steps in the process. The classes are sometimes known as labels, goals, or categories. The classes will be for the entire dataset, but the labels are pertaining to each and every dataset. Binomial classification and multi-class classification are the two different forms of classification. The categorization of email into spam or ham emails, identifying tweets as having favorable or bad feelings, classifying various visuals like fruits, animals, and insects, among other things intricate jobs of the primary areas where classification is applied.

Random Forest Algorithm-The supervised learning method includes the well-known random forest ML algorithm. it can be utilized for both regression and classification issues. The code snippet and usage of random forest model is shown in figure 3.

```
import sklearn.metrics as metrics

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100,max_features='sqrt')
rf.fit(X_train,y_train)
y_pred_rf = rf.predict(X_test)
print(classification_report(y_test,y_pred_rf,target_names=['benign',
                                                           'defacement',
                                                           'phishing',
                                                           'malware']))

score = metrics.accuracy_score(y_test, y_pred_rf)
print("accuracy:   %0.3f" % score)
```

Fig. 3 Random Forest classifier

Light GBM classifier- is a decision tree-based gradient boosting framework that makes the model work more efficiently and uses less memory. The code snippet light GBM model is shown in figure 4.

```
lgb = LGBMClassifier(objective='multiclass',boosting_type= 'gbdt',n_jobs = 5,
          silent = True, random_state=5)
LGB_C = lgb.fit(X_train, y_train)


y_pred_lgb = LGB_C.predict(X_test)
print(classification_report(y_test,y_pred_lgb,target_names=['benign',
                                                           'defacement',
                                                           'phishing',
                                                           'malware']))

score = metrics.accuracy_score(y_test, y_pred_lgb)
print("accuracy:   %0.3f" % score)
```

Fig. 4. LightGBM Classifier

Extreme Gradient Boosting or XGboost classifier- is a concept developed by University of Washington researchers. It is a C++ library that enhances the training process for gradient boosting. The code snippet of XGboost classifier model is shown in figure 5.

```
xgb_c = xgb.XGBClassifier(n_estimators= 100)
xgb_c.fit(X_train,y_train)
y_pred_x = xgb_c.predict(X_test)
print(classification_report(y_test,y_pred_x,target_names=['benign',
                                                          'defacement',
                                                          'phishing',
                                                          'malware']))


score = metrics.accuracy_score(y_test, y_pred_x)
print("accuracy:   %0.3f" % score)
```

Fig. 5. XGboost Classifier

B. Data Visualization

Collection of Data-To train and test machine learning models tagged an open-source dataset of 651,191 websites is taken from the Kaggle repository.

```
(651191, 2)
```

	url	type
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=...	defacement
4	http://adventure-nicaragua.net/index.php?optio...	defacement

Fig. 6. Dataset specifications

Figure 6 shows the dataset specifications and type, there are two aspects in the data-URL and label.

```
benign       428103
defacement    96457
phishing      94111
malware       32520
Name: type, dtype: int64
```

Fig. 7. Analysis of data

Figure 7 shows the detailed analysis of data. The dataset contains four types of URLs namely Benign (428,103), Defacement(96,457), Phishing (94,111) and Malware (32,520). Using the Word Cloud module from python, commonly repeated words in the particular type of dataset are displayed [3].

