International Journal of Pure and Applied Mathematics
Volume 118 No. 20 2018, 4159-4163
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

URL Phishing Analysis using Random Forest

S. Jagadeesan (Asst. Professor)(jagadeesan.s@ktr.srmuniv.ac.in),
AnchitChaturvedi(anchitmudit@gmail.com), Shashank Kumar(shashank.kumar14@gmail.com),
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai.

***Abstract*—Phishing is quite a popular form of cyber-attacks these days in which the user is made to visit illegitimate websites. In these websites, the user can be tricked into revealing his sensitive information such as username, passwords, bank details, card details, etc. Thus, it has been used quite a lot by phishers to obtain a user's credentials. The phishing URLs look almost similar to the legitimate ones but actually differ in some respect. In our method, we make use of only the information about the URL of a website to determine whether the website is a phishing website or not. Thus, there is no need of actually visiting a website to determine whether it is phishing or not. This also allows the user to not visit the phishing websites and expose themselves to malicious codes that it may carry. Also, we discuss how meta data of the URLs can be used to determine whether the URL is phishing or not. Random forest and SVM algorithms can then be applied to a dataset having such features that contain the meta data of the URLs. Random forest algorithm offers the advantage of not overfitting the data as well.**

***Keywords*: URL phishing, Random Forest, Classification, Machine Learning**

I. INTRODUCTION

THISPhishing URL is a URL that is created to obtain a user's personal credentials like usernames and passwords, to download some malicious malware in a user's computer or to manipulate search engine's results for a user [13]. In a typical phishing attack, the user can be tricked into clicking some link to a phishing website where they can be made to reveal their sensitive information like usernames and passwords [4]. The phisher may replicate a legitimate website and can have some click baits in this website that can be used to trick a user into revealing their credentials. These credentials can further be made use of by the phisher for digital identity thefts or some financial profits as well [10][11][12]. Online banking and debit card payments have also become quite popular from the past few years. These phishing websites can thus be used by the phishers to get a user's card details or other bank details as well. Sometimes they can be tricked into making banking transactions into illegitimate websites as well. This is quite a dangerous situation for a user. So, a phisher can have access to a user's identity and can misuse it for personal benefits. Motivated by this, there has been a lot of increase in phishing attacks these days. So, a lot of efforts have been made in this field so as to reduce these phishing attacks.

We can detect whether a website is a phishing website or not by looking at the contents of the website or the webpage (such as the words in the website) [3], or by making use of the meta data of a URL [4][7]. In our work, we have made use of the meta data of the URL of a website to detect whether it is a phishing website or not. By making use of the meta data of the URL, we do not have to visit any phishing website or download any of its contents, and thus it is a safer approach.

We can make use of certain features of a URL such as number of slashes, keyword within path portion of URL, etc., to perform the classification [7]. We make use of a number of such features in our work to determine whether a URL is a phishing URL or not. After obtaining the required information about a number of URLs, we just need to perform classification using certain algorithms. We chose Support Vector Machine (SVM) and Random Forest algorithm for this work.

The remaining sections in this paper are organized as follows: the next section talks about the related works in this field. Section 3 introduces our proposed approach. It discusses what datasets were used for the work and why. Also, we discuss about the classification algorithms that were used for the classification between phishing and non-phishing URLS. In section 4, we discuss about the results that were obtained on the datasets from the training models.

II. RELATED WORK

In [2], they have talked about "drive-by-downloads". In such an attack, a web user downloads malicious codes, unintentionally. These codes are JavaScript codes that can be used to attack a web browser. This attack results in download of malwares that can take over the control of the computers and can harm the various files in it. These codes are not easy to detect. In this paper, they discuss about finding certain properties of a JavaScript code that may contain harmful malwares alongwith it. They work by finding out anamolous features for each of the visited web page. They work on an instrumented browser to check for the anamolous features for any executed HTML elements or JavaScript codes. By using this methodology, they were able to obtain 10 such features of a malicious webpage. There can be different attacking classes which may not have these features. So, the system won't work in such cases.

