Identifying Suspicious URLs using Supervised Learning and Lexical Analysis

Debanjan Mitra

School of Engineering,
University of Guelph
Guelph, Canada
E-mail: mitrad@uoguelph.ca

Ruthvik Raja M.V

School of Engineering,
University of Guelph
Guelph, Canada
E-mail: rmunirra@uoguelph.ca

***Abstract*- The internet has evolved into a platform for a wide range of illicit acts, from spam advertising to financial fraud, thanks to technological advancements. Malware programmes embedded in URLs are used to carry out some of these actions. Malicious websites are used by criminals and hackers to carry out illegal actions such as implanting malware, getting unauthorised access to networks, and illegally gathering personal information. They post superfluous contents (inappropriate ads, malware, spam, spoofing, defacement, and so on) and lure unsuspecting users into becoming scam victims (malware installation, private information disclosure, financial loss, fake shopping, extortion, site, unexpected prize, and so on), resulting in billions of dollars in losses each year. In this project, we look at the detection of malicious URLs as a binary and multi-class classification problem, and we compare the performance of many well-known supervised machine learning classifiers such as K-Nearest Neighbors, Support Vector Machines and Random Rain Forest. We show that lexical analysis can be used to detect these URLs in a proactive manner. To train the model, we used a publicly available dataset from the University of New Brunswick, which had 58972 URLs and 80 distinct Lexical characteristics. The best model is then chosen as the one with the highest accuracy score and the shortest compilation time.**

***Keywords* - Supervised Learning, URL, Malicious, Lexical Analysis, Classification, MinMax Scaler, Principal Component Analysis, Hyper Parameter Tuning, Machine Learning.**

Abbreviations

KNN - K Nearest Neighbors
PCA - Principal Component Analysis
SVM - Support Vector Machines
URL - Uniform Resource Locator
Tld - Top Level Domain

I. INTRODUCTION

The broad acceptance of the network communication technologies and the vast development of the web have resulted in an increasingly widespread range of online banking, e-commerce, online social networking, and e-government as well as other business processing, online social services, and consulting purchases. But the Internet is a sword with two sides. In same time, network security issues are also convenient. The Internet, driven by interests, has gradually developed into an active criminal platform. Several e-mails, phishing websites, scam messages and spam seriously interfere in the lives of people, resulting in serious financial losses and information leaks. The URL shows the resource location and the internet access. Malicious URLs were developed quickly and with complex and diverse changes as a prelude to cyber-attacks. These changes have transformed into a broader problem in network security. Attackers create network attack operations using malicious URLs and incentivize users to click in order to do phishing, virus distribution, and malicious script execution. Due to the high frequency of ransomware, phishing, fraud, and major data leaks, ongoing research on an effective malicious URLs detection system is critical

for the network's continued health. The ability to identify and detect fraudulent URLs has become a trending topic in academia.

Malicious URL detection approaches are currently separated into 3 categories: URL based on feature, based on blacklist and feature-based web page methods. However, each of these methods has its own set of benefits and drawbacks, and the effectiveness of malicious URL detection is still lacking.

1.1 Method- URL features

One of the popular research topics of harmful URLs is the detection approach based on malicious URL properties, which is frequently applied by deep learning and machine learning algorithms. This process is frequently based on the URL string's features, such as upper-case letters, URL length, domain name information, special characters and so on. This type of detection method's primary steps is to first choose and extract URL features from a vast number of labelled samples, then the classification model is trained using the features extracted, and then utilise the trained classification model to predict unknown URLs [13]. The algorithm for ML uses manual rules for extraction of features so that the function extraction is incomplete or not good. In addition, malicious URL features are quick to change and more complex and varied. It is difficult to adapt manual function extraction to quick updating malicious URLs. The precision of the marking data and the extracted features will significantly impact on the model's accuracy and effectiveness. The deep learning algorithm, on the other hand, has a better automated learning performance; it can automatically take sample characteristics, reducing workforce consumption and manual removal defects significantly. However, because it is still based solely on text characteristics of malicious URLs in the training set, the model's adaptability and ability to be updated automatically is questioned. Furthermore, this detection method does not result in the attack of malicious URLs, and there will be considerable detection and errors.

