Finding effective classifier for malicious URL detection

Chunlin Liu
State Key Lab of Software Development Environment
School of Computer Science and Engineering,
Beihang University
Beijing, China
jackliu@buaa.edu.cn

Lidong Wang
National Computer Network Emergency Response
Technical Team/Coordination Center of China
Senior Engineer
Beijing, China
wld@cert.org.cn

Bo Lang
State Key Lab of Software Development Environment
School of Computer Science and Engineering,
Beihang University
Beijing, China
langbo@buaa.edu.cn

Yuan Zhou
National Computer Network Emergency Response
Technical Team/Coordination Center of China
Senior Engineer
Beijing, China
13910020110@163.com

ABSTRACT
Malicious URL is an important security issue to the Internet, which has a significant economic impact. By now, it is still a challenging problem. In this paper, we propose that combining statistical analysis of website URLs with machine learning techniques will give a more accurate classification of malicious URLs. We focus on the Character features of malicious URLs by statistical methods to obtain char distribution features and structural features. Then, In order to find effective classifier for malicious URL detection, we use six different classifiers to perform cross training. The experimental results on our data set demonstrate that the combination of the URL features extracted in this paper and the Random Forest classification algorithm can achieve 99.7% precision with a false positive rate of less than 0.4%. We also show that these features render better performance than the previously used features which combine lexical features and structural features and render similar results to the N-Gram or TF-IDF based features. Besides, we adjust the number of iterations of random forest and random choice characteristic number of random forest in experiment.

CCS Concepts
• **Security and privacy → Network security**

Keywords
URL classification; char distributions; Random Forest

1. INTRODUCTION
In recent years, the rapid development of the Internet has created enormous convenience for people's daily life. At the same time, convenient network service also attracted the network attackers through phishing, spam and malicious software and other ways to profit illegally. Although the purposes and means of these illegal activities are different, but they require unsuspecting users to access malicious URLs provided by attackers. Even if the user's computer to install anti-virus software or firewall, which prevent most of the malicious attacks, various forms of malicious URLs, such as “viruses, spam, phishing sites, especially web hanging horse”, the users inadvertently will be caught and cause a huge loss of information and property. Therefore, how to detect malicious URLs, to reduce the loss of ordinary users and maintain public Internet security, is of great significance.

Beyond blacklisting heuristics, domestic and foreign scholars have conducted a lot of research work on malicious URL detection. There are two main research ideas in the literature. One is that attempt to find the best classification features which can be extracted only from the URL, and the other is that visit the potentially malicious site and extract features from the crawled content. For example, Choi et al. [1] uses a lot of lexical features, host-based features, link breadth features (the number of websites linked to this site) and web content features to achieve a 98% accuracy. Different from [1], Ma, J et al.[2] analyze the vocabulary of suspicious URL (Lexical Features) and host attributes (Host - Based Features), and use the bag of words model (Bag-of-Words) to represent features, and obtain thousands of features. In lexical features, on the one hand consider the host name length, URL length, URL point number; on the other hand, for each word symbol host and path in URL, using the bag of words model to build a two valued feature. In the host features, the IP address attributes, WHOIS attributes, domain name attributes and geographical location attributes are considered. Finally, the accuracy rate of 95-99% is obtained. Zhang Y et al. [3] uses a web content based feature to achieve a 94% accuracy. Fu et al. [4] based on Web image similarity detection, this method calculates the similarity between the images by EMD distance, but it can only be applied to the case where malicious URL is very similar to a benign URL and have certain limitations.

Then, let’s look at some other examples. Garera et al. [5] analyzed the URL structure of phishing sites, and introduced the selection process of feature set in detail. Logistic Regression Filter was used to classify URL. URL structure of phishing sites is analyzed, and 4 types of URL structure are obtained. Feature set consists of 18 features: page feature, domain name feature, type feature and word feature. The page features with Google search engine, select Page Rank of URL , Page Rank of Host, PageRank Present in Crawl Database, Page Present in Index ,the two page quality evaluation (Page Quality Score) a total of 6 characteristics; In the domain name feature selections ,the domain name in the white list

