Malicious URL Detection Using Machine Learning

Moulik Bhardwaj
NIT Hamirpur
Email: 185506@nith.ac.in

Karan Bhardwaj
NIT Hamirpur
Email: 185544@nith.ac.in

Harsh Sheth
NIT Hamirpur
185560@nith.ac.in

***Abstract*—With the increase in internet usage, internet security has become a major challenge . Different malicious URLs pull out different malicious software and attempt to capture user information. This paper suggests usage of host-based features and lexical features of related URLs to better improve the performance of classifiers for finding malicious websites. Random forest models and Gradient Boosting algorithms are used to create a URL classifier using the URL series attributes as features. High accuracy achieved by random forest as 99%. Results show that you can identify malicious websites based on URLs individually and classifying them as spam URLs without relying on page content will result in an important resource savings and secure user browsing information.**

1. Introduction

The importance of the World Wide Web (WWW) has attracted increasing attention because of growth and promotion of social media, online banking, and e-commerce. While a new development in communication technology promotes new e-commerce opportunities, creates new opportunities for attackers as well. Today, on the Internet, millions of such sites are commonly referred to as malicious websites. It was noted that technological advances had led to some strategies to attack again scam users like spam SMS on social media, online gambling, phishing scams, financial fraud, fraud winning a prize, and buying a fake TV.

In recent years, many have attacked methods are used for distributing endangered and fishing URLs, as well as the Same Cruel Service Location addresses (URLs) are the most common way criminals use malicious activity. Common types of attacks using malicious URLs can be categorized into Spam, Drive Downloads, Social Engineering, and Crime Theft of sensitive information. Spam is called unsolicited mail compulsory advertising or phishing scams, which we do not request and do not want to receive. These the attack caused extensive damage. Download of malware when visiting the URL is called Drive-by download. Finally, Social Media Attack and Crime Theft information directs users to disclose sensitive and confidential information by acting as real web pages. The invaders made famous copies webpages used by users such as Facebook and Google and endangering victims' computers by setting them up various pieces of malicious code in the HTML code of the modified website.

In addition, common use of smartphones that promote the proliferation of mobile identity theft and Quick Response (QR) activities, especially to trick adults into inserting fake URLs into QR codes. The black side of the internet has it has attracted growing attention and has hurt the world. Internet security software can't always detect malicious software on malicious websites and download. However, it can prevent do not get them early. Detection of incorrect URLs is not enough it is still being treated and causes great losses year after year. In the fourth quarter of 2019, there were more than 162,000 The different URLs for stealing sensitive information have been found worldwide.

In our research, machine learning algorithms are used to classify URLs based on the features and behaviors of URLs. The features are extracted from static and dynamic behaviors of URLs and are new to the literature. Those newly proposed features are the main contribution of the research. Machine learning algorithms are a part of the whole malicious URL detection system. Two supervised machine learning algorithms are used,Random forest(RF) and Gradient Boosting(GB).

mds
August 26, 2015

2. Background

2.1. Litrature Work

In "Malicious URL Detection using Machine learning: A survey" the authors conducted a systematic survey that discussed existing methods to improve the retrieval of malicious URLs.

In "Fake website detection using regression" authors implemented Logistic Regression Algorithm to detect phishing websites based on some parameters like URL and domain identity.

In "Malicious URL Detection based on Machine learning " Cho DO et al.,2020 proposed a machine learning algorithm for URL detection which is based on URL features.

In "Empirical Study on Malicious URL Detection using Machine Learning" authors implemented Random forests, Support Vector Machine and Naïve Bayes on training dataset to identify better approach of detection system.

2.2. Signature based Malicious URL Detection

Studies on malicious URL detection using the signature sets had been investigated and applied long time ago. Most of these studies often use lists of known malicious URLs. Whenever a new URL is accessed, a database query is executed. If the URL is blacklisted, it is considered as malicious, and then, a warning will be generated; otherwise URLs will be considered as safe. The main disadvantage of this approach is that it will be very difficult to detect new malicious URLs that are not in the given list.

2.3. Machine Learning based Malicious URL Detection

There are three types of machine learning algorithms that can be applied on malicious URL detection methods, including supervised learning, unsupervised learning, and semisupervised learning. And the detection methods are based on URL behaviors.