1.2 Method- Blacklist

The most classic and traditional method is blacklist based malicious URL detection. This is an effective and simple detection method with less consumption of resources. It is a malicious filtering URL method that many web applications and browsers use commonly. The blacklist is user reporting, manual marking, webcasting, honeypot technology and other types used to generate and maintain security monitoring institutions. The main concept of the blacklist method of detection is to match the URL of the given blacklist. If the URL has been hit, then the URL is malicious, as is normal otherwise. The greatest problem with this process is that only existing, detected malicious URLs can be identified and malicious URLs can’t be judged. It often has low detection and falsified alarm problems. As the number of URLs globally grows rapidly, the slowly updating blacklist can’t hit many new malicious URLs and cannot achieve satisfactory results with this detection method. Moreover, the heuristic blacklist algorithm can only identify malicious URLs that are existing or partially unknown under existing heuristic regulations.

1.3 Method – Webpage features

Because phishing web pages are frequently designed to deceive, mislead, and attack users, web pages accessed through malicious URLs typically have many behaviours and special tags that distinguish them from web pages that are normal, making the malicious URL detection method based on web page features a very effective method [13]. Most of these methods now use tags and text content in the source code of online sites to detect malicious URLs, or they use assaults on web pages to identify malicious URLs. Nevertheless, many Phishing web sites use different means to alter, encrypt and hide sensitive text and attack behaviours, so this algorithm usually does not provide efficient text content from the source code of web pages, and attack behaviour requirements may be too high that

cannot identify the attack modes and purposes of web pages, and there will be huge attacking web pages requirements [13]. Furthermore, triggering the website's attacking behaviour increases detection time and the consumption of resources which isn't very effective.

A specific URL can be categorised into various types, but this project aims at 5 types of namely Spam URLs, Phishing URLs, Malware URLs, Defacement URLs and Benign URLs [12]. The Benign URLs are the trusted websites whereas all other mentioned URLs are malicious URLs [10]. Generally classifying an URL will be limited to whether it is a spam or benign URL [3,4] but there are different types of URLs as mentioned above based on various features. The URL identification is a multi-classification problem, and this can be done by using various algorithms that are present in Machine Learning. To train the model, we used a publicly available dataset from the University of New Brunswick, which had 58972 URLs and 80 distinct Lexical characteristics. The best model is then chosen as the one with the highest accuracy score and the shortest compilation time.

II. LITERATURE REVIEW

Malicious URLs have traditionally been found using blacklisting services and plugin and plugin or APIs such as [1] and [2] using statistical approaches such as TF-IDF, URL weighing and manual collections of malicious URLs. The Blacklisting Services approach and analysis is presented in [3]. The heuristic approach to detected URLs became more popular as machine learning came to be. One of the first works in this respect was [4] where lexical URL characteristics are used compared with URL packet functionality such as TTL, Country of Origin and Server - used in this paper. Over time, more URL classification features are being explored. Phishing websites can be identified using their URLs as described in [5], but this paper concentrates on five types of suspicious URLs namely spam, phishing, malware, defacement and benign. A detailed explanation is provided in [6] of the various features that can be used to efficiently detect URLs. The extracted features are subjected to different types of machine learning algorithms.

Cui et al. [7] suggested an approach to detection using machine learning techniques for automatically classifying URLs as a malicious or benign website. In their analysis they used statistical algorithms, by selecting effective features with the combination of progressive learning and feature extraction through sigmoidal threshold level. They achieved an accuracy of 98.7% through this process.

A hybrid approach has been proposed by Altaher [8] to detect the Site as Phishing, Legitimate or Suspect Websites. The approach is the result of combining the Support Vector Machine (SVM) algorithms and the K Nearest-Neighbors algorithm (KNN) in two phases with two machine learning algorithms. The experiment with this approach proposed a 90.04 % accuracy.

Sirageldin et al. [9] used algorithms of machine learning to detect malicious websites according to two lexical and page content feature groups. The combination of the characteristics produced the best result; false positive rates were reduced, and the Artificial Neural Network (ANN) achieved the best performance with 96% accuracy.