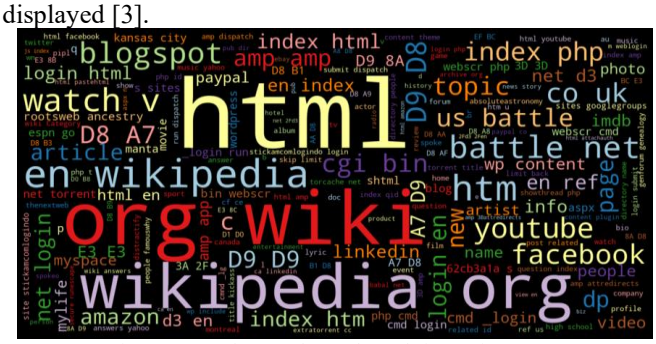

Fig. 8. Most frequently used words in Benign URLs

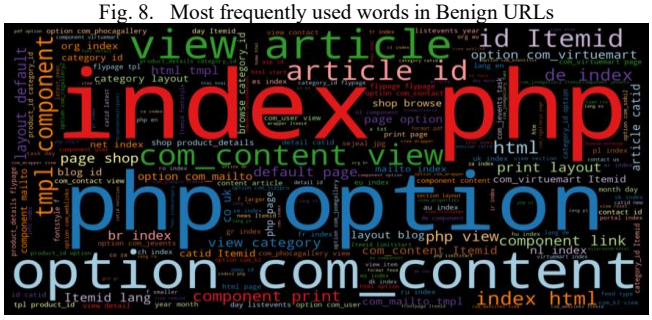

Fig. 9. Most frequently used words in Defacement URLs

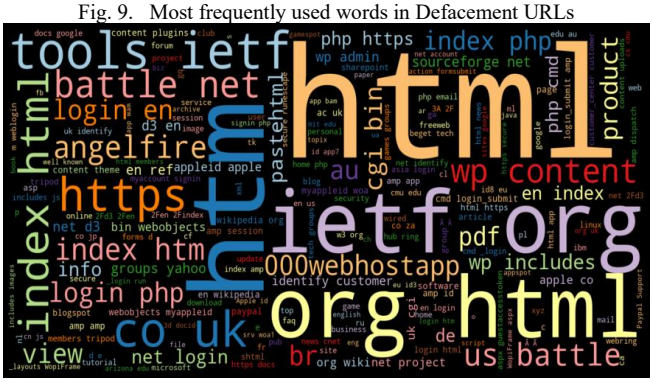

Fig. 10. Most frequently used words in Phishing URLs

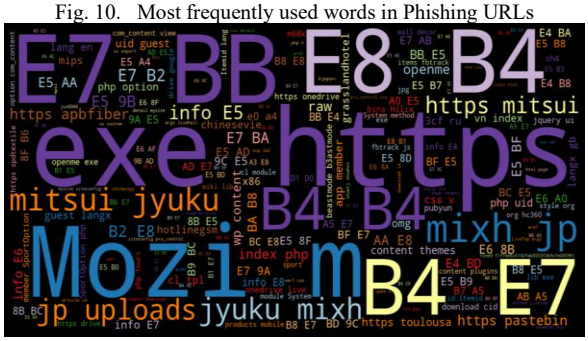

Fig. 11. Most frequently used words in Malware URLs

Figure 8 to 11 shows the most frequently used words in Benign URLs, Defacement URLs, Phishing URLs and Malware URLs respectively are displayed. It is evident from the figure that Wikipedia, .org, Youtube, Facebook etc. are the most used words in the Benign type of URLs displayed. Similarly for other types of URLs present in the dataset, the frequently used words are displayed using the WordCloud module. The WordCloud module from python is used to display the frequency or the importance of words in the given dataset. It uses the principle of thresholding, if some words reach the threshold, then those words will be displayed in the output. The bigger words are most important and smaller words will be of less importance.

Feature Extraction- The process of representing or enhancing features that improve the performance of machine learning models is known as feature extraction. it will be easy for ML models to take decisions better. It facilitates faster processing by reducing dimensionality. PCA and LDA are the two most prevalent methods. Table I shows the extracted characteristics.

TABLE I CHARACTERISTICS EXTRACTED FOR FEATURE ENGINEERING

Features	Features
Use of IP	Google Index
Count(.)	Count(www)
Count(@)	Count(dir)
Count(embed_domain)	Count(http)
Count(%)	Count(?)
Count(-)	Count(=)
URL length	Hostname length
Count(Digits)	Count(letters)

Due to the fact that a ML model is unable to directly interpret text, the categorical features, such as the use of IP, abnormal URL, and Google Index, are encoded as numbers. A count encoder is the encoding method used. The target column has benign websites set to 0 and malicious websites set to 1. The distribution of various categorical features in the dataset are displayed.