In this project, a number of malicious URL systems based on machine learning algorithms have been investigated. Those machine learing algorithms include SVM, Logistic Regression, Decision Trees, Ensembles. In this paper, the two algorithms, RF and GB, are used. The accuracy of these two algorithms with different parameters setups will be presented in the experimental results.

The behaviors and characteristics of URLs can be divided into two main groups, static and dynamic. In their studies authors presented methods of analyzing and extracting static behavior of URLs, including Lexical, Content, Host, and Popularity-based. The machine learning algorithms used in these studies are Online Learning algorithms and SVM. Malicious URL detection using dynamic actions of URLs is presented in . In this paper, URL attributes are extracted based on both static and dynamic behaviors. Some attribute groups are investigated, including Character and semantic groups; Abnormal group in websites and Host-based group; Correlated group.

2.4. Malicious URL Detection Tools

- **URL Void:**
  URL Void is a URL checking program using multiple engines and blacklists of domains. Some examples of URL Void are Google SafeBrowsing, Norton SafeWeb and MyWOT. The advantage of the Void URL tool is its compatibility with many different browsers as well as it can support many other testing services. The main disadvantage of the Void URL tool is that the malicious URL detection process relies heavily on a given set of signatures.
- **UnMask Parasites:**
  Unmask Parasites is a URL testing tool by downloading provided links, parsing Hypertext Markup Language (HTML) codes, especially external links, iframes and JavaScript. The advantage of this tool is that it can detect iframe fast and accurately. However, this tool is only useful if the user has suspected something strange happening on their sites.
- **Dr.Web Anti-Virus Link Checker:**
  Dr.Web Anti-Virus Link Checker is an add-on for Chrome, Firefox, Opera, and IE to automatically find and scan malicious content on a download link on all social networking links such as Facebook, Vk.com, Google+.
- **Comodo Site Inspector:**
  This is a malware and security hole detection tool. This helps users check URLs or enables webmasters to set up daily checks by downloading all the specified sites. and run them in a sandbox browser environment.
- **Other tools:**
  Some Among aforementioned typical tools, there are some other URL checking tools, such as UnShorten.it, VirusTotal, Norton Safe Web, SiteAdvisor (by McAfee), Sucuri, Browser Defender, Online Link Scan, and Google Safe Browsing Diagnostic.
  From the analysis and evaluation of malicious URL detection tools presented above, it is found that the majority of current malicious URL detection tools are signature-based URL detection systems. Therefore, the effectiveness of these tools is limited.

3. Dataset

The dataset was taken from here [link of data]. The dataset contains extracted attributes from websites that can be used for Classification of webpages as malicious or benign. The dataset also includes raw page content including JavaScript code that can be used as unstructured data in Deep Learning or for extracting further attributes. The data has been collected by crawling the Internet using MalCrawler . The labels have been verified using the Google Safe Browsing API. Attributes have been selected based on their relevance . The details of dataset attributes is as given below:

- url - The URL of the webpage.
- ip add - IP Address of the webpage.
- geo loc - The geographic location where the webpage is hosted.
- url len - The length of URL.
- js len - Length of JavaScript code on the webpage.
- js obf len - Length of obfuscated JavaScript code.
- tld - The Top Level Domain of the webpage.
- who is - Whether the WHO IS domain information is compete or not.
- https - Whether the site uses https or http.
- content - The raw webpage content including JavaScript code.
- label - The class label for benign or malicious webpage.
- hopcount- Number of hops taken by a url to reach at its final destination.

4. Data Analysis

After cleaning of data, we need to do Exploratory Data Analysis of our dataset, which mainly includes checking for co-relation among parameters of the dataset .

4.1. On the Basis of Different Parameters

1) **On the basis of label**
Here we have analyzed the data on the basis of the Label parameter, i.e. whether the given url is Malicious or Benign. As can be seen from the

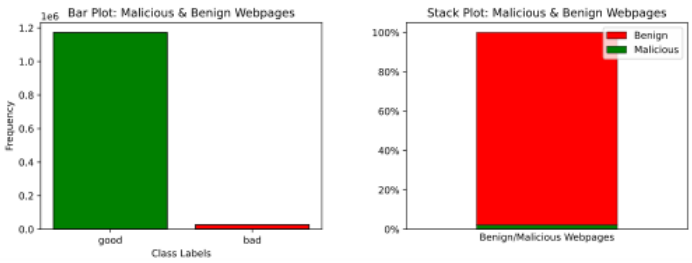

Figure 1. Analysis on basis of label

visualisations this dataset has significant class imbalance. Hence, during any machine learning process, adequate measures will have to be undertaken to handle or compensate this imbalance in order to get accurate results.

