ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse242942020.pdf
https://doi.org/10.30534/ijatcse/2020/242942020

Detecting Phishing URLs using Machine Learning & Lexical Feature-based Analysis

Mohammad Alshira'H[1], Mohammad Al-Fawa'reh[2]
[1]Prince Hussein Bin Abdullah College for Information Technology, Al alBayt University, Mafraq, Jordan, alshirah@aabu.edu.jo
[2] King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan, fawareh@outlook.com

ABSTRACT
Phishing URLs is one of the greatest threats for cybersecurity professionals and practitioners. This requires hold hands together, make great efforts, and use current technology to help identifying Phishing URLs and control the spread of this threat. Many researchers have investigated various machine learning techniques to tackle this threat. However, there are many difficulties and obstacles of using machine learning. The proposed approach detects Phishing URLs through analyzing URLs to extract lexical characteristics features. Afterward, apply machine learning approach based on the extracted features. The dataset was collected from different sources, it includes four different attack scenarios: Defacement, Spam, Phishing, Malware. However, in this research, the focus was on Phishing URLs. The dataset was used as an input for various machine learning and statistical detectionmodels (RF: Random forest, DT: Decision Tree Classifier, GNB Gaussian Naive Bayes, KNN: k-nearest neighbour, Logistic regression, SVC: Support Vector Classifier, QDA: Quadratic Discriminant Analysis, Perceptron, SMOTE: Synthetic Minority Oversampling Technique) . These models were employed to predict Phishing URLs based lexical characteristics features. The result indicates a relatively good accuracy rate. The Random forest (RF)modelhas produced the best accuracy (98%) compared to the other detection models. As well as, the RF has produced the best precision and recall (98%) respectively.

Key words: Cybersecurity, Lexical analysis, Machine learning, Phishing

1. INTRODUCTION

The Internet has grown from a small number of networks to a worldwide network within excess of two billion users [1]. As it has expanded, it has modernized how individuals communicate with each other, how they do business, and how they provide services. However, there is no book of law for people's behavior on the Internet, even there is no central Internet control or authority. It is full of threats and suspiciously behavior [1]. Fraud is one of the expanding problems on the Internet community. Phishing is a type of fraud in which a phisher attempts to deceive the victim into giving sensitive information such as date of birth, ATM pin numbers, passwords, and bank account numbers. Phishing use spam emails and fake websites to trick victims to provide this information. The acquired information is exploited to steal individual identities or gain unauthorized access. Thus, this escalating threat result in billions of dollars in losses each year [1]. Therefore, it has risen the need for more cybersecurity efforts to secure the websites ([28], [29]). According to Google online service called Google's Safe Browsing technology, where they scanned billions of URLs to detect unsafe URLs. In this technology, when it recognized unsafe URLs using machine learning approach, it displays threats warning in web browsers. Daily, they discoveredthousandsof new suspicious URLs addresses, many of these URLs are compromised valid URLs [2]. There are several approaches to detect risky URLs on the internet. One of the common approaches is the blacklisting technology, and Whitelisting technology; blacklisting is widely used by many online services. However, along with other weaknesses in blacklisting, it is incapable to identify guided attacks and new suspicious URLs which are not previously considered as blacklisted URLs [3]. Moreover, whitelisting technology is the contradictory of blacklisting technology; the list of URLs, e.g. domain names, is a list of what can penetrate on a system. a domain names whitelist is a list of URLs that are legal to display. Currently, the growths in Artificial intelligence and machine learning area have promoted the concentration in its application to tackle many issues in cybersecurity. Machine learning has been using to detect suspicious URLs addresses. As mentioned above, a Google Safe Browsing utilizes a machine learning approach to classify suspicious URLs addresses. Likewise, this approach is used by many researchers and several studies have been performed in this area [4].

2. PROBLEMS WITH CURRENT TECHNIQUES

Regardless of the massive efforts and contributions in the field of detecting suspicious URLs, there are many challenges

and problems remain open questions; Firstly, the main problem is the huge volume of URLs in the world. There are trillions of URLs addresses on the internet, handling such large number of these URLs addresses is a challenging [5].Secondly, extracting appropriate URLs features. The extracted lexical features are characteristics of the URL address, which can contain a URL length, length of hostname, the name of top-level domain. Additionally, the presence of several special characters or keywords in a URL address. Choosing the suitable features is very significant for the performance of the classifier. Furthermore, collecting the features require large time and efforts. Thirdly, the transient of the presence of suspicious URLs. According to McGrath and Gupta [6], the suspicious URLs are being accessible (live) for temporary of time. Thus, it is crucial to find an effective technique to collect the suspicious URLs. Fourthly, blacklisting & whitelisting are not able to classify new suspicious URLs which are not previously considered as blacklisted or whitelisted URLs.

