ISSN No:-2456-2165

Comparative Analysis of Algorithms Detecting Malicious URL's.

Yashaswini J[1], Garima Varshney[2], Rupinder kaur sandhu[3], M Nagaraju[4]
School of computer science and engineering, VIT University, Vellore, India

Abstract:- With the advent of technology, any piece of information can be retrieved from internet by giving an appropriate URL. But this can also be the source of criminal activities causing the URL to be malicious. Such websites contain unwanted contents like phishing sites, spam-advertised products, dangerous "drive-by" harness that infect a visitor's system with malware. Using URL features such as length of URL, domain of URL, Presence of Ip Address in Host Name, Presence of Security Sensitive Words in URL and many more which are relying on the fact that users directly deal with URLs to surf the internet and provides a good approach to detect malicious URLs. In this paper we perform comparative analysis of various machine learning algorithms such as logistic regression, decision trees, gradient boosting, Random forest and adaboosting on various performance metrics like their learning approach, accuracy and many more in detecting malicious content of URL's.

Keywords:- *logistic regression, Decision tress, Adaboosting, Gradient boosting, Random forest.*

I. INTRODUCTION

The only technology that has made information retrieval within a blink of an eye is internet. More and more people are staying up connected due to the sole reason of internet. Right from low level business functionalities to high level enterprise, we all are surrounded by the internet being its biggest slaves. But internet too comes with many disadvantages which users might not be aware of. Let us take a simple example of shopping, where users surf the online shopping website to buy a certain product and are being tracked by the phishing attackers. Phishing is nothing but an act where people are attracted by the fraud and fake websites in order to extract the confidential information which consists of credit card or debit card numbers, passwords. The list of dangerous websites is increasing by 10,000 per day.

URL is considered to be the foremost entry place for most of the fake and fraud websites. Hence detecting malicious URL turns out to be the most important research topic in today's times. We require a handful of solutions to deal with such daily occurring problems. Among the solutions we have blacklisting, but its disadvantage is that we have to do the tedious task of searching the list for presence of the entry. And the list can be very large considering the number of websites on the internet. And the list cannot be kept up to date because of the ever-growing growth of web link each and every hour.

In this paper we refer to a different approach where the URL is detected to be whether benign or malicious on the basis of host-based and lexical features without referring the actual web page content. This approach provides a light weight mechanism for detecting URL as we need not download the entire web page. Also, one of the prominent advantage provided by this method is that ample amount of information is available with the host-based and lexical features which are useful enough for classifying the website. Learning algorithms can be implemented in either online or batch settings. Online algorithms proved to be more efficient as their capability to accustom the classifier is increased with the increase in the number of training examples. The power of online mechanism is measured by its two most important contributions such as first, it can handle huge amount of data and second is they can easily comply with feature distribution which are very useful in classifying the malicious URL. Section 3 shows related work, section 4 gives brief definition of various algorithms named logistic regression, decision trees, random forest, gradient boosting, adabosting. Section 5 shows the comparative analysis of the above machine learning algorithms in terms of their performance in accuracy in detecting the malicious URL. Section 6 shows the result and discussion.

II. LITRATURE REVIEW

Many approaches had been undertaken to fight back the various malicious activities and overcome phishing attacks. Accessing the web page and collecting phishing features is a tedious and time-consuming process. On the other hand there occurred a prime necessity for a highly accurate mechanism for malicious URL detection. Kan and Thi[1] provided one of the earliest machine learning URL classification method which used to train model and analyzing the lexical features without analyzing the page content and extracting each URL's host features. Garera et al.[2] in order to classify malicious URL used logistic regression approach by selecting 18 features and achieved 97.3% accuracy. McGrath and Gupta[3] performed comparative analysis of nonphishing and phishing URL's. Provos et al.[4] extracted content based features and used machine learning algorithm to detect whether the inline frame contains way to malicious websites. Fette et al[5] used machine learning approach and applied statistical methods including the features like email structure and classifying the phishing messages. To further analyses the email content Bergholz[6] added text classification models

in order to achieve higher accuracy that obtained by the fette. As URL is represented "bag of words" kolari et al.[7] determined the spammed page of the URL achieved from blog space. Moshchuk adopting the non-machine learning approach, taking the support of antispyware tools and virtual machine to detect whether the downloaded material from the web contains Trojan. Wang et al[8] used behavioral based features to detect drive-by exploitation which is again a non-machine learning approach. Moshchuk[9] taking the idea of behavioral features developed web proxy Virtual machine for detecting malicious content of the web page.