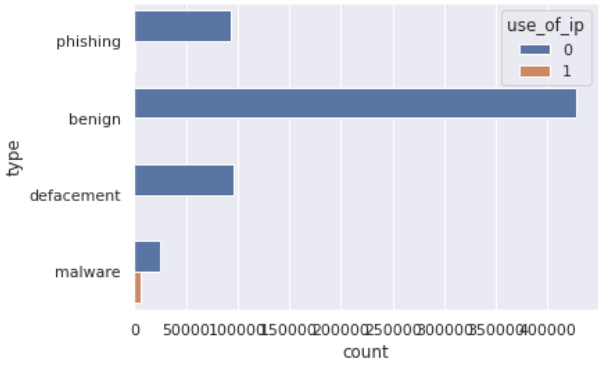

Fig. 12. Distribution of Use of IP

In figure 12, the distribution of various types of URLs with respect to the use of IP addresses are shown. It is evident that only malware type URLs use the IP addresses to receive unauthenticated information. Malware is very useful in IP spoofing because the user sees them as normal URLs and they work upon it but it goes unnoticed by the users that it has the potential to steal the information. In figure 13, the distribution of various types of URLs with respect to abnormalities are shown. It is evident that only the Benign and Phishing type of dataset have normal types of URLs.

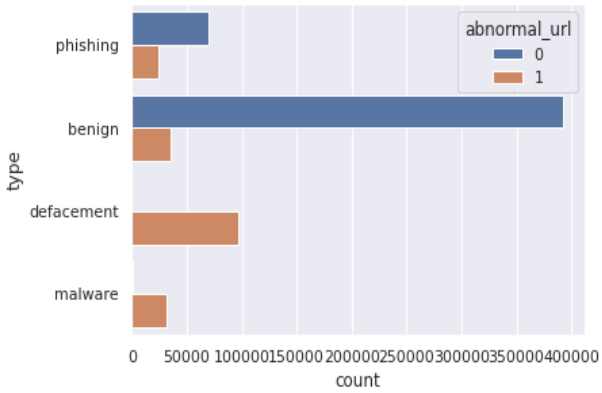

Fig. 13. Distribution of abnormal URL

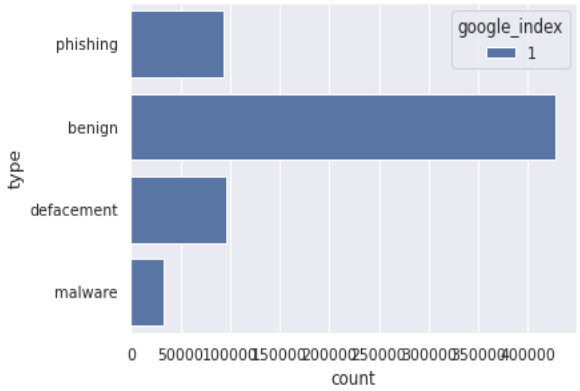

Fig. 14. Distribution of Google Index

In Figure 14,The distribution of various types of URLs with respect to google index are shown. Google index 1 specifies that the URL is not malicious, and it does not contain any fraudulent information. It is evident that Benign type of malicious are higher google index which means that they are not fraudulent.

