



# Towards the Detection of Malicious URL and Domain Names Using Machine Learning

Nastaran Farhadi Ghalati, Nahid Farhady Ghalaty, José Barata

## ► To cite this version:

Nastaran Farhadi Ghalati, Nahid Farhady Ghalaty, José Barata. Towards the Detection of Malicious URL and Domain Names Using Machine Learning. 11th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Jul 2020, Costa de Caparica, Portugal. pp.109-117, 10.1007/978-3-030-45124-0\_10 . hal-03741567

**HAL Id: hal-03741567**

**<https://inria.hal.science/hal-03741567>**

Submitted on 1 Aug 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Towards the Detection of Malicious URL and Domain Names Using Machine Learning

Nastaran Farhadi Ghalati<sup>1</sup>, Nahid Farhady Ghalaty<sup>2</sup>, and José Barata<sup>1</sup>

<sup>1</sup> Universidade Nova de Lisboa (UNL), CTS-UNINOVA, Lisbon, Portugal

<sup>2</sup> George Mason University, Virginia, USA

[n.ghalati@campus.fct.unl.pt](mailto:n.ghalati@campus.fct.unl.pt), [nfarhady@gmu.edu](mailto:nfarhady@gmu.edu), [jab@uninova.pt](mailto:jab@uninova.pt)

**Abstract.** Malicious Uniform Resource Locator (URL) is an important problem in web search and mining. Malicious URLs host unsolicited content (spam, phishing, drive-by downloads, etc.) and try to lure uneducated users into clicking in such links or downloading malware which will result in critical data exfiltration. Traditional techniques in detecting such URLs have been to use blacklists and rule-based methods. The main disadvantage of such problems is that they are not resistant to 0-day attacks, meaning that there will be at least one victim for each URL before the blacklist is created. Other techniques include having sandbox and testing the URLs before clicking on them in the production or main environment. Such methods have two main drawbacks which are the cost of the sandboxing as well as the non-real-time response which is due to the approval process in the test environment. In this paper, we propose a method that exploits semantic features in both domains and URLs as well. The method is adaptive, meaning that the model can dynamically change based on the new feedback received on the 0-day attacks. We extract features from all sections of a URL separately. We then apply three methods of machine learning on three different sets of data. We provide an analysis of features on the most efficient value of N for applying the N-grams to the domain names. The result shows that Random Forest has the highest accuracy of over 96% and at the same time provides more interpretability as well as performance benefits.

**Keywords:** Cyber-Security · URL Classification · Machine Learning.

## 1 Introduction

With the advent of new communication technologies, a huge amount of growth has been observed in the sector of business applications such as e-commerce, healthcare, education, travel, and commute, etc. Many of these applications are critical since their database contains private information and critical information of the customers. Therefore, technique that can be misused to access this information violates the privacy of the customers and may lead to irreparable consequences for the business.

The World Wide Web (WWW) provides massive amounts of information for the users. This information could be benign or malicious. The information is transferred to people by clicking on the Universal Resource Locator (URL). Unfortunately, the advance of technology is also coupled with the advent of new cyber-attacks on such technologies. Such attacks include rogue websites that try to sell counterfeit goods, or

intrigue people to share the sensitive information in exchange for subscriptions or gifts, as well as phishing attacks that install malicious software or malware on the user's device without him/her knowing. These attacks have resulted in billions of dollars lost every year. The techniques to launch such attacks is a long list starting with spam campaigns, pop-ups, spyware and malware[11].

A URL has three main components: 1) the protocol identifier and 2) the IP address or domain name for the resource of the page, 3) The path that specifies a resource in the host. The protocol and the identifier are separated by `://` shown in Figure 1.

It has been shown that 39% of URLs are malicious or compromised [12]. Popular types of attacks using malicious URLs include: Drive-by Download, Phishing and Social Engineering, and Spam [16]. Drive-by download [6] refers to the (unintentional) download of malware upon just visiting a URL. Such attacks are usually According to the RSA Online Fraud Report3 for 2018, "Phishing accounted for 48 percent of all cyber-attacks observed by RSA. Canada, the United States, India and Brazil were the countries most targeted by phishing."[1]. Such attacks are carried out by injecting vulnerabilities and malicious code using Javascript.