3. BACKGROUND

The URLs features that can be extracted include "Content-based", "Lexical-based", and "Host-based" features. Collecting "Content-based" features require downloading the full website pages. Canali et al and Eshete et al ([7], [8]) presented an experimentation for analyzing the content of webpages including HTML and JavaScript. In this study, they obtained that URLs Content-based features were created based on HTML file structure, the presence of specific "JavaScript" commands, or some elements of "ActiveX". URLs "Lexical-based" features can be obtained from the string of URL names. Specifically, URLs classification distinguishes benign URLs from suspicious ones consistent with their string structure. For example, URLs length, binary features, domain name, and special characters. Furthermore, Patil study [9] extracted the URLs lexical-based features from of HTML webpages content. URLs host-based features are usually obtained by requesting it from DNS servers. These features are about the domain name , website location, IP addresses , live time, and update dates. Suspicions URLs be subject to recurrently change their location and live appearance in a quick time. Consequently, host-based features are more valuable in making precise classification for URLs. Some host-based features can be clearly extracted from a webhost, for instance, speed of the connection, IP addresses. In Sahoo et al [10], they indicate that it is challenging task to modify website IP addresses for each individual attack. Therefore, IP addresses information are more helpful for URLs classification process.

4. RELATED WORK

The literature review in this area have indicated that there are several research directions that have been purposed to afford a safe browsing experience for individual users on the internet. The scope of the proposed approach is concentrated only on machine learning approach through lexical Feature-based Analysis. Consequently, the related work is explored to highlight this specific area. As mentioned earlier, lexical feature plays a significant role in the effort of a security practitioner and appropriately selected features provides a good rate of accuracy in term of classifying URLs. Many researchers have presented a pioneering application of machine learning approach in securing websites [15, 16, 17].In Sahoo et al [10], they emphasized that the most available features of URLs are the lexical-based features. Furthermore, Sorio and Medvet [11], they have examined the impact of lexical feature on detecting suspicious URLs. In [15], a study was presented on using data mining techniques and machine learning for intrusion detection. In [12], they examined different machine learning techniques for detecting suspicious URL addresses. In [13, 14], they presented a comprehensive overview of URL Phishing. However, they do not broadly review the lexical feature representation or the aspects of machine learning. In [18], they concentrated on the section of key feature for detecting suspicious URL addresses. In Sahoo et al [10], they highlighted that URLs collection is the main limitations for detection approaches based on machine learning. Since not all lexical features can be straightforwardly gathered owing to their weight and the massive number of URLs addresses on the internet.

5. PROPOSED APPROACH

The approach proposed in this paper analyses the websites URLs, through extracting URLs feature representations, and then train the detection on the data of malicious and normal URLs This section describes the methodology that is used to perform this experiment of the proposed approach. Furthermore, set up experimental configuration and consequent sources. The steps in the following had been implemented to accomplish the research objective:

- Build the dataset by Collecting data from different sources.
- Employ the selected statistical model and machine learning methods (describes in Section below) individually to evaluate the acquired data from the dataset in order to detect Phishing URLs.
- Evaluate the performance (Accuracy, Time, Precision, and Recall) of each model to choose the most suitable one.

5.1 Methodology

The methodology of conducting this research includes four phases, specifically, Data collection, Data pre-processing (which includes subphases namely, Removing Duplicate, Feature extracting, Handling Missing Data, Data normalization), and finally Training and Testing phase. The goal of this research can be achieved by applying various machine learning and statistical models. These models were employed to predict Phishing URLs based lexical characteristics features. Finally, the performance of each model is calculated. The proposed approach phases are illustrated in in Figure 1 below.