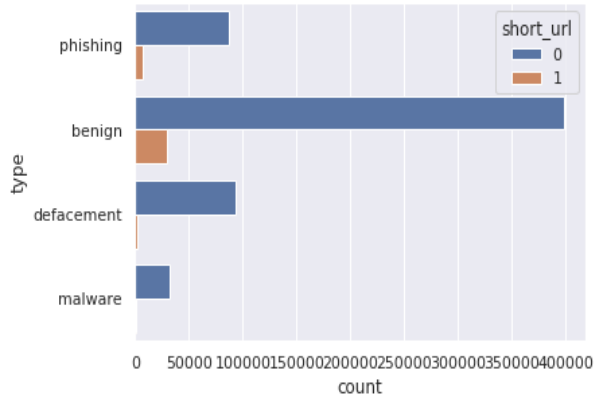

Fig. 15. Distribution of Suspicious URL

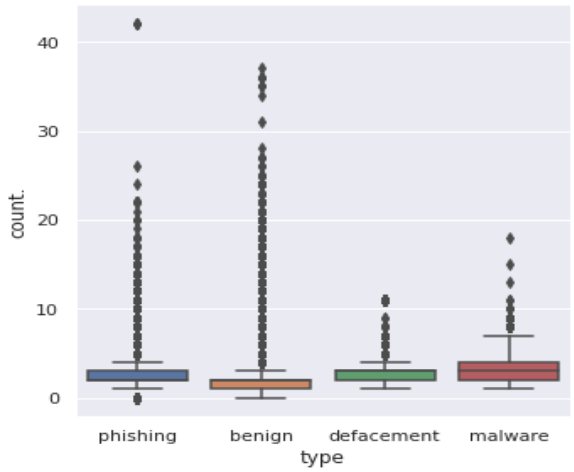

Fig. 16. Distribution of count of [.] dot

In Figure 15,The distribution of various types of URLs with respect to the length of the URLs are given. The shorter the URL is the more authoritative. It is evident that only Benign and Phishing type of dataset contain the shorter URLs which means that those URLs are not fraudulent. In figure 16, the distribution of various types of URLs with respect to the count of [.] is displayed. From the Boxplot it is evident that more dots[.] are present in the Benign and Phishing type of URLs.