2) **On the basis of Attributes**
There are various parameters in the dataset, that can have a boolean value of 0 or 1. Below, we have done some analysis on them.

3) **on basis of scheme**
Scheme here means whether the url prefix is 'http' or 'https'. As seen from visualization, we can see

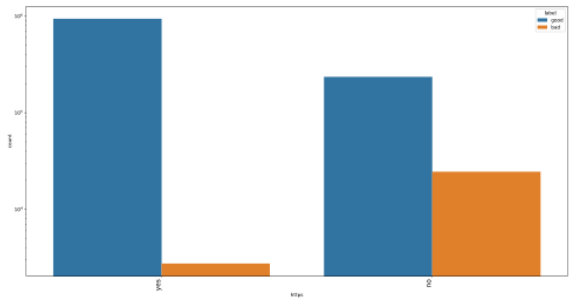

Figure 2. Univariate analysis of HTTPS attribute

that much higher amount of websites using HTTPS protocol are benign. In other words, The sites that dont provide HTTPS protocol are more likely to be malicious than the sites that provide https protocol.

4) **On basis of WHO-IS Information**
The information we are seeking from this attribute is whether the website has registered WHO-IS information or not. As seen from visualizations, we

can see much higher number of malicious URLs have incomplete WHO-IS information.

5) **On basis of Geographic Location**
We have looked up on the geographic distribution of malicious and benign urls.

6) **On basis of TLD**
A top-level domain (TLD) is one of the domains at the highest level in the hierarchical Domain Name System of the Internet after the root domain. From the visualizations, we can see that maximum

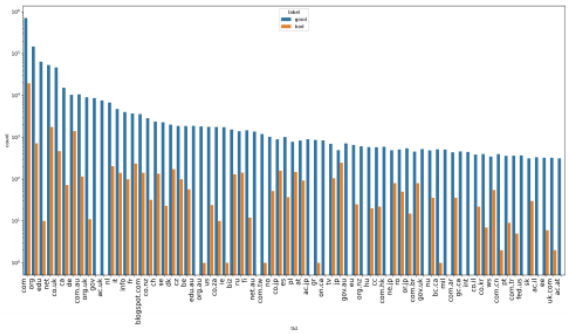

Figure 3. TLD-wise Analysis

number of malicious as well as benign domains are uses .com as TLD.

7) **On basis of Country**
Here, we have visualized the distribution of benign and malicious URLs on the basis of country. Here,

we can see that US has the highest proportion of Malicious sites to benign sites.

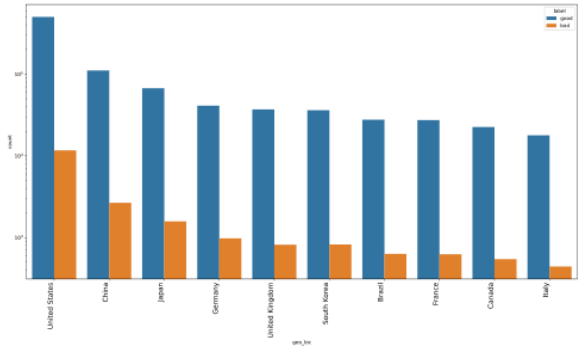

Figure 4. Country wise analysis(Top 10 countries)

8) **On basis of URL Lengths**
Here, we have visualised the distribution of malicious and benign URLS on the basis of it's URL length Here, we can see that the malicious URLs