The datasets used by Yahoo-Phish tank [10] have taken into account both lexical and host features. The performance of the various classifiers is evaluated using these features [11] thereby offering the use of RIPPER technique [12] and creating an admin user interface that gives an input URL, receiving output as a fraud or a true user interface. Where the URL is fraudulent, the user is notified. The URL classification approach in this paper closely relates to [13] where a broad range of machine learning (for URL detection) algorithms are used, such as SVM, KNN, Random Forest, Decision Tree or Naïve Bayes. Random forest is found to be extremely precise. This paper contains similar algorithms, but a different dataset that consists of newer URLs and features is used.

The arrival of the Big-Data era gradually led us to detect methods based on profound knowledge. Common algorithms include convolutional neural network (CNN) and recurrent neural network (RNN). Le et al. [13] [14] proposed URLNet, which includes char-CNN and word-CNN respectively. The

URL dataset was represented first in parallel as character level and word-level vectors, then incorporated into an automatic feature extraction and training CNNs. After training of the model, test data would be placed in the model to yield the prediction results. Unlike [14], in data pre-processing Luo et al. [15] used an autoencoder to represent the characteristics in the form of a JEG image to make it more obvious to the human eyes. In the meantime, Peng et al. [16] have adopted a mechanism for the collection of URL information and developed a tool for it. They also produced URL prediction results based on LSTM and CNN model combinations. Deep learning methods have shown a good result in both efficiency and accuracy of malicious URL detection.

With the growing number of websites, it's becoming more important to add newer URLs in the dataset and use the most up-to-date algorithms. When the dataset was imbalanced, there was no consideration for classifying the website. This paper focuses on the issues that have previously gone unnoticed. The use of feature ranking approaches, as well as the correction of dataset imbalance, has been investigated. Techniques for choosing elements that contribute considerably to the URL being malicious are also discussed.

III. BACKGROUND

3.1 Lexical Features Analysis

Lexical features can be considered as the textual properties of an URL like QueryLength, DomainLength, URL LetterCount, Length of the HostName etc. Lexical Features are lightweight in nature so, it takes less time for computation and due to its lightweight property, it is popular in the field of Machine Learning [15]. The Lexical features are extracted from an URL and it does not depend on any specific application like email, social networking websites etc. Since most of the Malicious or Spam URLs have a short life span, the features that are extracted will remain present and can be utilised to detect new incoming Malicious URLs even when the old Malicious URLs are unavailable [16].

In this paper, the following 79 Input parameters and 1 Output parameter are considered for analysis [Table 1]:-

Table 1: 79 Lexical Features and 1 Output parameter

Lexical Features	Lexical Features
Query length	Directory DigitCount
Domain token count	File name DigitCount
Path token count	Extension DigitCount
Avgdomaintokenlen	Query DigitCount
Longdomaintokenlen	URL LetterCount
Avgpathtokenlen	Host LetterCount
Tld	Directory LetterCount
Charcompvowels	Filename LetterCount
Charcompace	Extension LetterCount
Ldl url	Query LetterCount
Ldl domain	LongestPathTokenLength
Ldl path	Domain LongestWordLength
Ldl filename	Path LongestWordLength
Ldl getArg	Sub Directory LongestWordLength
Dld url	Arguments LongestWordLength
Dld domain	URL sensitiveWord
Dld path	URLQueries variable
Dld filename	SpcharUrl
Dld getArg	Delimeter Domain
UrlLen	Delimeter path
Domainlength	Delimeter Count

Lexical Features	Lexical Features
PathLength	NumberRate URL
SubDirLen	NumberRate Domain
FileNameLen	NumberRate DirectoryName
This.fileExtLen	NumberRate FileName
ArgLen	NumberRate Extension
PathurlRatio	NumberRate AfterPath
ArgUrlRatio	SymbolCount URL
ArgDomanRatio	SymbolCount Domain
DomainUrlRatio	SymbolCount Directoryname
PathDomainRatio	SymbolCount FileName
ArgPathRatio	SymbolCount Extension
Executable	SymbolCount Afterpath
IsPortEighty	Entropy URL
NumberofDotsinURL	Entropy Domain
ISIpAddressInDomain Name	Entropy DirectoryName
CharacterContinuityR ate	Entropy Filename
LongestVariableValue	Entropy Extension
URL DigitCount	Entropy Afterpath
Host DigitCount	URL Type (Output)