III. OVERVIEW OF MACHINE LEARNING

Various machine learning algorithms used for detecting malicious content of URL are described briefly in this section.1. logistic regression- It is a form of regression that allows the prediction of discrete variables by a mix of discrete and continuous predictors. Its is used in the cases where the relationship between the predictor and the discrete variable is non- linear. 2. Decision trees- it is made up of decision nodes and terminal leaves which keeps on splitting recursively to make the fine-grained decision over a topic. 3. Gradient Boosting- By gathering models of weak prediction, ensembling it forms a prediction model by optimizing differential loss function. 4. Adaboosting-stands for adaptive boosting which is used to combine with other machine learning algorithms to upgrade the performance of algorithm learning. It boosts the classifier, include the instances which are missed by the earlier classifiers and results better learning of the data. 5. Random forest- at the training time this algorithm forms a congregation of decision trees and calculates mode of classes. It falls under the category of ensemble learning and is suitable for both regression and classification.

IV. METHODOLOGY

Datasets- in this paper, both benign and malicious URL dataset have been taken from Dmoz open directory(https://www.dmoz.org/)phistank.com(https://www.phistank.com/) respectively. We took total of 7000 URL's randomly mixed both benign and malicious.

	URL	L
0	http://secure-data.info/webapps/d1dfd/	1
1	http://10251200.at.et.ua/incorrect_email.html	1
2	http://qat-gps.com/log-in/Secure_Account/Service	1
3	http://www.bostonathenaeum.org/	0
4	http://www.palaisdesthes.com/	0

Table1: URL Label Description

Table 1 indicates that URL is malicious whereas label 0 indicates that the URL is benign. As it is known that host-based and lexical features contain ample amount of information regarding a URL, it is divided into hostname, query and path which are further divided into tokens, token into delimiters.

A. Evaluation metrics

The performance metric considered in this paper is Score. After learning the training data, the learning algorithm forms the model to test the testing data. Score is one of the metric that is used to test whether the test model generated is accurate enough to test the testing data and generate correct results. It is a specification of how accurately it classifies the data.

V. IMPLEMENTATION

In this section we represent implementation of URL classification system. The system is working on the lexical features which includes length of URL, presence of security words in URL, domain length, presence of IP address in the Host name. and many more. Python code of various algorithms are written on Ipyhton Jupiter notebook to evaluate the score of each algorithm and determine which comes out to be best. Code consists of python scripts of fetching malicious as well as benign URL's from the respective .csv files. The following figures shows malicious and benign behavior on each attribute of URL.