ICMSS 2018, January 13–15, 2018, Wuhan, China

ACM ISBN 978-1-4503-5431-8/18/01…$15.00
DOI: https://doi.org/10.1145/3180374.3181352

represents 1 feature ; Type feature select the type of I, II, and III ,3 features; Word features select secure, account, webscr, login, ebayiapi, signin, banking and confirm 8 words . Finally, the accuracy rate of 97.31% was obtained. Kan et al. [6] uses word level n-grams features to classify and obtain 76% accuracy when there is no combination of web page text features. When combined with web page text features, 81% accuracy is achieved. Aldwairi et al. [7] extracts n-grams, TF-IDF and content features from URL, and achieves an average accuracy of 87%.In 2010, Liu Gang et al. [8] took link relations, keywords sorting relations, text similarity relations, hierarchical similarity relations as statistical features, using unsupervised learning algorithm DBSCAN (Density-Based Spatial Clustering of Applications with Noise) for fishing URL attack target recognition. Experimental results show that the proposed method can identify 91.44% phishing targets, and the false positive rate is controlled at 3.4%.

Although the above studies made some progress in the detection of malicious URL, but there are still some shortcomings, such as no general features; link breadth, host-based features need to get through third party tools , so these features are expensive to acquire and are less efficient while comparing with lexical features, n-grams, TF-IDF, structure characteristics, character distribution. Web content features require access to potentially malicious URL at a much higher cost and lower efficiency. In this paper, a feature extraction scheme is proposed, which combines the character frequency and structural features with small acquisition cost, and performs the model contrast experiment and parameter optimization experiment.

The contributions of our paper are as follows.

- The first novel aspect of our work is our choice of aspects to focus on, specifically we focus on character distributions.

- We use several machine learning algorithms and two different real data sets.

The rest of the paper is organized as follows. Related work is presented in Section 2. Section 3 presents our classifiers, the datasets they were trained on, the algorithms used, and covers our feature extraction process. Section 4 presents our analysis and results for our classifiers and compare our experimental results with other results. Section 5 concludes with implication of the results.

2. RELATED WORK

Many researchers have presented their work on malicious URLs detection, so there are some similarities between our work and theirs, but also some key differences. The smart reader will see from the presentation below, there are also several similarities among the features used by previous researchers, since the URL is basically a string of symbols.

We use some same features and classifiers as the below authors do. The authors of [9] do use some structural features such as IP address and lengths of domain and URL and a strict subset of the classifiers (SVM, logistic regression, and a naive Bayes classifier). Unlike their work we have not used a bag-of-words feature implementation because our goal is to demonstrate the effectiveness of a small feature vector and improve efficiency. The work by [10] has been influential in the detection of phishing websites. There is some overlap between their work and ours, specifically in their URL analysis section of their methods. We share a set of four features: whether or not the URL is an IP address (also used by Ma et al. and others), the number of dots in the domain name, the length of domain in the malicious URL, the depth of a URL path. However, bigger differences between our work and theirs are that they also use the PageRank and calculate domain similarity and test their approach on smaller datasets.

Some recent studies are as follows. Recently in [11] the authors are able to accurately classify 99.3% of their dataset, with a false positive rate of less than 0.4% by using a two-sample Kolmogorov-Smirnov test along with other features. Our method can get slightly better results by using different feature combination. In [12], the authors have created a system based only on URLs to detect phishing websites. The authors achieve good accuracy, but lower than ours, with a confidence weighted classifier, using a bag-of-words implementation that generates a huge set of 369,000 features. Additionally, the authors have some hand selected features that correspond to ours, such as IP detection, and Length analysis. The authors of [13] extract n-grams and tf-idf scores from malicious URLs. They train and test using Logistic Regression, Linear SVC, open-source XGBoost, and Multilayer Perceptron and their classifiers achieves good accuracy with low false true rate. We get similar results on their dataset with focusing on the character of malicious URLs.

3. CLASSIFIERS

3.1 Feature

This section discusses our feature extraction algorithms. Prior to feature extraction “http://” and “https://” were removed from the URL, so as not to the features such as number of punctuation, and to prevent URLs classified without such parts from being misclassified.