Following is the description about some of the features that are used for the analysis: -

Entropy domain and extension: Most of the time Malicious websites inserts characters in the URL to make it look like the original URL. For example, CITI can be represented as C1TI, ICICI can be represented as IC1CI etc, by replacing the letter I with the number 1. English has low entropy, and the value gets affected when we try to insert characters thereby, identifying the malicious URLs Entropy value can be used for prediction.

Character continuity rate: This feature is used to calculate the length of the longest continuous token length of a particular type in an URL. Usually, malicious websites have variable lengths of different character types, and the Character continuity rate can be calculated by dividing the sum of the longest token length of a character type by the length of the URL [17].

Length Ratio: The following features are used to find any abnormal parts [17] in an URL: -

Argument Path ratio: The ratio of Argument and path in an URL.

Argument URL ratio: The ratio of Argument and URL.

Argument Domain ratio: The ratio of Argument and Domain in an URL.

Domain URL ratio: The ratio of Domain and URL.

Path URL ratio: The ratio of Path and URL.

Path Domain ratio: The ratio of Path and Domain in an URL.

Frequency of Alphabets, Tokens and Symbols: The following features are used to calculate the frequency of different character types in an URL [16,18,19]: -

Symbol count Domain: This feature is used to calculate the number of delimiters present in a domain of an URL (Ex: /?=.,;+] etc). Phishing URLs consists of so many dots when we compare with Benign URLs [20,21].

Domain token count: This feature is used to calculate the number of tokens present in a domain and usually Malicious URL's consists of multiple domain tokens.

Query digit count: This feature is used to calculate the number of digits present in the query part of an URL.

tld: A few URLs which are considered as Phishing URLs uses multiple top level domain names within a domain name.

3.2. About the Dataset

The dataset [23] consists of more than 60,000 URLs and 79 different Lexical features like the length of the hostname, URL length, tokens that are found in the URL etc which are initially collected containing benign and malicious URLs in four different categories as Spam [24], Malware [25], Phishing [26] and Defacement. The output variable in the datasets is the URL type (Spam, Benign, Malware, Phishing, Defacement). There are four different files in the dataset and each file consists of benign and a single class among malicious URLs.

Initially, pre-processing of data is done to remove null values, noisy data and strip white spaces from the column names. In the previous papers [22], the authors have dropped all the rows with at least 1 Null value but in this paper, we have examined the four datasets individually to identify the columns that are having high Null values. The Malware dataset consists of nearly 40% of Null values in the Number Rate Extension column, Spam dataset consists of nearly 34% Null values in the Number

Rate Extension column, Phishing dataset consists of nearly 48% Null values in the Number Rate Extension column and Defacement dataset consists of nearly 39% and 32% Null values in the Entropy Directory Name and Number Rate Extension columns. Thereby we have dropped the columns that have high Null values because to prevent the rows from getting dropped because of 1 or 2 Null values among 79 features. After dropping all the Null values MinMax scaler was implemented on all the datasets to normalise all the features because each feature in the dataset is of a different scale and it is very important to scale each feature before it is sent to the model for training. Then, binary classification is implemented on each category to test the algorithm and to extract the important features that affect the output variable. Finally, all the files are combined, shuffled and applied PCA on the features to do the multi-classification.