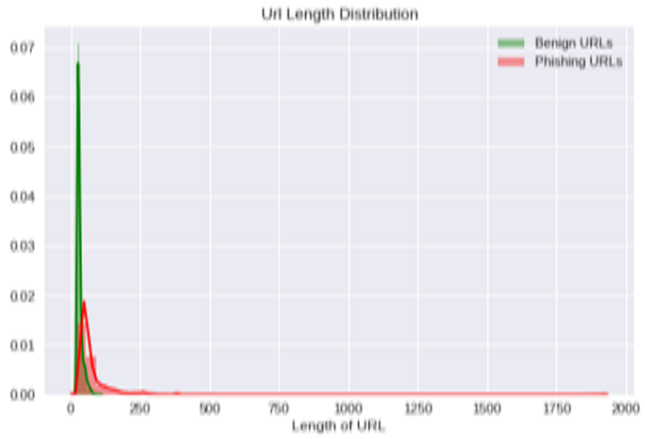

Fig. 1:- URL length Description

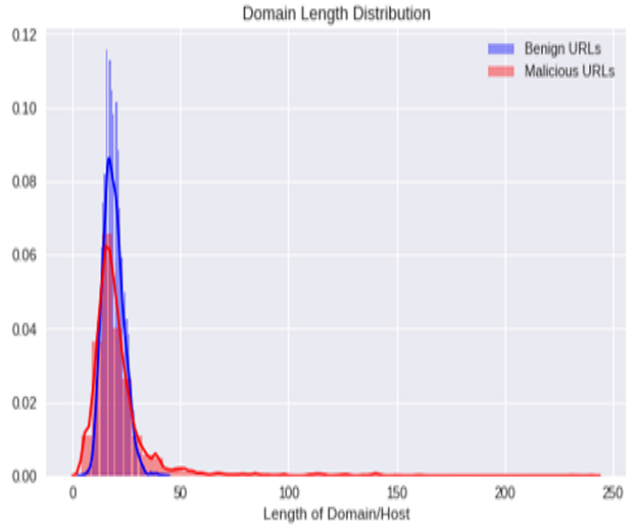

Fig. 2:- Distribution of no of dots in URL

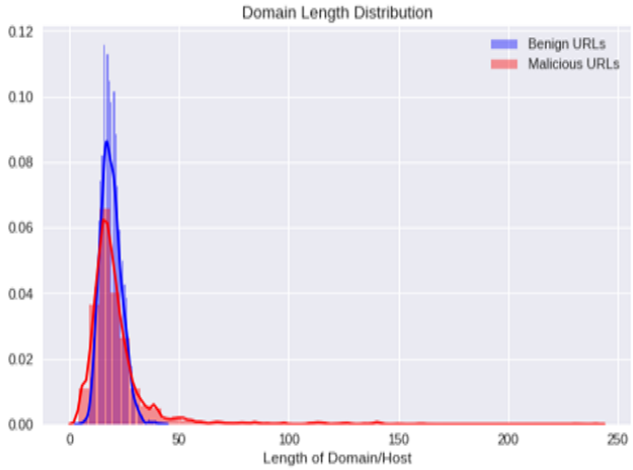

Fig. 3:- Domain Length Distribution

A. Comparative Analysis of the machine learning algorithms.

	Logistic	Decision tree	Gradient Boosting (GB)	Adabossting	Random forest
Learning approach	Supervised	Supervised	Unsupervised	Unsupervised	Unsupervised
Suitable for	Where outcome variable has several values	Best suitable for error prone data, also different output value for a target function	When boosting of learner is required by focusing on difficult observations	When boosting of learner is required by focusing on the misclassified observations.	When the data is missing and many more outliers are present.
Advantage	Linearity in Relationship between the dependent variable and independent variable is not assumed.	We can assume as many number of outcomes as required.	The processing of GB is sequential. Hence it takes longer time	Simple and easy to implement	Good for huge databases and error balancing for unbalanced data.
Disadvantage	Continuous outcomes are not predicted	It results less accuracy when number of decisions increases.	More likely to get affected by overfitting.	Very sensitive to outliers and noisy data.	Overfitting is the main disadvantage.

Table 2: Comparative Analysis

B. Experimental Results.

To measure the performance of each algorithm in determining whether the given URL is malicious or not, Score is taken as a performance metric. Below table is the comparison of the scores evaluated using each algorithm.

Algorithm used	Score
Logistic regression	0.847083926031
Random forest	0.909672830725
Gradient Boosting	0.887624466572
Decision tree	0.906116642959
Adaboost	0.876955903272

Table 3: Experimental Analysis

VI. CONCLUSION AND FUTURE WORK

As internet is one of the highly used resource in today's era, it is also one of the most dangerous place where frauds and fake website can gain access to user's confidential information. By undertaking lexical and host-based features we can determine whether the given URL is malicious or benign. As per the recent requirement, we want a classifier which can adapt the dynamic behavior and determine its type. Hence, we require algorithms that can adapt to new features as criminals too come up with new strategies every day. In this paper, we adopted various approaches by taking supervised and unsupervised learning algorithms for determining the malicious content in URL. Batch algorithms could not fetch efficient results. So, we come up with the conclusion that URL classification problem can be handled online. The algorithms used are efficient in detecting the malicious URL with the incoming new features dynamically.

REFERENCES

[1]. Kan, M.-Y. And Thi, H. O. N. . Fast webpage classification using url features. In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM).(2005)

[2]. Garera, S., Provos, N., Chew, M., And Rubin, A. D. . A Framework for Detection and measurement of phishing attacks. In Proceedings of the ACM Workshop on Rapid Malcode (WORM). Alexandria, VA.(2007).

[3]. McGrath, D. K. And Gupta, M. . Behind phishing: An examination of phisher modi operandi. In Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET).(2008).

[4]. Provos, N.,Mavrommatis, P.,Rajab,M. A., And Monrose, F. All your iFRAMEs point toUs. In Proceedings of the USENIX Security Symposium.(2008)

[5]. Fette, I., Sadeh, N., And Tomasic, A. Learning to detect phishing emails. In Proceedings of the International World Wide Web Conference (WWW).(2007)

[6]. Bergholz, A., Chang, J.-H., Paass, G., Reichartz, F., And Strobel, S. Improved Phishing Detection using Model-Based Features. In Proceedings of the Conference on Email and Anti-Spam (CEAS).(2008)

[7]. Kolari, P., Finin, T., And Joshi, A. SVMs for the blogosphere: Blog identification and splog detection. In Proceedings of the AAAI Spring Symposium on Computational Approaches to Analysing Weblogs.(2006)

[8]. Wang, Y.-M., Beck, D., Jiang, X.,Roussev, R.,Verbowski, C.,Chen, S., And King, S. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In Proceedings of the Symposium on Network and Distributed System Security (NDSS).(2006)

[9]. Moshchuk, A., Bragin, T., Deville, D., Gribble, S. D., And Levy, H. M. SpyProxy: Execution-based detection of malicious web content. In Proceedings of the USENIX Security Symposium.(2007).