In malicious attacks, the evildoers always try to absorb victims into clicking a URL pointing to the malicious site. They usually obfuscate malicious URLs through various methods. Every method attaches some features to the malicious URLs and these features can differentiate it from a legal one. Therefore, the URL features are essential to detect the malicious activities. By analyzing the malicious URLs we collect, the prominent features of a malicious URL are listed as follows:

1) Mixing IP address in the malicious URL. According to 5000 malicious URLs and 5000 legitimate ones, we can find that legitimate URLs containing an IP address almost do not happen.

2) Obfuscating the domain with a mass of dots. Malicious URLs usually use lots of dots to confuse users, for example,http://https.secure.paymentaccount.customer.service.com .setup-onlinecustomerhelp.inc-faq.cgi-ticket-help.desk.updated-setup-connect.uroproct.com.br/myaccount/.This kind of URL rarely exists in a legitimate URL.

3) Confusing users with abnormal depth of a URL path. In other words, there are many “/” in malicious URLs.

4) Confusing users with some special characters, such as “@”, “~”, “-”. These special characters are often found in malicious URLs.

5) Abnormal length of domain in the malicious URL. Under normal circumstances, the string appearing between the http:// and the first “/” is considered as domain and the length is relatively longer in parts of malicious URLs.

6) The character distribution of benign URL and malicious URL is quite different. We have statistics of each character in URL (A-Z (ignore case), 0-9) the number of occurrences, then the total length divided by, then calculate the entropy, to construct a

set of features, which greatly increases the size of the feature vectors, but also received a higher precision rate.

A total of 41 features are identified, of which the first 5 are structural features, and the latter 36 are letters (a-z||A-Z, ignore case) and numbers (0-9) frequencies. The first 5 features can be easily obtained by regular expression matching, and the character distribution frequency vectors need to be obtained by statistical analysis.

3.2 Machine Learning Algorithms

We use the WEKA library to construct our models. Using the WEKA library allow us freedom to quickly compare and contrast the different machine learning algorithms and their results. By comparing their results, we can obtain the effective algorithms in this scene. We use multiple different algorithms types in our approach to ensure that a large scope is covered. We examined the following tree based algorithms Random Forest and J48. We looked at functional algorithms such as Logistic Regression, SVM and Multilayer Perceptron. Finally, we looked at Na ïve Bayes.

Random Forest: The random forest algorithm constructs a multitude of decision trees, and then during classification the algorithm returns the most often occurred class from the array of individual trees. The trees are all constructed using a random subset of features from the entire feature list. The method has been proven quite effective at classifying.

For most values, we used WEKA'S default implementation of the Random Forest class. We just adjusted the numFeatures and the numIterations to achieve the best result.

J48: Algorithm J48 generates a C4.5 decision tree for classification. We used the WEKA 3.8 default pruning threshold.

Logistic Regression: The logistic regression implementation in WEKA is a standard multinomial logistic regression model, with the addition of a ridge estimator. There are a few modifications to leCessie's algorithm in the WEKA implementation, such as the addition of weighted instances.

LibSVM: Weka and LibSVM are two efficient software tools for building SVM classifiers. Each one of these two tools has its points of strength and weakness. Weka has a GUI and produces many useful statistics (e.g. confusion matrix, precision, recall, F-measure, and ROC scores). LibSVM runs much faster than Weka SMO and supports several SVM methods (e.g. One-class SVM, nu-SVM, and R-SVM). Weka LibSVM (WLSVM) combines the merits of the two tools. WLSVM can be viewed as an implementation of the LibSVM running under Weka environment.

Multilayer Perceptron: A Classifier that uses backpropagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the output nodes become unthresholded linear units).

Na ïve Bayes: These classifiers have been used quite successfully in many real-world problems, but have been shown to be less effective than more current approaches such as boosted trees. The threshold values are calculated by default in the WEKA implementation, and we used these defaults for our experiments.

3.3 Datasets

For our experiments we used two different real data sets. The Data Set I consists of 29,000 URLs comprised of 12,483 malicious URLs from PhishTank.com, accessed on November, 2016 – February, 2017. We then combined that with 16516 legitimate URLs taken from digg58.com. The Data Set II comes from the authors of [13]. In a few datasets provided by the authors, we chose a slight imbalance remained dataset (PE) which consists of 62,573 URLs comprised of 37,667 benign URLs and 24,905 malicious URLs. Finally, we labeled the Data Set: 0 for benign, 1 for malicious.