The size of each dataset before and after cleaning is mentioned in the Table 2:-

Table 2: Size of each Dataset

Dataset Name	Before Cleaning		After Cleaning	
Rows & Features	No of Rows	No of Input Features	No of Rows	No of Input Features
Malware Dataset	14493	79	12442	78
Spam Dataset	14479	79	12420	78
Phishing Dataset	15367	79	13084	78
Defacement Dataset	15711	79	15477	77

IV. PROPOSED METHODS

The classification of an URL can be done using various methodologies that are available in Machine Learning like K-Nearest Neighbours, Logistic Regression, Support Vector Machines, Random Forests etc [27] and finally the model with high accuracy score and low compilation time can be chosen as the best model. If the accuracy score is not satisfactory by the Machine Learning algorithms, then Deep Learning can be implemented [Figure 1] and also to extract the best features from the dataset Correlation evaluation has been implemented. To measure the performance of the model various performance metrics like Accuracy score, Confusion-matrix, Sensitivity etc can be applied to the algorithm.

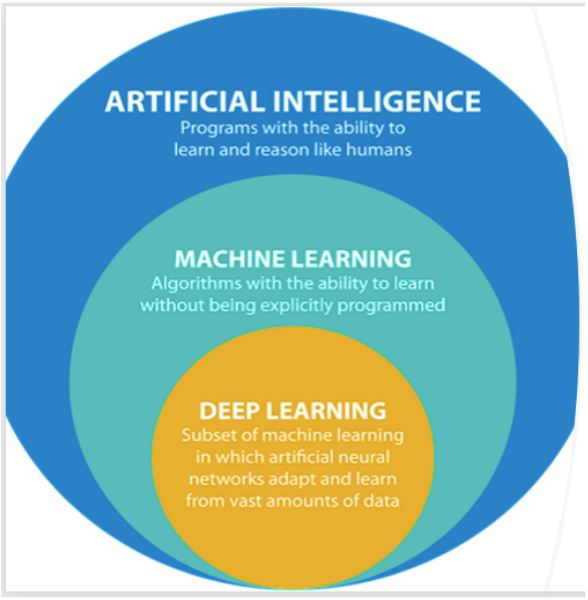

Fig 1: Machine Learning (vs) Deep Learning

The proposed methods mainly consist of four steps:- Loading all the four datasets and pre-processing them for feature extraction, Partitioning the data and applying standardisation techniques like MinMax scaler on the datasets, Applying a Supervised Classification algorithm on the training set and the final step is to evaluate the model (or) algorithm on the test data [Figure 2]. Classification of an URL whether it comes under Spam (or) Malware (or) Defacement (or) Phishing (or) Benign [Original or safe URL] URL can be done using various Classification Methodologies that are available in Machine Learning. In this paper, the KNN algorithm, Random Forest Classifier and SVM was implemented on all the datasets for Binary classification and also for multi-classification. In statistics, the K Nearest Neighbors algorithm is widely used on classification problems which is a non-parametric method. In KNN Classification, a new data point is assigned to a particular class based on the plurality vote of its neighbors (K - Neighbors), a data point is assigned to the class most common among its K Nearest Neighbors. KNN has outperformed in both binary and multi classification. In binary classification the model has to classify whether an URL is a Benign (or) Malicious URL whereas in multi-classification the model (or) algorithm has to classify whether an URL is a Benign (or) Defacement (or) Spam (or) Malware (or) Phishing URL.