In this paper, we first go over our motivation on solving the problem of phishing attacks and malicious URLs and how it affects the industrial, electrical and computing community. Then, we go over the previous works and the advantages and disadvantages of several methods. In section 4, we go over the proposed method, the details of the model and features. In section 5, we provide the results of the prediction model and an overview of the dataset and compare our achievement with the previous works. Finally, we conclude the paper with proposals on how to continue this effort and future works.



**Fig. 1.** Example of a URL

## 2 Contribution to Industrial and Service

DoCEIS has been focused on representation of innovative technologies such as Industry 4.0, manufacturing systems, and Internet of Things. Cyber-security is one of the main concerns around the new industrial technologies. With the rapid growth of the digital world and the expansion of the Internet, along with the improvement of technological advances in business and industrial systems, business users inevitably involve taking serious risks. Detection and prevention of such threats will improve the security of these technologies which results in life improvement.

Phishing attacks is the practice of sending fraudulent communications to business and personal services to look like they are coming from a reputable source. Symantec has reported that the average user receives 16 phishing attempt emails per month [2]. According to Wombat Security 76% of businesses experienced a phishing attack in 2018

[3]. Also, Verizon's 2018 Data Breach Investigation Report showed that 93% of security incidents are the result of phishing. Phishing attempts are only successful if they are clicked on by the user, otherwise they are harmless. Therefore, efforts on development of techniques to avoid exposing the users to these threats is valuable to industry. In this paper, we work on the development of a fast technique for phishing attack detection and prevention.

### 3 Related Work

Over the years, researchers have worked on several methods for detection of malicious URLs. One of the traditional methods that has been deployed by many anti-viruses is Blacklist method. In this method, a list of previously known URLs that have been confirmed is stored and maintained in a database. The database often becomes compiled by several toolbars such as PhishBook [8], and PhishTank [15]. The method is very fast since it is only querying against a database, however, because the new technology has made the attackers capable of only hosting malicious domains for only a couple of hours, this method is no longer as effective [19]. Also, there are techniques that can be used for obfuscation of the URLs so there are several equivalents even for one malicious URL. There are also methods that are being used to shorten the URL to make it look legit such as [5].

Since Blacklist methods cannot be trusted with the newly generated URLs, there are several techniques such as Heuristic methods that look at the signature generated by the behavior of specific attacks. In these methods, the tool looks for monitoring the behavior of the URL such as the number of redirects it makes, or the unusual process creation. These behaviors are the so-called signatures of the URL [20][17]. The attackers launched several attacks that exploited obfuscation of the signature. As a result, the signatures are not detectable in such attacks.

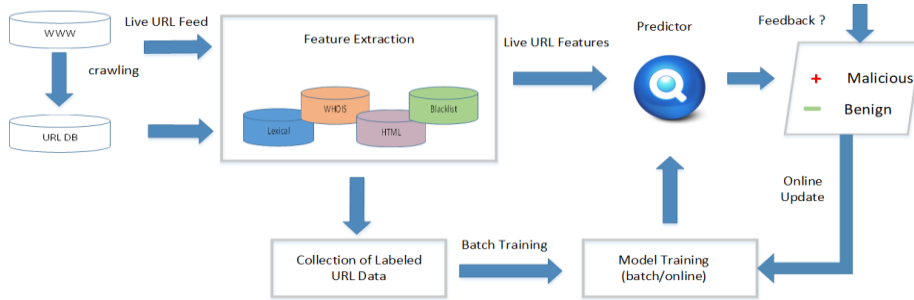
To overcome the above methods for breaking the blacklist and signature-based methods, researchers have relied on machine learning techniques. Machine learning methods are based on extracting features from a set of training dataset and performing statistical analysis to be able to predict and classify the URLs into benign and malicious. There are two types of features that can be extracted from the URLs, the dynamic features and the static features. The static features include lexical attributes of the URL, the host and sometimes the JavaScript and other content of the host. The dynamic features require the execution and clicking on the URL in a sandboxing environment and monitoring the live behavior of the URLs. It is expected that the behavior of the URL is considered anomalous compared to the behavior of benign URLs [21][4]. One of the drawbacks of the second category is that the decision-making process takes longer compared to the static feature detection methods.