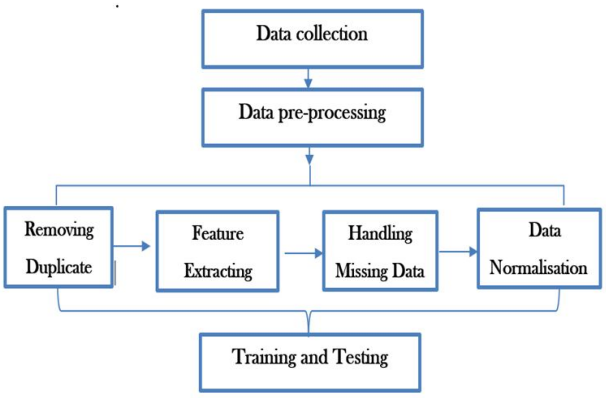

Figure 1: Proposed approach methodology

Machine learning & statistical models are commonly used in various applications owing to its robust superior prediction accuracy [24] [25]. In the proposed approach in this paper, the following models were used:

RF: "Random forests or random decision forests are an ensemble learning method for classification".

DT: "Decision Tree Classifier is a class capable of performing multi-class classification on a dataset".

GNB: Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.

KNN: k-nearest neighbors' algorithm is a non-parametric method used for classification and regression.

Logistic Regression: is a predictive analysis which calculates the likelihood of one dependent variable based on the of one or more independent variables . regardless of its name, Logistic Regression is a linear model for classification rather than regression.

SVC: Support Vector Classifier is a class capable of performing binary and multi-class classification on a dataset.

QDA: Quadratic Discriminant Analysis is a classic classifier, with, as its name suggest, a quadratic decision.

Perceptron: is simple classification algorithm appropriate for large level learning.

SMOTE: Synthetic Minority Over-sampling Technique (SMOTE) is a standard over-sampling method that was produced to enhance random over-sampling.

5.1.1 Data Collection

Finding anappropriateURLs dataset is a difficulty since many datasets are internally shared owing to their privacy issues. Besides, many datasets are heavily anonymized and do not reflect existing Cybersecuritytrends. Furthermore, they lack of certain lexical & statistical characteristics. Therefore, the optimal dataset does not exist yet. Some previous studies [19,20,21] use an implementation of Support Vector Machines SVMLight. However, the application of this dataset was unfeasible since URLs lexical features were converted into massive digital data. To overcome these problems and challenges, a systematic approach has been developed to generate dataset that permits the analyze, test, and evaluate of Malicious URLs focusing on their lexical features. For this purpose, data were obtained from Alexa top websites [23]. The final dataset includes different attack scenarios including Normal, Spam, Phishing, Malware. Although, this research focuses on Phishing attacks. Figure 2 illustrates the number of URL samples gathered from each source.

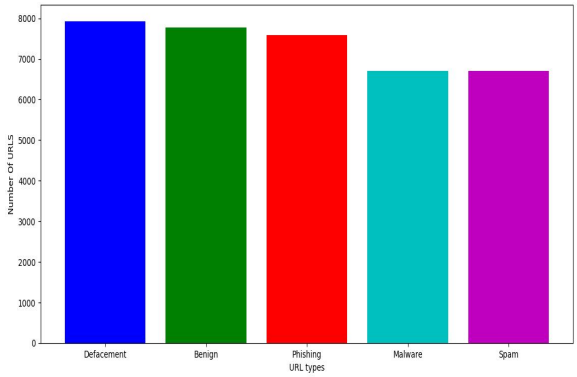

Figure 2: Number of samples collected from each source

The lightweight approach was explored to detect and categorise the suspicious URLs along with their attack type and indicate that "lexical-feature" is efficient proactive detection approach for these URLs. The impact of the "obfuscation techniques" on suspicious URLs was analysed, this helps to understand the type of obfuscation technique. The concentrate was mainly on two types of URLs:

- Benign URLs: Over 35,300 benign URLs were gathered from Alexa top websites . The domains have been passed through a "Heritrix web crawler" to extract the URLs. Over half a million unique URLs are crawled initially and then passed to eliminate duplicate and domain only URLs. Afterward, the extracted URLs have been examined through Virustotal to filter the benign URLs.
- Phishing URLs: Over 10,000 phishing URLs were taken from Open Phish which is a source of active phishing sites.

Obfuscation was usually used to mask malicious URLs; in details, attackers try to prevent static analysis of lexical features through using obfuscation techniques for malicious URLs that are statistically identical to benign ones. In this research the obfuscation techniques of URLs were studied for the purpose of detecting malicious behaviour. Phishing URLs were primarily studied to discover what kind of obfuscation technique were being applied to the URLs.