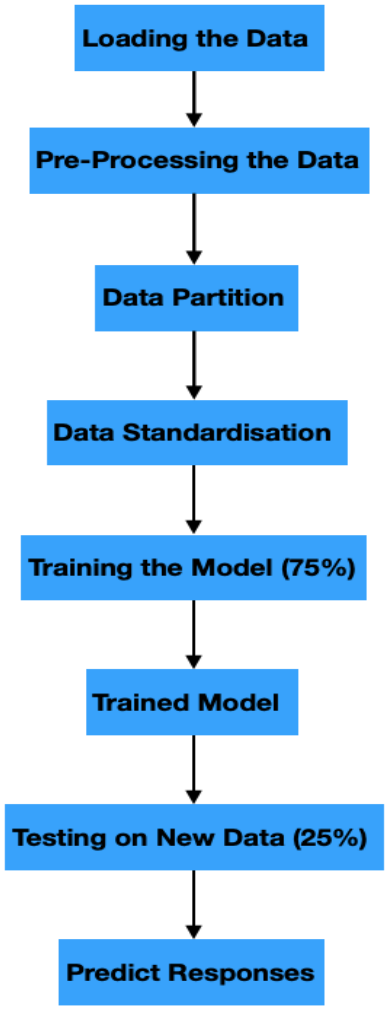

Figure 2: Workflow of a Supervised Learning Algorithm

V. RESULTS

Initially, using all the features as mentioned in Table 2 are applied on the KNN algorithm for performing binary classification and the Accuracy Score, F1-Score metrics are used to evaluate the performance of a model also the time taken by the model to train and predict are mentioned in Table 3, 4 and 5 by the algorithms KNN, Random Forest and SVM. While implementing Random Forest initially we have used nearly 100 trees to train the model but further, we reduced it to 20 because of time complexity, still, it performed well with a fewer number of trees.

From Table 3,4 and 5 the algorithms have performed well even after reducing the number of features using Correlation analysis and PCA thereby it is good to use datasets with a fewer number of features that captures the variability of all the data points because this reduces the time complexity which plays a major role in real-life scenarios. After performing the Binary classification, there are so many features that show no impact on the output variable. So, after dropping the features which are considered useless for training the algorithm, only 27 features has shown some impact on the output variable. Finally, PCA is applied on the 27 features to capture the variance and to reduce the time complexity in-terms of training and predicting the values using an algorithm. The results of a Multi-classification using different algorithms are shown in Table 6, 7 and 8.

From Table 6, 7 and 8 after dropping so many features and performing PCA also the model has achieved high accuracy score. Also, there is a huge difference between the time taken by the KNN algorithm to train and predict the responses. Hence, after dropping unnecessary features and applying PCA, we have reduced the time complexity by 24 times for KNN model. Random Forest took like the same time even after dropping so many unnecessary features, also it performed well. Out of the three, the SVM model took the maximum time and also the accuracy was less as compared to other models. So, it is not as useful in this project.

Table 3: Results of Binary Classification (KNN)

Binary Classification						
Dataset	Before applying PCA			After applying PCA		
	No of Features	Accuracy Score (Train, Test) F1-Score(Test)	Time (sec)	No of Features	Accuracy Score (Train, Test) F1-Score(Test)	Time (sec)
Malware	78	99%, 98.4% 0.98	1.2 s	10	98.5%, 97.7% 0.98	0.18 s
Spam	78	99.5%, 99.4% 0.99	1.15 s	6	99.3%, 99.1% 0.99	0.14 s
Phishing	78	97.7%, 96.8% 0.96	1.15 s	10	96.6%, 96.4% 0.96	0.23 s
Defacement	77	99.5%, 99.1% 0.99	1.33 s	10	99.5%, 99% 0.99	0.24 s

Table 4: Results of Binary Classification (Random Forest)

Binary Classification						
Dataset	Before applying PCA			After applying PCA		
	No of Features	Accuracy Score (Train, Test) F1-Score(Test)	Time (sec)	No of Features	Accuracy Score (Train, Test) F1-Score(Test)	Time (sec)
Malware	78	100%, 99.6% 0.99	0.92 s	10	99.9%, 98.9% 0.99	0.2 s
Spam	78	100%, 99.8% 0.99	0.71 s	6	99.9%, 99.4% 0.99	0.17 s
Phishing	78	100%, 98.4% 0.98	1 s	10	99.9%, 97% 0.96	0.24 s
Defacement	77	100%, 99.7% 0.99	0.93 s	10	99.9%, 99.5% 0.99	0.24 s

Table 5: Results of Binary Classification (SVM)

Binary Classification						
Dataset	Before applying PCA			After applying PCA		
	No of Features	Accuracy Score (Train, Test) F1-Score(Test)	Time (sec)	No of Features	Accuracy Score (Train, Test) F1-Score(Test)	Time (sec)
Malware	78	93.68%, 93.54% 0.94	3.30	10	90.31%, 89.77 % 0.90	0.98
Spam	78	99.12%, 98.94% 0.99	0.57	6	98.66%,98.78% 0.99	0.13
Phishing	78	95.91%, 95.30% 0.96	2.37	10	95.77%, 96.40 0.96	0.52
Defacement	77	97.92%,97.65% 0.98	2	10	96.63%, 96.61% 0.97	0.63