In another method[3], they make use of the words contained in a website. Brand names can be placed in the URL by the phishers to convince the user of their authenticity. So, in this methodology, they focus on the HTML content of a webpage. They perform TD-IDF weighting to assign weights to the various words in the HTML content. Then further weighting is performed by considering the URL weighting system which helps in getting the brand name. This brand name is then searched with the search engine to get the domain name of the webpage. Then a WHOIS lookup is performed to check whether the domain names match. If they do so, the website is considered legitimate. Thus, it mainly makes use of the 'words' that are contained in the 'webpage' rather than looking up for features about the URL.

R. B. Basnet and A. H. Sung[4] introduce us to a broader definition of website phishing. This tells how it can be used to trick a user into visiting a new website and revealing their username, password and other sensitive information. It talks about how meta data is a very useful tool in determining whether a URL is phishing or not. Meta data can be made available about a URL from the various search engines like Google, Yahoo, Bing, etc. This meta data can help in differentiating a phishing URL from a non-phishing one. It proposed the use of Logisctic Regression as a classifier. But it can fail to work when there are a large number of features and are not necessarily linear in nature.

An analysis of anamoly detection techniques was performed by Mohiuddin Ahmed, AbdunNaserMahmood and Jiankun Hu[5]. Anamolous behaviors are important to detect as they can tell about a suspected attack on a system. A computer could be showing an anamolous behavior because of a certain malware attack. Thus it is important to detect any anamolous behavior of any system. The paper then discusses clustering and classification techniques (like SVM, Neural Networks, etc.) for anamoly detection.

In [6], they made use of the various Domain names (like PrimaryDomain, SubDomain, PathDomain, etc.) and checked whether they have been correctly spelled or not for a legitimate website. They also considered the Page rank as a feature to determine whether the website is phishing or not.

S. CarolinJeeva and Elijah Blessing Rajsingh [7] majorly focused on finding the features that are essential in discriminating a phishing website from a non-phishing one. They performed 'association rule mining' which can help in distinguishing between the two. It revealed a number of features that can be useful in performing URL phishing analysis. They made use of the apriori algorithm to determine the rules that can be used to determine a phishing website. This revealed a number of useful features for the classification like number of slashes, keyword within the path portion of the URL, etc.

In the past works [8][9], Neural Networks model has been used in doing the classification process. But it is prone to underfitting if it is poorly structured [8]. And, on the other hand, it can overfit into the training data set if it is structured to meet every single item in the dataset [9][14].

III. PROPOSED METHOD

In this method, we are going to make use of two different datasets to select the appropriate model. Both of the datasets are obtained from UCI Machine Learning Repository[15]. One dataset consists of 30 features and 1 target feature. It consists of 2456 entries of phishing as well as non-phishing URLs. This dataset consists of some of the features that were determined essential for this task in [7]. The features such as presence of double slashes, or some keywords in the URL portion are present in this dataset, as were found essential in the work by [7]. There are some additional features present in the dataset such as presence of IP address in URL, length of URLs, having '@' symbol or not, etc.

The second dataset consists of 1353 URLs with 10 features, and these URLs are classified into 3 categories: Phishing, non-phishing and suspicious. We are going to make use of both of these datasets.

The first thing to do after the datasets are obtained is data slicing. Here, we divide the datasets into two parts: testing dataset and training dataset. The training dataset is used to train a model. The testing dataset is only used once the trained model is ready. Once the model is trained, we test its accuracy on the testing dataset.

While training the model on the training dataset, we check its accuracy by performing repeated cross-validation on the dataset. This also allows us to perform tuning of the parameters in the two datasets to check out which parameter gives the best accuracy for the dataset. Once, the most suitable parameter for the model has been determined, the training of the model is complete, and then we can move on to perform testing of the trained model on the testing dataset.

We are going to perform the classification task on the datasets. We make use of two algorithms to perform the classification task: Support Vector Machines (SVM) and Random Forests.

A. Random Forest

Random Forest is a supervised machine learning algorithm that can be used to perform both regression and classification task in data mining. It is an ensemble based technique that can be used to perform classification. It makes use of a number of classification trees (like decision trees) and then gives the final result.