tends to have higher URL length by a small margin

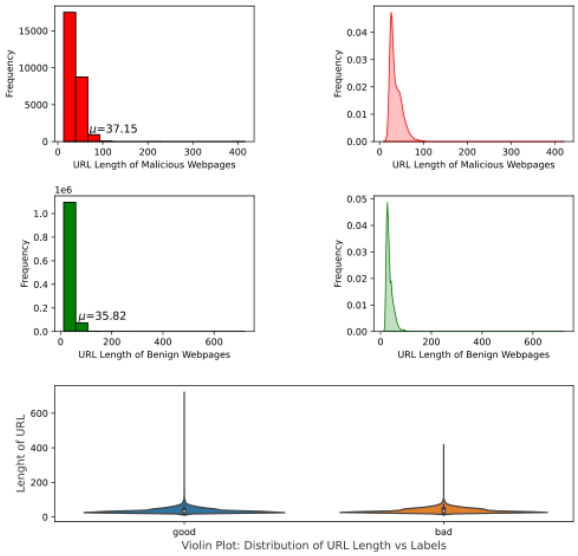

Figure 5. Analysis on the basis of URL length

9) **On basis of JS Length** Here, we have visualised the distribution of malicious and benign URLS on the basis of it's length of normal Javascirpt Code Here, we can see that the malicious URLs tends to have more javascript content by a huge margin.

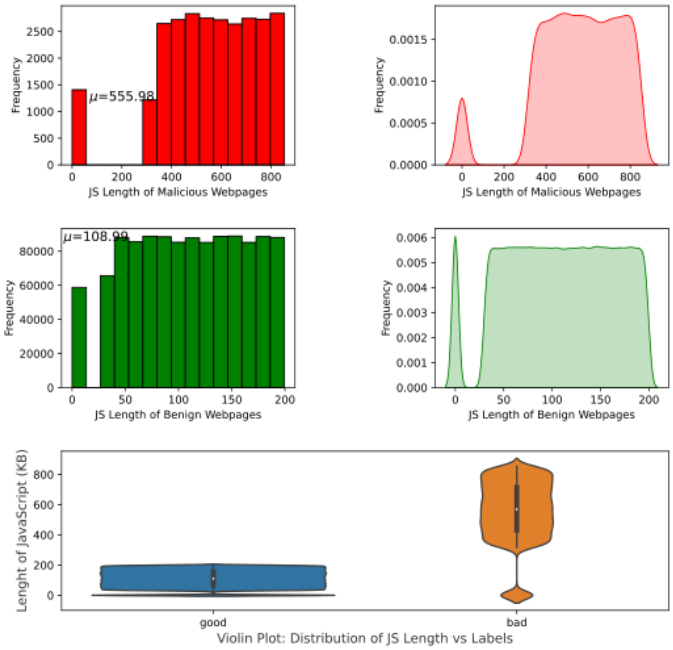

Figure 6. Analysis on the basis of length of Javascript Code

10) **On basis of Obfuscated JS Length** obfuscation is the deliberate act of creating source or machine code that is difficult for humans to understand.[6] Here, we have visualised the distribution of malicious and benign URLS on the basis of it's length of obfuscated Javascirpt Code Here, we can see that the benign URLs tends to have negligible obfuscated JS content

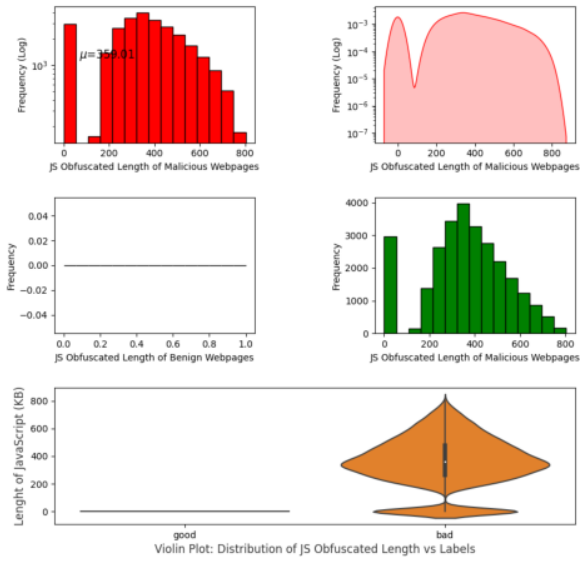

Figure 7. Analysis on the basis of length of Obfuscated Javascript Code

11) **On basis of content length**
Here, we have visualized the distribution of malicious and bening URLs on the basis of the length of HTML content of website. Here, we can see

that Benign Web Pages tends to have higher content length than malicious web pages