Table 6: Results of Multi Classification (KNN)

Multi Classification						
Dataset	Before applying PCA			After dropping features and applying PCA		
	No of Features	Accuracy Score (Train, Test)	Time (sec)	No of Features	Accuracy Score (Train, Test)	Time (sec)
Combined Dataset (58972 rows)	77	97%, 95.9%	22.5 s	10	97%, 96%	0.93 s

Table 7: Results of Multi Classification (Random Forest)

Multi Classification						
Dataset	Before applying PCA			After dropping features and applying PCA		
	No of Features	Accuracy Score (Train, Test)	Time (sec)	No of Features	Accuracy Score (Train, Test)	Time (sec)
Combined Dataset (58972 rows)	77	99.9%, 98.4%	1.3 s	10	99.9%, 97.2%	1.2 s

Table 8: Results of Multi Classification (SVM)

Multi Classification						
Dataset	Before applying PCA			After dropping features and applying PCA		
	No of Features	Accuracy Score (Train, Test)	Time (sec)	No of Features	Accuracy Score (Train, Test)	Time (sec)
Combined Dataset (58972 rows)	77	88.58, 88.659	111.92	10	89.45, 89.45	22.68

VI. DISCUSSION

As mentioned in the Section About the Dataset, after performing the pre-processing step, Firstly the URLs with class Benign is labelled as 1 whereas, the URLs with class spam (or) malware (or) defacement (or) phishing is labelled as 0 in each four different data frames. Then, all the features are considered for doing the Binary classification and after performing the mentioned steps, hyperparameter tuning is performed to derive the best parameters that yield high accuracy scores. Secondly, the correlation between the input features and the output parameter is calculated and the features with correlation values almost equal to zero are dropped from the data frame. Also, the correlation between the input features is calculated and one of the features with a correlation value greater than 0.8 are dropped from the final dataset because to avoid a multicollinearity conundrum.

Finally, deep analysis is performed on all four datasets to drop the unnecessary features from the datasets. Nearly, 46 features have shown no impact on the output variable so, we have dropped all the 46 features from the final dataset (Malware + Phishing + Spam + Defacement). At last, PCA is implemented on the final dataset to further reduce the number of features. The following 46 features are considered as unnecessary features from the analysis:- ‘Querylength', ‘Entropy_DirectoryName’, 'path_token_count', 'avgdomaintokenlen', 'longdomaintokenlen', 'avgpathtokenlen', 'charcompvowels', 'charcompace', 'ldl_url', 'ldl_path', 'ldl_filename', 'ldl_getArg', 'dld_url', 'dld_domain', 'dld_path', 'dld_filename', 'urlLen', 'domainlength', 'pathLength', 'subDirLen', 'this.fileExtLen', 'ArgLen', 'pathurlRatio', 'ArgUrlRatio', ‘argDomanRatio', 'argPathRatio', 'executable', 'isPortEighty', 'ISIpAddressInDomainName', 'LongestVariableValue', 'URL_DigitCount', 'host_DigitCount', ‘Directory_DigitCount', 'Extension_DigitCount', 'Query_LetterCount', 'Path_LongestWordLength', ‘URL_sensitiveWord', 'URLQueries_variable', 'spcharUrl', 'delimeter_Count', 'NumberRate_Domain', 'NumberRate_DirectoryName', 'NumberRate_AfterPath', 'SymbolCount_URL', 'SymbolCount_FileName', ‘SymbolCount_Extension'.

VII. CONCLUSION

In this study Machine Learning is used to make predictions on the Malicious URLs Datasets (Spam, Benign, Defacement and Malware) and Benign URLs Dataset. This technique is an addition to the blacklist techniques, in which new Malicious URLs can be easily captured accurately. Random Forest classifier has outperformed in classifying the Malicious and Benign URLs with an average Accuracy Score of 99%, F1-Score of 0.99 in Binary Classification and with an average Accuracy Score of 98.8% in Multi-Classification. Despite, SVM didn’t perform well also, it took high compilation time compared to KNN and Random Forest.