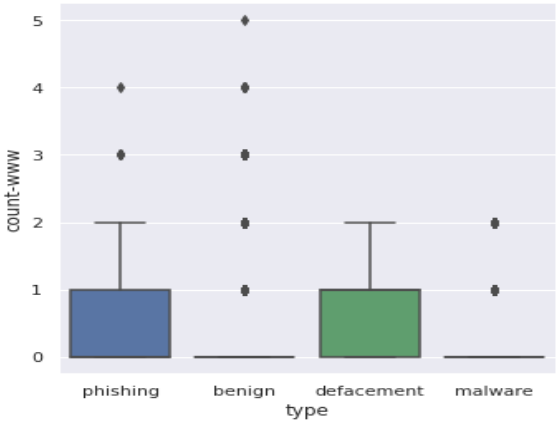

Fig. 17. Distribution of count of www

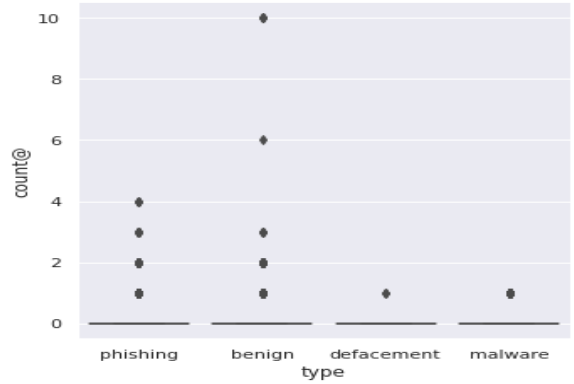

Fig. 18. Distribution of count[@]-at

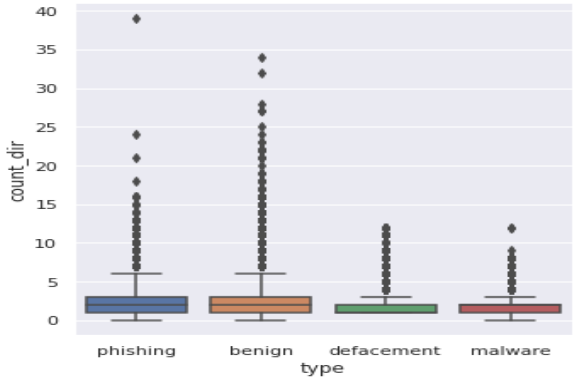

Fig. 19. Distribution of count dir

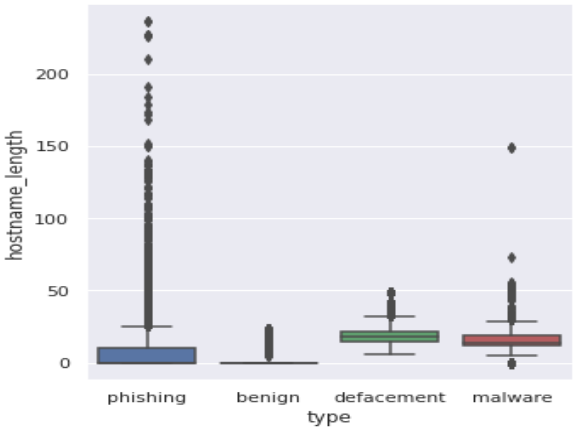

Fig. 20. Distribution of Hostname length

Figure 17 shows the distribution of various types of URLs with respect to the count of www displayed. The presence of www in the URL makes it more authentative. It is evident that Benign and Phishing type of dataset contain a greater number of www as it is more spreaded. In figure 18,the distribution of various types of URLs with respect to the count of [@] at is displayed. It represents how the URL is directed. The presence of at makes it more authentative. It is evident that Malware and Defacement type of URLs does not have at symbol which makes them more malicious. In figure 19,The distribution of various types of URLs with respect to the number of different types of URLs is displayed. In the considered dataset, there are more Benign types of dataset present. In figure 20,The distribution of various types of URLs with respect to the host name length is displayed. Benign and Phishing type of URLs have lengthier host names.

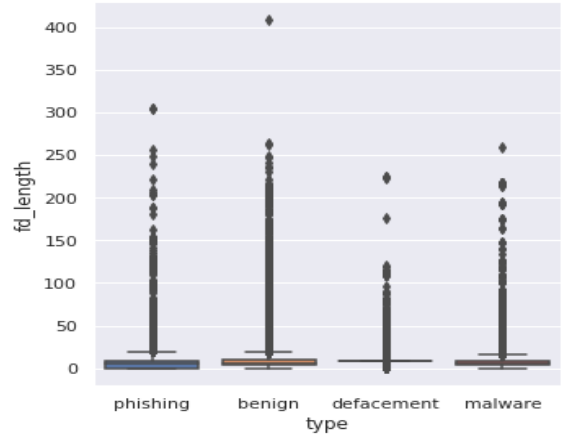

Fig. 21. Distribution of first directory length

Figure 21 shows the distribution of various types of URLs with respect to the length of the first directory displayed. Benign type of URLs has lengthier directories than other types of URLs.

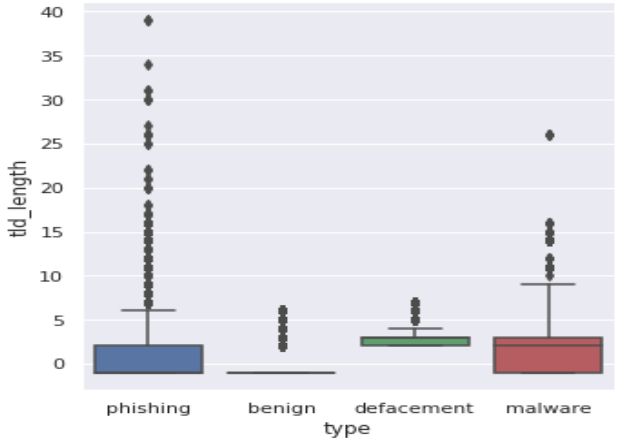

Fig. 22. Distribution of Top-level Domain length

Figure 22 shows the distribution of various types of URLs with respect to the length of the top domain displayed. Phishing type of URLs have lengthier top domain lengths compared to other types of URLs.