5.1.2 Data pre-processing

Thisphase was considered as most important phase. In this phase, data was converted to form that can be suitable format for the selected machine learning models. This helps for including data-only important features, and preparation of

data set for classification task. Data pre-processing includes the following steps:

5.1.2.1 Removing Duplicate Data

This phase deals with inappropriate and missing data values such as {NAN, Infinity}, this phase was required to eliminate duplicates from a dataset while training a Machine Learning models.

5.1.2.2 Feature Extracting

In this phase, raw data were transported into numerous features and all textual features were transformed into digital. This phase includes some feature engineering tasks for example, finding the mean, median, or ratio. Feature engineering is the technique of using domain knowledge to extract URL lexical features from raw data. This process of "mathematical operation on extracted features for creating other independent variables. Mathematical operations such as finding the mean, normalizing, or calculating ratio are usually applied by machine learning practitioners for boosting the accuracy and computational efficiency of classifier models" [22]. A list of useful features is presented below. Finally, benign URLS was symbolized by 1 and Phishing URLs by 0, due to some machine learning algorithms cannot accept textual data as input.

['Querylength'
'domain_token_count'
'path_token_count'
avgdomaintokenlen'
'longdomaintokenlen'
'tld''avgpathtokenlen'
'charcompvowels'
'charcompace'
'ldl_url'
'ldl_domain'
'ldl_path'
'ldl_filename'
'ldl_getArg'
'dld_url'
'dld_domain'
'dld_path'
'dld_filename'
'dld_getArg'
'urlLen'
'domainlength'
'pathLength'
'subDirLen'
'fileNameLen'
'this.fileExtLen'
'ArgLen'
'pathurlRatio'
'ArgUrlRatio'
'argDomanRatio'
'domainUrlRatio'
'pathDomainRatio'
'argPathRatio'
'executable'
'isPortEighty'
'NumberofDotsinURL'
'ISIpAddressInDomainName'
'CharacterContinuityRate'
'LongestVariableValue'
'URL_DigitCount'
'host_DigitCount'
'Directory_DigitCount'
'File_name_DigitCount'
'Extension_DigitCount'
'Query_DigitCount'
'URL_Letter_Count'
'host_letter_count'
'Directory_LetterCount'
'Filename_LetterCount'
'Extension_LetterCount'
'Query_LetterCount'
'LongestPathTokenLength'
'Domain_LongestWordLength'
'Path_LongestWordLength'
'sub Directory_LongestWordLength'
'Arguments_LongestWordLength'
'URL_sensitiveWord'
'URLQueries_variable'
'spcharUrl'
'delimeter_Domain'
'delimeter_path'
'delimeter_Count'
'NumberRate_URL'
'NumberRate_Domain'
'NumberRate_DirectoryName'
'NumberRate_FileName'
'NumberRate_Extension'
'NumberRate_AfterPath'
'SymbolCount_URL'
'SymbolCount_Domain'

```
'SymbolCount_Directoryname'
'SymbolCount_FileName'
'SymbolCount_Extension'
'Entropy_URL'
'SymbolCount_Afterpath'
'Entropy_Domain'
'Entropy_DirectoryName'
'Entropy_Filename'
'Entropy_Extension'
'Entropy_Afterpath']
```

5.1.2.3 Handling Missing Data

After converting all URLs to feature and labels all duplicate values were removed. Subsequently, the missing and inappropriate values were replaced by median values. There are many methods to handle missing values in the data set such as complete-case (CC) analysis. Therefore, every data row contains at least single missing values the entire row was deleted. Figure 3 shows the after-Pre-processing results for the selected samples.

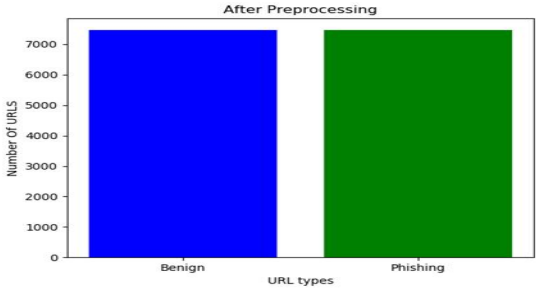

Figure 3 : After-Pre-processing

5.1.2.4 Data Normalization

In this phase, normalise/scale continuous values amongst all the other features is also normal trend, so that the machine learning algorithm trains on data that is all within the same feature space. This phase use Minimax() data scale, typically [0,1].**Xnorm = X – Xmin/Xmax – Xmin**

5.1.3 Training and Testing

The dataset was split into two different training sets and test sets according to best practices. Firstly, 20% of the data considered as a test set and 80 % as a training set, the dataset was separated out.