## 4 Proposed Method

In this section, the problem formulation is explained. Our system architecture has three main parts as shown in the list below. In the following sections, we go over the details of the proposed architecture.

- The feature extraction
- The training module
- The prediction component

The framework for this model has been shown in Figure 2. The approach schematically presents the procedure of detecting malicious URLs. As it is shown in Figure 2, the URLs should be analyzed in terms of the available information and their corresponding websites. To achieve these, two different methods have been proposed naming static and dynamic. The static method only uses the information and executing of the analyzed URLs is not necessary. This advantage makes the static method safer compared to the dynamic model. Further extensions were performed by researchers to improve the static method. They employed machine learning techniques to increase the accuracy of the response of the static based features. Despite the advantages of the static procedure, in real world applications, they usually suffer from some major limitations. To overcome these limitations some online learning methods have been considered by researchers.



**Fig. 2.** Machine learning for URL Detection Framework

### 4.1 Problem Statement

The problem of this paper is explained as a binary classification problem. The classes for prediction are either malicious URL versus Benign URL. Consider a set of  $N$  URLs as in

$$\{(u_0, y_0), (u_1, y_1), \dots, (u_n, y_n)\} \quad (1)$$

Where  $u_n$  is defined as the URL string number  $n$  and  $y_n$  is the corresponding label for  $u_n$ , in which  $y_n = +1$  depicts malicious and  $y_n = -1$  depicts benign. The first step in this framework is to extract feature  $u_n \rightarrow x_n$  where  $x_n \in R^d$  presents a numerical feature of the URL. The next step is to train prediction function  $f: R^d \rightarrow R$  which is predicting the class

for URL  $x$  which is represented by  $y'$ . The subtraction of the predicted value from the actual value,  $y - y'$ , is defined as the mistake in the prediction and the goal of the machine learning algorithm is to minimize the amount of total mistake for all predictions. The first part of the algorithm, feature extraction, is mostly based on available knowledge of the URL, while the second part is mostly achieved by trying different models in training the machine learning model.

## 4.2 Feature Extraction

The first step in the feature extraction is to split a URL into three sections:

1. Protocol
2. Domain
3. Path

As a result, the first feature extracted is whether the string format of the domain is a URL or an IP address. The reason that we divided our database into three sections is that the features extracted from a domain will be different from a URL or from an IP address. The extracted features from URL such as bag of word or n-grams will show only 0 for domains or IP addresses and this will result in an unbalanced bias in training the model.

Here is the list of different features extracted:

**Blacklist features:** As mentioned, blacklist features cannot be trusted with the new URLs generated by the attackers. But it has been shown that most of the attackers make very small changes to the URL string [9]. So, we applied a method of fuzzy matching using the *fuzzybuzzy* and *diffli* library of python and used the resulting number of similarity matches as a feature. This number could be any value from 0 to 100 as a percentage. The database used for comparison is PhishTank [15].

**Lexical features:** These features are mostly extracted from the URL string. The features we extracted from the strings are the length of the string, number of dots, number of characters such as /, =, \*, ?, ., , number of sub-domains, Shannon entropy of the URL string [13]. The list of features will be shown in detail below.

- N-grams : Previous research has shown that the obfuscation of the URLs will cause bias in the distribution of the characters. N-grams are extracted from the URL characters. The value of  $N$  can be anything between 1 to 10. For example, the first three bigrams of *google.com* are go, oo, og. This is much stronger than the bag of words method used in [10], since it captures punctuation as well.