Data can be scaled in a fixed range using a technique called feature scaling. To manage high variance data, it is applied during the pre-processing of the data. In the absence of data scaling, machine learning models tend to place more weight on higher values than lower values. The two most widely used methods are standardization.

$$X^1 = \frac{X - \mu}{\sigma}$$

Normalization- The values are rescaled using this method between 0 and 1.Normalization is used to remove the redundant data points(i.e., outliers) and to organize the data for better performance.

$$X^1 = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Performance Metrics-The data is divided with 80:20 ratio for training and testing. The various metrics used for evaluation are

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

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

$$\text{Recall} = \frac{TP}{(TP+FP)} \quad (3)$$

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

V. Results and Analysis

A. Simulation Results

The software tool used for the simulation is Google colab. It is an open source, online platform that contains various python modules built in it and it is easy to use due to its versatile nature.

Results of random forest classifier

	precision	recall	f1-score	support
benign	0.97	0.98	0.98	85621
defacement	0.98	0.99	0.99	19292
phishing	0.99	0.94	0.96	6504
malware	0.91	0.86	0.88	18822
accuracy			0.97	130239
macro avg	0.96	0.95	0.95	130239
weighted avg	0.97	0.97	0.97	130239

accuracy: 0.966

Fig.23.Performance measures of random forest classifier

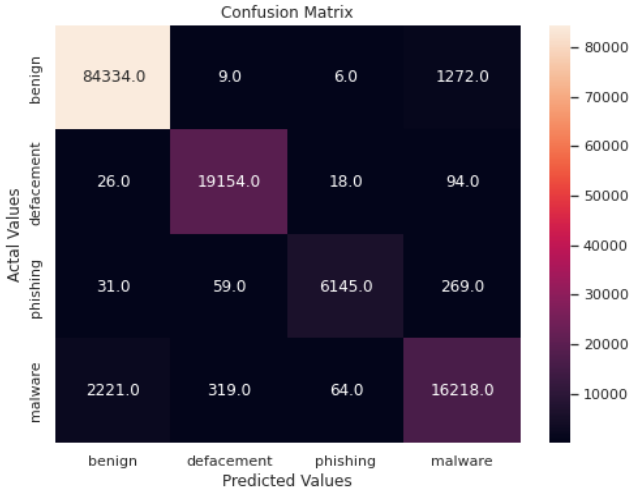

Fig. 24.Confusion matrix of random forest classifier

Figure 23 and 24 shows the performance measures and confusion matrix of random forest classifiers. Obtained results show that 84,334 Benign, 19,154 defacement, 6,145 phishing and 16,218 websites were detected correctly with the accuracy of 96.6%.

Results of LightGBM Classifier

	precision	recall	f1-score	support
benign	0.97	0.99	0.98	85621
defacement	0.96	0.99	0.97	19292
phishing	0.96	0.89	0.92	6504
malware	0.90	0.81	0.85	18822
accuracy			0.96	130239
macro avg	0.95	0.92	0.93	130239
weighted avg	0.95	0.96	0.95	130239