3.4 Evaluation

Confusion matrix is a table that describes the classification results in detail. Whether the classification is correct or wrong, and the different categories are distinguished. For the two category, it is a 2*2 matrix, and for the N classification it is the n*n matrix. The two classification confusion matrix is shown in Table 1.

Table 1. The two classification confusion matrix

	Predicted as Positive	**Predicted as Negative**
Labeled as Positive	True Positive(TP)	False Negative(FN)
Labeled as Negative	False Positive(FP)	True Negative(TN)

The True Positive Rate (TPR), also known as sensitivity: the correct classification number of positive samples accounts for the proportion of the number of positive samples, namely: TPR=TP/ (TP+FN)

The False Positive Rate (FPR): the number of negative samples of misclassification accounts for the proportion of the whole negative sample number, namely: FPR=FP/ (FP+TN)

Accuracy (Accu): the correct classification number of samples accounts for the proportion of all samples, that is, Accu= (TP+TN) / (TP+FP+TN+FN).

Precision (Prec): the correct classification number of positive samples accounts for the proportion of all positive samples, that is, Prec=TP/ (TP+FP)

Recall rate (Rec): the proportion of positive samples with correct classification accounted for the number of positive samples, that is: Rec=TP/ (TP+FN)

F1-Score: the harmonic average of precision and recall, and its value is close to the smaller value of Precision and Recall

F1-Score = 2*precision*recall/ (precision + recall)

The abscissa of the ROC curve is false positive rate (FPR), and the ordinate is true positive rate (TPR). The reason why we consider the ROC curve is that the ROC curve has a good characteristic: when the distribution of the positive and negative samples in the test set changes, the ROC curve can remain unchanged. AUC (Area Under Curve) is defined as the area under the ROC curve. It is evident that the area is not greater than 1. The AUC value is used as an evaluation criterion because many times the ROC curve does not clearly show which classifier works better, and as a numerical value, the classifier with a larger AUC is better.

4. RESULTS AND ANALYSIS

In this section, we present the results obtained using each of the classifiers, and different features. We begin by presenting the results of combinatorial structural features with Random Forest, and the importance of structural characteristics is analyzed. Then, we present the comparison of features into three sections, character frequency and structure based methods, structure and

lexical based methods, and other methods. Finally, we adjust some parameters of the effective classifiers to experiment.

4.1 Importance evaluation of structural features

When the data set used in the experiment is public data set and uses the five structural features described in 3.1 section, we use the random forest algorithm in Weka to cross training. Meanwhile, we adjust the parameter computeAttributeImportance to true, so the result of the importance of each feature will be outputted.

Experimental results of importance evaluation of structural characteristics using random forests is shown in Table 2.

Table 2. Importance evaluation of structural features

Feature	Importance
F4(abnormal depth ,"/")	0.23
F3(abnormal length of domain)	0.08
F5(special characters)	0.07
F2(The number of '.' in domain)	0.05
F1(pure IP)	0.04

It can be seen from the upper table that the classification effect of path depth feature is the best, and the other structural features are similar.

4.2 Character Frequency and structure based Methods

When the data set used in the experiment is public data set and uses the 3.1 features, the experiments results of 10-fold cross validation are shown in Table 3.

Table 3. The experiments results on Data Set I

model	TPR	FPR	Prec	Rec	F1-Score	AUC
RF	0.997	0.004	0.997	0.997	0.997	1.000
MLP	0.992	0.009	0.992	0.992	0.992	0.999
Naive Bayes	0.973	0.028	0.973	0.973	0.973	0.993
LR	0.994	0.007	0.994	0.994	0.994	0.999
J48	0.996	0.005	0.996	0.996	0.996	0.996
SVM	0.993	0.008	0.993	0.993	0.993	0.993

When the data set used in the experiment is the data set provided by the paper [13] and uses the 3.1 features, the experiments results of 10-fold cross validation are shown in Table 4.

Table 4. The experiments results on Data Set II

model	TPR	FPR	Prec	Rec	F1-Score	AUC
RF	0.930	0.079	0.930	0.930	0.930	0.982
MLP	0.890	0.128	0.890	0.890	0.890	0.954
Naive Bayes	0.805	0.199	0.809	0.805	0.806	0.898
LR	0.840	0.165	0.842	0.840	0.840	0.907
J48	0.892	0.119	0.892	0.892	0.892	0.880
SVM	0.876	0.137	0.876	0.876	0.876	0.870

From our results we conclude that these classifiers can achieve very high classification precision, with low false false-positive rates. We found that most classification algorithms run very fast at classifying the URLs, with the exception of the SVM and MLP .The SVM and MLP was significantly less effective at classification using our feature extraction techniques.