5.2 Performance Evaluation

Evaluation plays a principal role in measuring the performance of the selected models. There are many traditional systems of prediction measurement that are used for assessing the selected models for instance Accuracy, Precision, Recall, and F_1 score. Each prediction can be one of these four categories [26]:

- **True Negative (TN):** the number of instances that are classified false and detected as false. TN is defined as the ratio of negatives instances that are categorized correctly
- **True Positive (TP):** the number of instances that are classified true is detected as true. TP rate is the percentage of positive instances that are accurately categorized.
- **False Positive (FP):** The number of instances that are wrongly detected as positive. FP rate is the percentage of negatives cases that are incorrectly classified as positive.
- **False Negative (FN):** the number of positive cases that are predicted as negative. (FN) rate is the percentage of positives cases that are incorrectly classified as negative

Based on the above categories; *Accuracy, Precision, Recall, and F1 score* can be calculated as follows:

- **Accuracy**: the ratio of cases that are correctly detected, total predictions that are correctly detected. It is defined using the equation.

$$Accuracy = TP + TN / (TP+FP+TN+FN)$$
$$= \text{Correct Prediction/Total Prediction.}$$

- **Precision**: calculates the number of positive instances that predicted positive cases. Precision illustrates how accurate or precise the selected model is. It is defined using the equation:

$$Precision = TP / (TP+FP)$$
$$= TP / \text{Total Predicted Positive}$$

- **Recall**: estimates how many of the actual instances a selected model predicted as positives (TP).

$$Recall = TP/ TP+FN$$
$$= TP / \text{Total actual Positive}$$

F_1 Score: balance between recall and precision values. Moreover, it is better than accuracy, as with an F1 score is not considering any TN cases.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

6. RESULTS AND DISCUSSION

Python API has been utilized to construct the various models in the proposed approach [27]. Experiments were made using 64-bit Windows 10 computer with 16 GB RAM and 2.60 GHz CPU, the machine learning models were implemented using Python 3.7.3 , Numpy 1.16.2 , Scipy 1.2.1 , and SPYDER 3.3.3.the training testing (80%, 20%). Class 0 means Normal URLs where class 1 means malicious URLs. eight machine-learning algorithms were employed, including RF, DT, QDA, NB, Perceptron, LR, KNN, and SVM. **Accuracy**, **Recall, Precision, F_1 score, Testing time, and Training time**are shown respectively in Figures (4 – 9)for the selected detection models. The results in Table 1 indicate that when the classier runs with **RF**, **DT**, **QDA**, **NB** , **Perceptron**, **LR**, **KNN** and **SVM**, the average accuracies are 98%, 97%, 78%, 77%, 88%, 98%, 95% and 98%, respectively as shown in Table 1.

QDA, Naive Bayes (NB) are statistical classical machine learning algorithms. RF and KNN accuracy values were almost the same with 98%. Naive Bayes results were the lowest among all because its classification speed (high) leads

to suboptimal result. The results in Table 2 shows that RF was the best algorithm with 99% precision for class 0 and 98% for class 1 predicting (normal traffic/no attack). RF is one of machine learning algorithms that ensemble learning algorithms, which mean that they more than one classifier to obtain the best classification result. For predicting class 1 (malicious traffic/attack), RF and KNN algorithms have the highest average precision over all with 98%, while GNB precision, Recall and F1 are the lowest with 77%, 83%, and 76% respectively. The reason why RF overall results were the best was that the splitting reduces the variance by averaging the deep decision trees choosing a value for the trees split to boost the performance of the last decision tree. KNN runtime in the testing phase was the highest with (5.6344359), it is well known as a slow machine learning algorithm because of its dense distance calculations, where SVM was the slowest in the training phase.