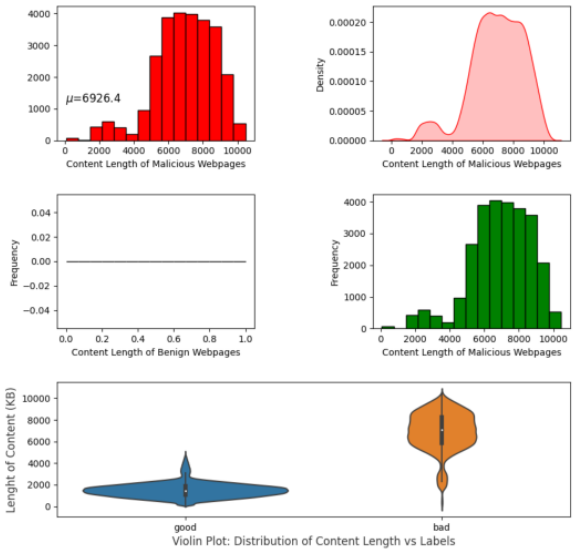

Figure 8. Simulation results for the network.

12) **On basis of hop count**

Hop count refers to the number of network devices through which data passes from source to destination. The notion behind this is that a person running malicious site would try to reroute the connection to make it difficult to track the site.

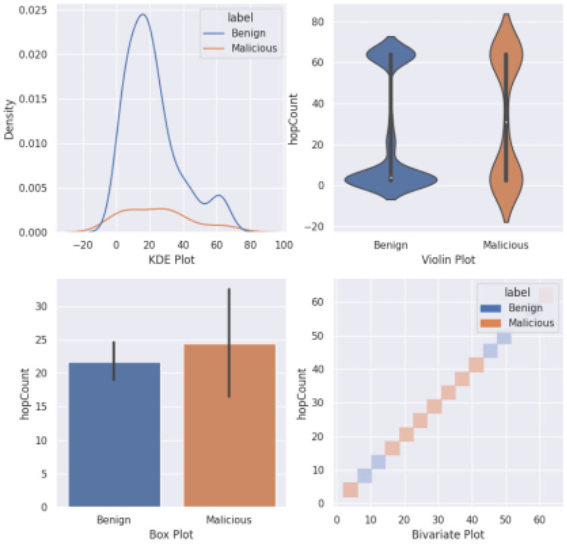

Figure 9. Simulation results for the network.

5. Methods

5.1. Random Forest

Random Forest is a supervised learning algorithm. The forest it builds is an ensemble of decision trees, usually trained with the bagging method. A common idea of the bagging method is that a combination of learning models enhances the overall effect. Simply put: a random forest forms a lot of decision trees and groups them together to get the most accurate and stable prediction.

Algorithm

Input: Number of classifier c, Training dataset X
for i to c do
Random sampling Xi with replacement from X
Build full decision tree classifier using Xi
Return all classifiers

5.2. Gradient Boosting

Gradient Boosting is a type of machine learning boosting. It is based on the assumption that the next best model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error:

•If a small change in the prediction of the case results in a significant reduction in error, the next direct result of the case is a higher value. Predictions from the nearest new target model will minimize error.

•If a slight change in the prediction of the case does not cause a change in error, the next direct result of the case is zero. Changing this prediction does not reduce the error.

5.3. Naïve Bayes:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

•Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

•Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem

6. Results

Classifier()	Accuracy	LogLoss	F1 Score	Precision	Recall
Voting	0.99900	0.03453	0.99948	0.99897	1.0
Random Forest	0.99900	0.03453	0.99948	0.99897	1.0
Gradient Boost	0.99966	0.01151	0.99982	0.99965	1.0
AdaBoost	0.99900	0.03453	0.99948	0.99931	0.99965
SVC	0.99700	0.10361	0.99846	0.99693	1.0
GausianNB	0.99900	0.03453	0.99948	0.99897	1.0

7. Conclusion

With the advent of computer and system technology, people are exchanging information online that attracts people because of the simplicity of the services they provide on a daily basis and beyond, they do a lot other things related to daily life. During these processes, users gain intelligence and valuable information such as descriptive words and passwords. Many network applications find their users as well. The rapid growth of web pages and applications has made it a major target for attackers. Today, the number of malicious websites has increased dramatically. Cruel behavior of loyal or malicious users threaten network applications. Inexperienced users become a the only victim by visiting these dangerous pages. Attackers can easily exploit a web site uploading or embedding malicious code into a web page instead of distributing malicious software.