Our results show that the RF and J48 algorithms performed slightly better than other classification algorithms, but on Data Set I the difference is very small, and may not be statistically significant. While on Data Set II the difference is a little big. We also found that the RF on different Data Set obtained the highest precision, the lowest false positive rate, and the largest ROC area, so Random forest is the most suitable model for the current application scenarios among these models.

4.3 Structure and lexical based Methods

When the data set used in the experiment is public data set and uses the five structural features described in 3.1 section and 9 suspicious words obtained by statistical analysis (index, login, includes, content, admin, account, secure, session, submit)[10], the experiments results of 10-fold cross validation are shown in Table 5.

Table 5. The experiments results on Data Set I

model	TPR	FPR	Prec	Rec	F1-Score	AUC
RF	0.995	0.006	0.995	0.995	0.995	0.999
MLP	0.993	0.008	0.993	0.993	0.993	0.999
Naive Bayes	0.977	0.020	0.977	0.977	0.977	0.997
LR	0.993	0.007	0.993	0.993	0.993	0.999
J48	0.994	0.006	0.994	0.994	0.994	0.997
SVM	0.993	0.008	0.993	0.993	0.993	0.993

When the data set used in the experiment is the data set provided by the paper [13] and uses the five structural features in 3.1 section and 9 suspicious words obtained by statistical analysis (source, expire, range, key, signature, sparams, googlevideo, cpn, exe)[10], the experiments results of 10-fold cross validation are shown in Table 6.

Table 6. The experiments results on Data Set II

model	TPR	FPR	Prec	Rec	F1-Score	AUC
RF	0.880	0.136	0.880	0.880	0.880	0.950
MLP	0.860	0.162	0.859	0.860	0.859	0.933
Naive Bayes	0.776	0.329	0.818	0.776	0.755	0.899
LR	0.848	0.184	0.848	0.848	0.847	0.920
J48	0.881	0.137	0.881	0.881	0.881	0.946
SVM	0.865	0.148	0.865	0.865	0.865	0.859

From our results we also conclude that the RF with different Data Set and different features obtain the highest precision, the lowest false positive rate, and the largest ROC area, so Random forest is the most suitable model for the current application scenarios among these models. We also find that the results based on structural and lexical features are generally worse than those based on character frequency and structural features.

4.4 Other Methods

The authors of the paper [13] used different features and algorithms on data set II and results are as follows in table 7:

From the above results, we concluded that the effect of using only lexical features is much worse than the effect of the combination of features. Comparing our results with the above results, we found that our method based on character frequency and structural features can achieve similar result with the method based on TF-IDF or N-Gram or lex+2gram.

Table 7. The experiments results on Data Set II

model	Feature	Prec	Rec	F1-score	Accu	AUC
SVC	lex	0.79	0.65	0.72	0.79	0.89
LR	lex	0.80	0.66	0.72	0.80	0.90
SVC	tfidf	0.95	0.94	0.94	0.95	0.99
LR	tfidf	0.95	0.92	0.93	0.95	0.98
SVC	ngram	0.93	0.94	0.93	0.95	0.98
LR	ngram	0.96	0.94	0.95	0.96	0.99
xgboost	lex+ 2gram	0.95	0.93	0.94	0.95	0.99

4.5 Adjusting some parameters of effective classifier

We also did the experiments of adjusting the RF's numFeatures and MaxIterations. The experiments results were shown in Table 8.