Table 1 : Detection Results

Detection model	Training Time	Testing time	Accuracy	P	R	F1 Score
RF	0.31	0.0079052	98%	98%	98%	98%
DT	0.50	0.0019937	97%	97%	97%	97%
QDA	0.19	0.0195832	78%	77%	84%	76%
GNB	0.10	0.0139635	77%	77%	83%	76%
Perceptron	0.16	0.189508	88%	89%	90%	88%
KNN	1.05	5.6344359	98%	98%	98%	98%
LR	0.22	0.2156732	95%	95%	95%	95%
SVM	4.37	5.4477117	96%	96%	96%	96%

Table 2 : Overall Detection Results

Predication Model	Training Time	Testing time	Class	precision	recall	F1	Accuracy
RF	0.31	0.0079052	0	0.99	0.97	.98	98%
			1	0.97	0.99	.98	
DT	0.50	0.0019937	0	0.98	0.97	0.97	97%
			1	0.97	0.98	.97	
QDA	0.19	0.0195832	0	0.99	0.69	0.82	78%
			1	0.56	0.56 0.99 0.71	0.56 0.99 0.71	
GNB	0.10	0.0139635	0	0.99	0.69	0.81	77%
			1	0.56	0.98	0.71	
Perceptron	0.16	0.189508	0	0.79	0.98	0.87	88%
			1	0.99	0.82	0.89	
KNN	1.05	5.6344359	0	0.99	0.96	0.98	98%
			1	0.96	0.99	0.97	
LR	0.22	0.2156732	0	0.97	0.94	0.95	95%
			1	0.94	0.96	0.95	
SVM	4.37	5.4477117	0	0.98	0.95	0.96	96%
			1	0.95	0.98	0.96	