**Host-based features:** The host-based features are extracted from the URLs. They help with knowing the location, identity, management style and properties of the URL [14]. Phishers tend to use short-term services, WHOIS information [7], location and domain name properties. This information can be extracted using **PyWhois** library. There are several other host-based features that could be extracted; however, the disadvantage of host-based features is that they need to be extracted based on a response by running the algorithm live (dynamic features) and we need a sandboxing environment for that.

### 4.3 Training Model

The unpredictable nature of this problem, and the fact that attackers can generate any type of string, only by changing a couple of characters in a legit URL has made us choose a classification algorithm that is more resistant against noise. Also, malicious URL and phishing detection are the main tasks of CSOC (Cyber Security Operations Center) in industrial firms and companies. Machine learning is a tool that can be used to assist CSOC analysts in their decision-making process. As a result, choosing an algorithm that makes interpretable results is key in helping CSOC make the final decision easier. These circumstances made us try out the results of Random Forest Classification because we can extract the importance of features from the output of this algorithm and present it as part of the results. Random Forest is a good choice in this problem because the training is on multiple clusters of the data and this will lead to reduction invariance.

## 5 Experimental Result

In this section, we explain what datasets we have used for our experiments and also go over the evaluation method and result of training the model.

### 5.1 Dataset

The experiment for this work has been collected using three different sources of data. The first source is the Ebbu2017 Phishing Dataset. Due to the lack of public malicious URL dataset, they have developed their own scripts to query for the Yandex Search engine to create a balanced dataset [18]. The dataset contains about 74k URLs, out of which 36k are legitimate and 37k are phishing. The second source of dataset is the DMOZ dataset and the Alexa.com dataset for providing benign data sources. We also used the prepared malicious URL dataset from [14].

### 5.2 Results

The results of the Random Forest have been compared against other machine learning algorithms will be discussed in this section.

The analysis of the results is first on the exploration of the percentage of overlapping and reusing in N-grams. The percentages of the N-gram overlaps and reuse is shown Figure 3. According to the results, as the value of the  $N$  increases, the number of overlapping decreases. Also, as the overlap decreases between the data sources, the accuracy increases because the features will be more useful for distinguishing the benign vs the malicious URL.

Next, we test the accuracy of our model against other models. For the sake of a fair comparison with previous works, we use our extracted features with other models. Table 1 shows the results. The ROC (Receiver Operating Characteristic) curve is shown in Figure 4. As shown in this table, the results for the Random Forest outweighs the results of the two other approaches. On top of the simplicity and explain ability of the model. it also has higher accuracy.



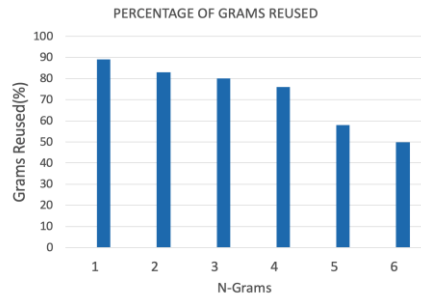


Fig. 3. Reused N-grams

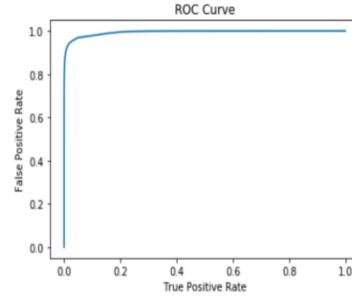


Fig. 4. Roc curve

Table 1. Comparison of classifiers (%)

Algorithm	Accuracy	F-Score	Recall	Precision
Random Forest	96.786	94.453	95.8	90.1
Linear Regression	92.324	93.183	95.58	90.90
Naive Bayes	87.634	89.231	82.23	85.57

## 6 Conclusion and Future Work

This paper proposes a static lexical feature-based Random Forest Classification model to classify malicious vs benign URLs. The results extracted from this experiment show that lexical features can be used for a high-performance and light-weight method for fast generation of URL labels. Our study also shows the result of feature importance on the N-grams lexical feature. Based on our analysis, it will be more useful to use a higher number for  $N$  as a feature because it has higher overlap and will result in better accuracy. The future steps for this work are to provide more analysis on the effects of separation of domain names and URLs and also observe the effects of speed vs the number of features extracted.