accuracy: 0.956

Fig.25.Performance measures of LightGBM Classifier

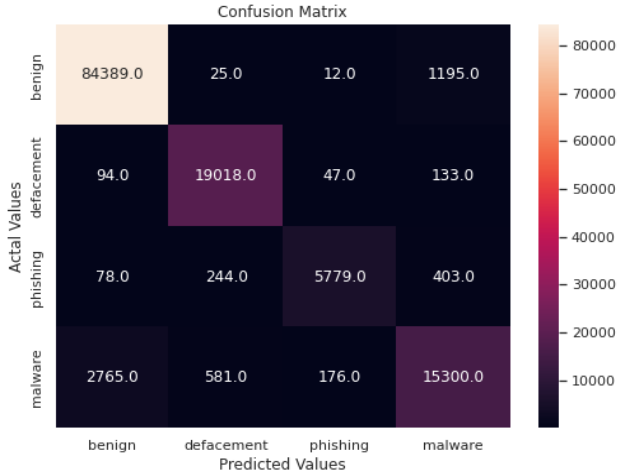

Fig. 26.Confusion matrix of LightGBM Classifier

Figure 25 and 26 shows the performance measures and confusion matrix of LightGBM classifiers. In Figure 26 it shows the confusion matrix for LightGBM classification which means that for the total number of Benign, Malware, Defacement and Phishing type of dataset the total correctly predicted URLs are displayed. Obtained results show that 84,389 Benign, 19,018 defacement, 5,779 phishing and 15,300 websites were detected correctly with the accuracy of 95.6%.

Results of XGBoost Classifier

	precision	recall	f1-score	support
benign	0.95	0.98	0.97	85621
defacement	0.89	0.96	0.92	19292
phishing	0.92	0.76	0.83	6504
malware	0.88	0.73	0.80	18822
accuracy			0.93	130239
macro avg	0.91	0.86	0.88	130239
weighted avg	0.93	0.93	0.93	130239

accuracy: 0.932

Fig.27.Performance measures of XGBoost Classifier

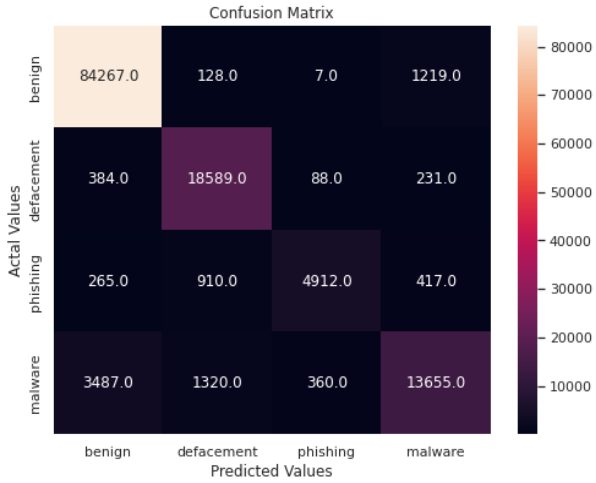

Fig. 28.Confusion matrix of XGBoost Classifier

Figure 27 and 28 shows the performance measures and confusion matrix of LightGBM classifiers. Obtained results show that 84,267 Benign, 18,589 defacement, 4,912 phishing and 13,655 websites were detected correctly with the accuracy of 93.2%.

B. *Performance Analysis*

TABLE II ANALYSIS OF PRECISION

Algorithms /Type	Benign	Defacement	Phishing	Malware
Random Forest	0.97	0.98	0.99	0.91
LightGBM	0.97	0.96	0.96	0.90
XGBoost	0.95	0.89	0.92	0.88

TABLE III ANALYSIS OF RECALL

Algorithms /Type	Benign	Defacement	Phishing	Malware

Random Forest	0.98	0.99	0.94	0.86
LightGBM	0.99	0.99	0.89	0.81
XGBoost	0.98	0.96	0.76	0.73

TABLE IV ANALYSIS OF F1 SCORE

Algorithms /Type	**Benign**	**Defacement**	**Phishing**	**Malware**
Random Forest	0.98	0.99	0.96	0.88
LightGBM	0.98	0.97	0.92	0.85
XGBoost	0.97	0.92	0.83	0.80

Table II to IV shows the analysis of precision, Recall and F1 score for random forest, LightGBM and XGBoost algorithms respectively. A random forest classifier gives the best precision comparing all the types of data. Since LightGBM classifier works on leaf-wise distribution of dataset high performance for classification can be achieved. LightGBM Classifier gives the best precision for Benign type of URLs. LightGBM classifier gave the best recall for the Benign and Defacement type of data and Random Forest Classifier gave the best recall for Defacement, Phishing and Malware type of datasets. Random forest classifier gives the best F1 score comparing all the types of the datasets. Also, LightGBM classifier gives the best F1 score for Benign type of URLs.