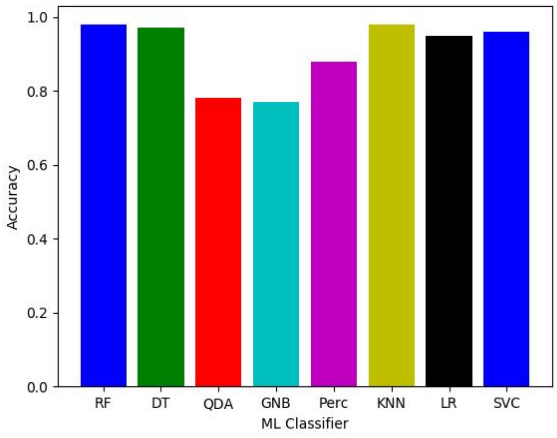

Figure 4 : Accuracy

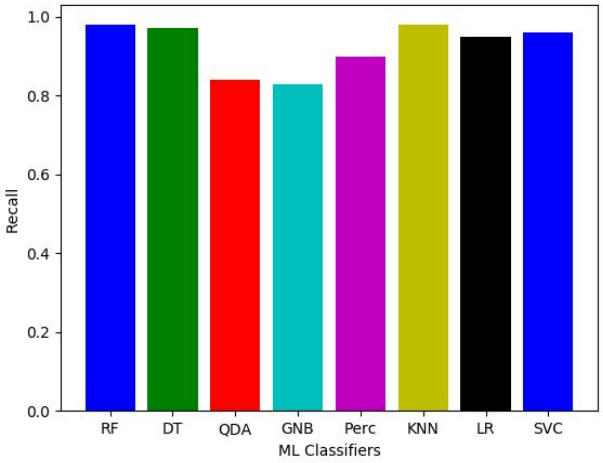

Figure 6: Recall

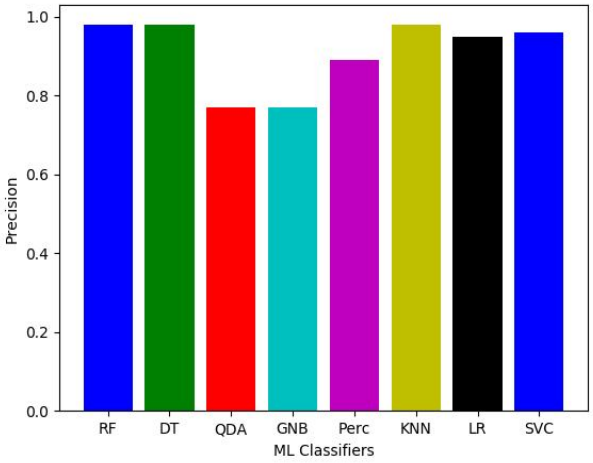

Figure 5: Precision

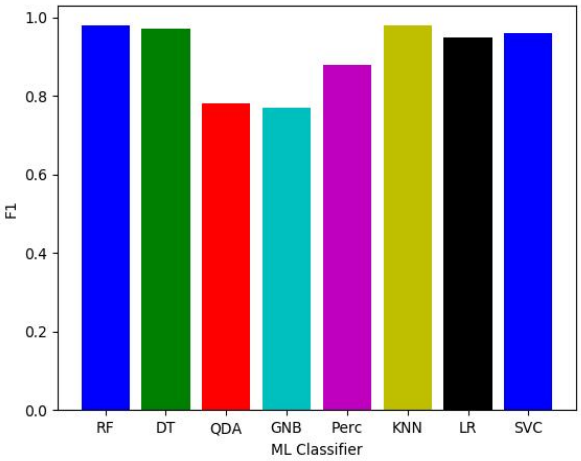

Figure 7: F_1 Score

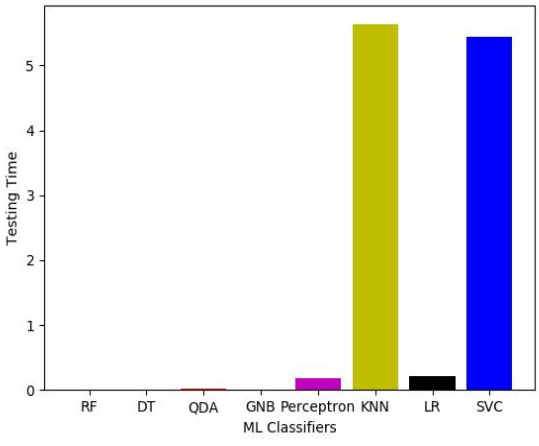

Figure 8: Testing Time

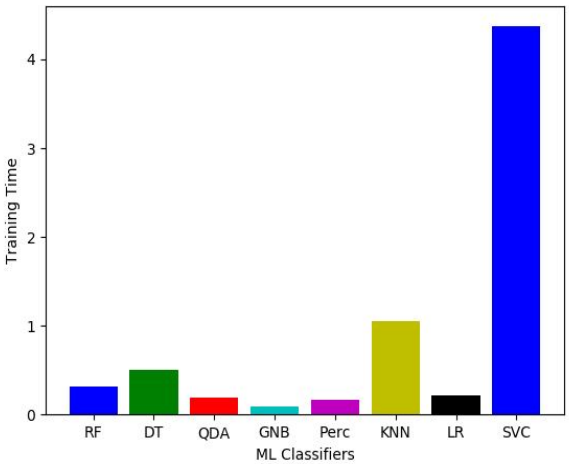

Figure 9: Training Time

7. CONCLUSION

Massive efforts have employed to combat a large extent of malicious URLs in the internet. The results were analyzed to support the global challenge in detecting potential suspicious URLs in order to protect the cyberspace from criminals, and to provide best & safe service. Numerous machine learning detection models were employed based on lexical features to classify the URLs' status. The anticipated opportunities for future work as follow, firstly, the opportunity is to employ these models in ISPs protection level with aim of detecting suspicious URLs in a speedy manner, provide reliable method to reduce in the precipitous spread of suspicious URLs. Secondly, employing the examined models on larger size of URLs dataset and examine its consequent factors.