References

[1] Google Safe Browsing API - Google Code, http://code.google.com/apis/safebrowsing/, accessed on June 12, 2010.

[2] SmartScreen Filter – Microsoft Windows http://windows.microsoft.com/enUS/internetexplorer/products/ie- 9/features/smartscreen-filter, 2011.

[3] Luong Anh Tuan Nguyen, Ba Lam To, Huu Khuong Nguyen, Minh Hoang Nguyen. Detecting phishing web sites: A heuristic URL-based approach. 2013 International Conference on Advanced Technologies for Communications (ATC 2013)

[4] Steve Sheng, Brad Wardman, Gary Warner, Lorrie Faith Cranor, Jason Hong, Chengshan Zhang. An Empirical Analysis of Phishing Blacklists. CEAS 2009 - Sixth Conference on Email and Anti-Spam July 16-17, 2009, Mountain View, California USA

[5] Min-Yen Kan and Hoang Oanh Nguyen Thi. Fast Webpage Classification Using URL Features. CIKM'05, October 31-November 5, 2005, Bremen, Germany. ACM 1-59593-140-6/05/0010.

[6] Jin-Lee Lee, Dong-Hyun Kim, Chang-Hoon, Lee. Heuristic-based Approach for Phishing Site Detection Using URL Features. Proc. Of the Third Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2015

[7] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. Department of Computer Science and Engineering University of California, San Diego

[8] B. Cui, S. He, X. Yao, and P. Shi, "Malicious URL detection with feature extraction based on machine learning," Int. J. High Perform. Comput. Netw., vol. 12, no. 2, p. 166, 2018.

[9] A. Altaher, "Phishing Websites Classification using Hybrid SVM and KNN Approach," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, pp. 90- 94, 2017.

[10] A. Sirageldin, B. Baharudin, and L. Jung, "Malicious web page detection: A machine learning approach," in Advances in Computer Science and its Applications, Springer, Berlin, Heidelberg, pp. 217224, 2014.

[11] Neha Sangal Urvashi Prajapati and Deepti Patole. Fraud Website Detection using Data Mining.International Journal of Computer Applications 141(3):40-44, May 2016

[12] RIPPER William Cohen, Fast Effective Rule Induction, Proceedings of the 12th International Conference on Machine Learning

[13] Yu Chen, Yajian Zhou, Qingqing Dong, Qi Li. "A Malicious URL Detection Method Based on CNN", 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), 2020

[14] Shantanu, B Janet, R Joshua Arul Kumar. "Malicious URL Detection: A Comparative Study", 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021

15) Le, A., Markopoulou, A., Faloutsos, M.: PhishDef: URL names say it all. In: Proceedings IEEE, INFOCOM. IEEE,(2011)

16) Chu, W., et al.: Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. In: IEEE International Conference on Communications (ICC), (2013)

17) Lin, M.-S., et al.: Malicious URL filtering- a big data application. IEEE International Conference on Big Data,(2013)

18) Thomas, K., et al.: Design and evaluation of a real-time URL spam filtering service.
In: Proceeding of the IEEE Symposium on Security and Privacy,(SP),(2011)

19) Abdelhamid, N., Aladdin, A., Thabtah, F.: Phishing detection based associative classification data mining. Expert Syst. Appl. 41(3), 5948–5959 (2014)

20) Ma, J., et al.: Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery andDataMining.ACM,(2009)

21) Xiang, G., et al.: CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. (TISSEC) 14(2),21,(2011)

22) Mohammad Saiful Islam Mamun, Mohammad Ahmad Rathore, Arash Habibi Lashkari, Natalia Stakhanova and Ali A. Ghorbani, "Detecting Malicious URLs Using Lexical Analysis", Network and System Security, Springer International Publishing, P467-482, 2016

23) Available online at https://www.unb.ca/cic/datasets/url-2016.html.

24) WEBSPAM-UK dataset. http://chato.cl/webspam/datasets/uk2007/.

25) Malware domain dataset. http://www.malwaredomains.com/.
26) OpenPhish dataset. https://openphish.com/.

27) Available online at https://www.mdpi.com/.