**Acknowledgments.** This work was supported in part by the FCT/MCTES (UNINOVA-CTS funding UID/EEA/00066/2019), UIDB/00066/2020 (CTS – Center of Technology and Systems), and the FCT/MCTES project CESME - collaborative and Evolvable Smart Manufacturing Ecosystem, funding PRDC/EEI-AUT/32410/2017.

## References

1. RSA QUARTERLY FRAUD REPORT, Volume 1, Issue 3 Q3 2018.

2. Nahorney, O. C. H. L. B., O’Gorman, D. O. B. B., Paul, J. P. P. S. W., Cleary, W. C. W. G., & Corpin, M. Internet security threat report. Technical Report 23, Symantec Corporation.(2018).
3. State of the Phish™ Report, wombat security technologies(2018).
4. Canfora, G., Medvet, E., Mercaldo, F., Visaggio, C.A.: Detection of malicious web pages using system calls sequences. In: International Conference on Availability, Reliability, and Security. pp. 226-238. Springer (2014)
5. Chhabra, S., Aggarwal, A., Benevenuto, F., Kumaraguru, P.: Phi. shocial: the phishing landscape through short urls. In: Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. pp. 92-101. ACM (2011)
6. Cova, M., Kruegel, C., Vigna, G.: Detection and analysis of drive-by-download attacks and malicious javascript code. In: Proceedings of the 19th international conference on World wide web. pp. 281-290. ACM (2010)
7. Daigle, L. (2004) "WHOIS Protocol Specification", RFC 3912
8. Fahmy, H.M., Ghoneim, S.A.: Phishblock: A hybrid anti-phishing tool. In: 2011 International Conference on Communications, Computing and Control Applications.pp. 1-5. (2011)
9. Felegyhazi, M., Kreibich, C., Paxson, V.: On the potential of proactive domain blacklisting. LEET 10, 6-6 (2010)
10. Gyawali, B., Solorio, T., Montes-y Gómez, M., Wardman, B., Warner, G.: Evaluating a semisupervised approach to phishing URL identification in a realistic scenario. In: Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. pp. 176-183. ACM (2011)
11. Hong, Jason. "The state of phishing attacks." *Communications of the ACM* 55.1.74-81(2012).
12. Liang, B., Huang, J., Liu, F., Wang, D., Dong, D., Liang, Z.: Malicious web pages detection based on abnormal visibility recognition. In: 2009 International Conference on E-Business and Information System Security. pp. 1-5. IEEE (2009)
13. Lin, J.: Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory* 37(1), 145-151 (1991)
14. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 681-688. ACM (2009)
15. OpenDNS, L.: Phishtank: An anti-phishing site. Online: <https://www.phishtank.com> (2016)
16. Patil, D.R., Patil, J.: Survey on malicious web pages detection techniques. *International Journal of u-and e-Service, Science and Technology* 8(5), 195-206 (2015)
17. Rieck, K., Krueger, T., Dewald, A.: Cujo: efficient detection and prevention of drive-by-download attacks. In: Proceedings of the 26th Annual Computer Security Applications Conference. pp. 31-39. ACM (2010)
18. Sahingoz, O.K., Buber, E., Demir, O., Diri, B.: Machine learning-based phishing detection from URLs. *Expert Systems with Applications* 117, 345-357 (2019)
19. Sheng, S., Wardman, B., Warner, G., Cranor, L.F., Hong, J., Zhang, C.: An empirical analysis of phishing blacklists. In: 6th Conf. on Email and Anti-Spam(CEAS). California, USA (2009)
20. Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., Ohsita, Y., Murata, M.: Malicious url sequence detection using event denoising convolutional neural network. In: 2017 IEEE International Conference on Communications. pp. 1-7. (2017)
21. Tao, Y.: Suspicious URL and device detection by log mining. Ph.D. thesis, Applied Sciences: School of Computing Science (2014)