VI. CONCLUSION

Three ML algorithms are implemented to detect and classify malicious URLs. The random forest model gives the most accurate results. By using more balanced data to train the random forest classifier, such as data that has nearly equal numbers of both harmful and beneficial websites. The investigation demonstrates that by training a model with a database of selected attributes and using the model to anticipate new attacking attempts, it is possible to identify fraudulent URLs. Further, the proposed models can be used in search engines or websites so that it can give alert messages to the users while addressing the fraudulent URLs[8] [9] [10] [11].

REFERENCES

[1] Dhanalakshmi Ranganayakulu *et al.* "Detecting Malicious URLs in E-mail – An Implementation", *AASRI Procedia,* vol. 4, pp. 125-131, 2013.

[2] Fuqiang Yu *et al.* "Malicious URL Detection Algorithm based on BM Pattern Matching", *International Journal of Security and Its Applications,* vol. 9, pp. 33-44.

[3] K. Nirmal *et al.* "Phishing - the threat that still exists", *International Conference on Computing and Communications Technologies (ICCCT)*, pp. 139-143, 2015.

[4] F. Vanhoenshoven *et al.* "Detecting malicious URLs using machine learning techniques", *IEEE Symposium Series on Computational Intelligence (SSCI),* pp. 1-8, 2016.

[5] Doyen Sahoo *et al.* "Malicious URL Detection using Machine Learning: A Survey", *arXiv:1701.07179v3,* 2019.

[6] Rakesh Verma *et al.* "What's in a URL: Fast Feature Extraction and Malicious URL Detection", *Seventh ACM Conference on Data and Application Security and Privacy,* pp.55-63,2017.

[7] Shantanu B *et al.* "Malicious URL Detection: A Comparative Study," *International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021*, pp. 1147-1151.

[8] A. Vikram *et al.* "Anomaly detection in Network Traffic Using Unsupervised Machine learning Approach," *5th International Conference on Communication and Electronics Systems (ICCES),* 2020, pp. 476-479.

[9] R. J. Franklin et al. "Anomaly Detection in Videos for Video Surveillance Applications using Neural Networks," *International Conference on Inventive Systems and Control (ICISC),* 2020, pp. 632-637.

[10] Ritika H J *et al.* "Fraud Detection and Management for Telecommunication Systems using Artificial Intelligence (AI)," *3rd International Conference on Smart Electronics and Communication (ICOSEC),* 2022, pp. 1016-1022.

[11] Niranjan DR *et al.* "Jenkins Pipelines: A Novel Approach to Machine Learning Operations (MLOps)," *International Conference on Edge Computing and Applications (ICECAA),* 2022, pp. 1292-1297.

[12] E. Nowroozi *et al.* "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework," *IEEE Transactions on Network and Service Management,* 2022.

[13] S. Mohanty *et al.* "Predicting Phishing URL Using Filter based Univariate Feature Selection Technique," *Second International Conference on Computer Science, Engineering and Applications (ICCSEA),* 2022, pp. 1-5.

[14] M. F. A. Razak *et al.* "Comparative Analysis of Machine Learning Classifiers for Phishing Detection," *6th International Conference on Informatics and Computational Sciences (ICICoS),* 2022, pp. 84-88.

[15] M. Atari *et al.* "A Machine-Learning Based Approach for Detecting Phishing URLs," *International Conference on Intelligent Data Science Technologies and Applications (IDSTA),* 2022, pp. 82-88.

Publication Link:

https://ieeexplore.ieee.org/document/10053422

Cite This:

U. S. D. R, A. Patil and Mohana, **"Malicious URL Detection and Classification Analysis using Machine Learning Models,"** 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 470-476, DOI: 10.1109/IDCIoT56793.2023.10053422.