REFERENCES

1. Michael E. Whitman and Herbert J. Mattord.**Principles of Information Security**, 6th Ed, Course Technology Press, Boston, MA, USA.2017
2. Safebrowsing.google.com. **Google Safe Browsing**. [online] Available at:https://safebrowsing.google.com/> [Accessed 29 May 2020].
3. Chen, C. M., Huang, J. J. and Ou, Y. H.**Efficient suspicious URL filtering based on reputation**, Journal of Information Security and Applications. Elsevier Ltd, 20, pp. 26–36, 2015 https://doi.org/10.1016/j.jisa.2014.10.005
4. Wen, A.**Keeping your company data safe with new security updates to Gmail**, Google Blog, [online]. Available at: http://www.googblogs.com/author/andy-wen [Accessed 29 May 2020].
5. Sullivan, D. and Sullivan, D. **Google: 100 Billion Searches Per Month, Search to Integrate Gmail, Launching Enhanced Search App For iOS**. [online] Search Engine Land. Available at: https://searchengineland.com/google-search-press-129925[Accessed 29 May 2020].
6. McGrath, D. K. and Gupta, M.**Behind phishing: An examination of phisher modi operandi**. In LEET: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2008
7. Canali, D., Cova, M., Vigna, G. and Kruegel, C.**Prophiler : A Fast Filter for the Large-Scale Detection of Malicious Web Pages Categories and Subject Descriptors**, Proc. of the International World Wide Web Conference (WWW), pp. 197–206. 2011
8. Eshete, B., Villafiorita, A. and Weldemariam, K.**BINSPECT: Holistic analysis and detection of malicious web pages**, Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 106 LNICS, pp. 149–166, 2013
9. PATIL, D. R. and PATIL, J. B.**Malicious Web Pages Detection Using Static Analysis of URLs**, International Journal of Information Security and Cybercrime, 5(2), pp. 57–70, 2016 https://doi.org/10.19107/IJISC.2016.02.06
10. Sahoo, D., Liu, C. and Hoi, S. C. H.**Malicious URL Detection using Machine Learning: A Survey**, pp. 1–21. Available at: <http://arxiv.org/abs/1701.07179> [Accessed: 29 May 2020], 2017
11. Sorio, E., Bartoli, A. and Medvet, E. **Detection of hidden fraudulent URLs within trusted sites using lexical features**, Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013, pp. 242–247, 2013
12. S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. **A comparison of machine learning techniques for phishing detection**. In Proceedings of the eCrime Researchers Summit, 2007
13. Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. **Phishing detection: a literature survey**. IEEE Communications Surveys & Tutorials, 2013
14. Dharmaraj Rajaram Patil and JB Pati.**Survey on Malicious Web Pages Detection Techniques**. International Journal of u-and e-Service, Science and Technology, 2015
15. Anna L Buczak and Erhan Guven. **A survey of data mining and machine learning methods for cyber security intrusion detection**. IEEE Communications Surveys & Tutorials, 2016 https://doi.org/10.1109/COMST.2015.2494502
16. Sumeet Dua and Xian Du. **Data mining and machine learning in cybersecurity**. CRC press. 2016
17. Jayveer Singh and Manisha J Nene. **A survey on machine learning techniques for intrusion detection systems**. International Journal of Advanced Research in Computer and Communication Engineering, 2013
18. Hiba Zuhair, Ali Selamat, and Mazleena Salleh. **Feature selection for phishing detection: a review of research**. International Journal of Intelligent Systems Technologies and Applications, 2016
19. H. Dong J. Shang D. Yu and L. C. Lu. "**Beyond the blacklists: Detecting malicious url through machine learning**" in BlackHat Asia, 2017
20. Vanhoenshoven, F., Napoles, G., Falcon, R., Vanhoof, K. and Koppen, M.**Detecting malicious URLs using machine learning techniques**, IEEE Symposium Series on Computational Intelligence (SSCI), (December), pp. 1–8. 2016
21. Ma, J., Saul, L., Savage, S. and Voelker, G. **Identifying suspicious URLs: an application of large-scale online learning**, … on Machine Learning, pp. 681–688. 2009
22. Brink, H., Richards, J. and Fetherolf, M.**Real-World Machine Learning**. 1st ed. Manning Publications. 2016
23. Alexa.com. 2020. Alexa - Top Sites. [online] Available at: <https://www.alexa.com/topsites> [Accessed 30 May 2020].
24. Pedregosa, F. Varoquaux, G. Gramfort, A. Michel, V. Thirion, B. Grisel, O. Blondel, M. Prettenhofer, P. Weiss, R. Dubourg, V. Vanderplas, J. Passos, A.

Cournapeau, D. Brucher, M. Perrot, M. Duchesnay, E. **scikit-learn, Scikit-learn: Machine Learning in Python**, Journal of Machine Learning Research, volume 12, pages={2825--2830}, 2011

25. Palmer, A., Jimenez, R., Gervilla, E., **Data mining: Machine learning and statistical Technique**, Knowledge Oriented Applications in Data Mining, Prof. Kimito Funatsu (Ed.) , pp.373-396, 2011
26. J. Han and M. Kamber, **Data mining: concepts and techniques**, San Francisco, Morgan Kaufmann Publishers, 2001.
27. Matthes E. **Python Crash Course: A Hands-On**, **Project-Based Introduction to Programming**, No Starch Press, ISBN-10: 1593276036, ISBN-13: 978-1593276034. 2016
28. R. K. Alqurashi, M. A. AlZain, B. Soh, M. Masud, J. Al-Amri.**Cyber Attacks and Impacts: A Case Study in Saudi Arabia**, International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 1, pp. 217–224, 2020 https://doi.org/10.30534/ijatcse/2020/33912020
29. Darus, Mohamad Yusof, Mohd Afham Omar, Mohd Farihan Mohamad, Zulhairi Seman, and Norkhusahini Awang. **Web Vulnerability Assessment Tool for Content Management System**. International Journal 9, no. 1.3, 2020