According to at the Google Research Center, more than 10of malicious web pages is very important to protect web site users from this threats. In this case, finding that user-targeted web pages are being used for malicious behavior it is very important that the institution and individual users overcome this situation to a minimum damage. Recent years have seen the discovery of a malicious URL that plays a key role in cybersecurity applications. Malicious URL has become a major threat to internet security. Without any questions, CPS can be considered an important step in the development of data access and processing services available on the internet.

Researchers have learned to come up with some interesting and concise results. Reading Machine- The methods used are most commonly used in the detection of malicious URLs. In this study, data for Web pages used for phishing scams distributed at the UCI database are used. This study analyzed performance of machine learning algorithms to detect non-computer programming. We used the Random Forest again Gradient Boosting machine learning algorithms for malicious URLs. Test results of the proposed method demonstrates the effectiveness of the Machine Learning Model (Random Forest) in processing large databases and predicting a website as dangerous or dangerous is very good impressive (98.6%). This shows that we can quickly build useful and reliable machine learning models of malicious URL detection models.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] Catak, Ferhat Ozgur S¸ahinba¸s, Kevser Dortkardes, Volkan. (2020). Malicious URL Detection Using Machine Learning. 10.4018/978-1-7998-5101-1.ch008.

[3] SINGH, AMIT KUMAR (2020), "Dataset of Malicious and Benign Webpages", Mendeley Data, V2, doi: 10.17632/gdx3pkwp47.2

[4] Google Safe Browsing API

[5] https://pandas.pydata.org/pandas-docs/stable/user guide/scale.htmlusechunking

[6] Postel, Jon (March 1994). "Domain Name System Structure and Delegation". Request for Comments. Network Working Group. Retrieved 7 February 2011. "This memo provides some information on the structure of the names in the Domain Name System (DNS), specifically the top-level domain names; and on the administration of domains."

[7] https://en.wikipedia.org/wiki/Obfuscation (software)

[8] https://cybersecurityguide.org/resources/phishing/

[9] https://en.wikipedia.org/wiki/Spamming

[10] Catak, Ferhat Ozgur S ahinba s, Kevser Dortkardes, Volkan. (2020). Malicious URL Detection Using Machine Learning. 10.4018/978-1-7998- 5101-1.ch008.

[11] SINGH, AMIT KUMAR (2020), "Dataset of Malicious and Benign

Webpages", Mendeley Data, V2, doi: 10.17632/gdx3pkwp47.2

[12] https://en.wikipedia.org/wiki/Obfuscation (software)

[13] https://www.brainstobytes.com/precision-vs-recall

[14] https://en.wikipedia.org/wiki/Hop$_{(networking)}Hop_{c}ount$

Bannur, S. N., Saul, L. K., Savage, S. (2011). Judging a site by its content: learning

the textual, structural, and visual features of malicious web pages. Proceedings of the

4 th ACM Workshop on Security and Artificial Intelligence. 10.1145/2046684.2046686 [15] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

doi:10.1023/A:1010933404324 [16] Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166.

doi:10.1109/72.279181 PMID:18267787 [17] Cova, M., Kruegel, C., Vigna, G. (2010). Detection and analysis of drive-by- download attacks and malicious JavaScript code. In Proceedings of the 19th international conference on World wide web (WWW'10) (pp. 281-290). Raleigh, NC: Association for Computing Machinery. 10.1145/1772690.1772720

[18] Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics Data Analysis, 38(4),367–378. doi:10.1016/S0167-9473(01)00065-2

[19] Kuyama, M., Kakizaki, R. S. (2016). Method for Detecting a Malicious Domain by Using WHOIS and DNS Features. The Third International Conference on Digital Security and Forensics (DigitalSec2016),74-80

[20] W. Xu, F. Zhang and S. Zhu, "The power of obfuscation techniques in malicious JavaScript code: A measurement study," 2012 7th International Conference on Malicious and Unwanted Software, 2012, pp. 9-16, doi: 10.1109/MALWARE.2012.6461002.

[21] Kim, W., Jeong, O.-R., Kim, C., So, J. (2011). The dark side of the Internet: Attacks, costs and responses. Information Systems, 36(3), 675–705. doi:10.1016/j.is.2010.11.003

[22]https://www.geeksforgeeks.org/metrics-for-machine-learning-model/

[23] https://docs.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs- machine-learning