Table 8. The experiments results on Data Set I

numF	MaxIte	TPR	FPR	Prec	Recall	AUC
0	100	0.990	0.013	0.990	0.990	0.999
1	100	0.988	0.015	0.988	0.988	0.999
5	100	0.989	0.014	0.989	0.989	0.999
6	100	0.989	0.014	0.990	0.989	0.999
7	100	0.990	0.013	0.990	0.990	0.999
8	100	0.990	0.013	0.990	0.990	0.999
8	200	0.990	0.013	0.990	0.990	0.999
8	500	0.990	0.013	0.990	0.990	1.000
8	50	0.990	0.013	0.990	0.990	0.999

Through the above result, generally speaking, the adjustment parameter will not have the very big fluctuation to the random forest experiment result. But increasing numFeatures generally improves the performance of the model, because we have more options to consider on each node. However, this is not necessarily true, because it reduces the diversity of individual trees, which is the unique advantage of random forests. However, it is certain that increasing numFeatures will reduce the speed of the algorithm. NumFeatures generally takes into account sqrt (n) or log (n). Increasing MaxIterations can make the mode obtain better result, but will also make the model run more slowly. Generally choose a moderate value, such as 100,200.

5. CONCLUSION

In this paper, we propose that combining statistical analysis of website URLs with machine learning techniques will give a more accurate classification of malicious URLs. Meanwhile, in order to extract the more essential features of malicious URL, We pay more attention to the character frequency features of URL. In experiments, we compare six machine learning algorithms to verify the effectiveness of the Random Forest than that of others. Besides, we verify the effectiveness of the features we extract while comparing with other feature schemes. The results show that the proposed feature schemes is feasible and performs best combined with the Random Forest algorithm. Finally, In order to obtain better results, we adjusted the Random Forest parameters, such as numFeatures and MaxIterations. The parameters of the random forest algorithm are adjusted to the best, such as the numFeatures is sqrt(n) or log(n),the MaxIterations is 100 or 200.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China 2016 under Grant 2016YFB0801302.

7. REFERENCES

[1] Choi, H., Zhu, B.B., Lee,H.: Detecting malicious web links and identifying their attack types. In: Fox, A. (ed.) 2nd USENIX Conference on Web Application Development, WebApps'11, Portland, Oregon, USA, June 15-16, 2011. USENIX Association(2011)

[2] Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: learning to detect malicious web sites from suspicious urls. In: IV, J.F.E., Fogelman-Souli_e, F., Flach, P.A., Zaki, M.J. (eds.) *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1245-1254. ACM (2009)

[3] Zhang Y, Hong J I, Cranor L F. Cantina: a content-based approach to detecting phishing web sites［C］//*16th International World Wide Web Conference*. Banff, Alberta, Canada, 2007: 639-648.

[4] Fu A Y, Liu W Y, Deng X T. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)［J］ *IEEE Transactions on Dependable and Secure Computing*, 2006,3 (4): 301- 311.

[5] Sujata Garera, Niels Provos, Monica Chew,et a1..A Framework for Detection and Measurement of Phishing Attacks[C]. *Proceedings of 2007 ACM Workshop on Recurring Mal-code*. Alexandria, VA, USA,2007:1-8

[6] Kan, M., Thi, H.O.N.: Fast webpage classification using URL features. In: Her-zog,O., Schek, H., Fuhr, N., Chowdhury, A., Teiken, W. (eds.) *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, Bremen, Germany, October 31 - November 5, 2005. pp. 325-326. ACM(2005)

[7] Aldwairi, M., Alsalman, R.: Malurls: A lightweight malicious website classification based on url features. *Journal of Emerging Technologies in Web Intelligence* 4, 128-133 (November 2012)

[8] Liu, Gang, Bite Qiu, and Liu Wenyin. Automatic detection of phishing target from phishing webpage. *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*. Istanbul, Turkey, 2010:4153-4156.

[9] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Learning to detect malicious URLs. ACM TIST, 2(3):30, 2011.

[10] J Cao, D Dong , B Mao , T Wang, Phishing detection method based on URL features. *Journal of Southeast University* (English Edition) , 2013 , 29 (2) :134-138

[11] Rakesh Verma, Keith Dyer.On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers. *CODASPY'15 Proceedings of the 5th ACM Conference on Data and Application Security and Privacy Pages* 111-122. 2015

[12] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. Lexical feature based phishing URL detection using online learning. In AISec, pages 54-60, 2010.

[13] Octavia-Maria Șulea, Liviu P. Dinu , Alexandra Peste. Using NLP Specific Tools for Non-NLP Specific Tasks. A Web Security Application. ICONIP (4), volume 9492 of Lecture Notes in Computer Science, page 631-638. Springer, (2015)