This algorithm works by creating a number of classification trees randomly. These trees are created by making use of different samples from the same dataset and also they may use different types of features each time to create the trees. Thus, all the trees are created randomly by making use of different sub sets of the same dataset, and also the features are taken randomly for the creation of any tree. By doing so, Random Forest ensures that it does not overfit the data, as in the case of the decision trees. Once the trees have been formed, we can do the classification by finding the results of each tree and then assigning it to the class that has been determined by the most number of trees.

B. Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning algorithm that can be used to perform the classification task in data mining. It works by creating hyper planes to create boundaries between the various classes. It creates boundaries between the various classes by the use of a hyper plane. The hyper plane is so chosen such that it separates the different classes and also maximizes the margin (distance from the nearest point) with the different classes. Once the hyper plane is created, a boundary is obtained between the various classes. Now, we can classify any data point to a class by finding out which class the data point lies in. It is of two types:

*1) Linear SVM:*This is the simple form of SVM, in which the algorithm tries to form a simple straight line as a hyper plane as a boundary to separate the different classes. It works very well on linearly separable data.

2) Non-Linear SVM: In the real world datasets, the data is not usually linearly distributed and thus the various classes cannot be separated with a simple straight hyper plane. Additional work needs to be performed to separate the various classes with a hyper plane. So, here the hyper planes are not straight lines. They are a better version of SVM that helps to classify Non-linear data as well.

IV. RESULTS AND ANALYSIS

The datasets were split into two parts, training and testing, in the ratio 70:30. The training dataset was mainly used to train the different models. And then the trained model was applied on the testing dataset to get the results.

While training the model, the 'repeated cross-validation' method was used to determine the accuracy of the trained model. The cross-validation was done by creating '10' folds of the training dataset. The process was repeated '3' times. And thus, we obtain the accuracy result of a trained model.

For tuning of parameters, grid search was used and repeated cross-validation was performed on these values to determine the most suitable parameter for a model. This revealed the most suitable parameter for the model which gives the highest accuracy on the training dataset.

While testing this trained model on the testing dataset, confusion matrix was used to determine the accuracies on the testing dataset.

First of all, the SVM Linear algorithm was applied on the first dataset. Grid search was used to tune this algorithm for different values of its tuning parameter, 'c'. The most suitable for 'c' was found to be 0.5. With this value of 'c', an accuracy of 92.62% was obtained on the training dataset. And when this model was applied on the testing dataset, an accuracy of 93.07% was obtained.

After this, SVM Non-linear algorithm was applied. The SVM Radial algorithm was used here. It consists of two tuning parameters, 'c' and 'sigma'. Different combinations of these parameters were tried for tuning of the parameters, and the most suitable values that gave the highest accuracy on the training dataset were c=1.5 and sigma=0.06. With these values, an accuracy of 94.54% was obtained on the training dataset and an accuracy of 94.97% was obtained on the testing dataset.

Random Forest was then applied on this dataset. Grid search was used to performing tuning of its parameter, 'mtry'. The most suitable value for mtry was found to be 6 which gave the highest accuracy on the training dataset. The accuracy on the training dataset using this value was 94.75%. And when this trained model was applied to the testing dataset, an accuracy of 95.11% was obtained. Thus, Random Forest gave better results than SVM-Linear as well as SVM Non-Linear algorithms, both on the training and the testing datasets.

The results for the first dataset are summarized in table 1.

TABLE I

ACCURACY OBTAINED ON DATASET I

Algorithm	Accuracy on training dataset (in %)	Accuracy on testing dataset (in %)
Linear SVM	92.62	93.07
Non-Linear SVM	94.54	94.97
Random Forest	**94.75**	**95.11**

Similar process was repeated with the second dataset. With SVM Linear algorithm, the most suitable value for the tuning parameter 'c' was found to be '1.5'. With this value, an accuracy of83.23 % was obtained on the training dataset and an accuracy of 85.19% was obtained on the testing dataset.

With SVM Radial algorithm, the highest accuracy for the trained model was 88.78%, which was obtained when the value of c was 15 and sigma was 0.09. Using this model on the testing dataset, an accuracy of 89.63% was obtained.

When Random Forest algorithm was used on this dataset, the value of mtry=5 gave the highest accuracy on the training dataset. The accuracy that was obtained was 89.52%. On the testing dataset, this model was able to give an accuracy of 90.12%. Thus, Random Forest performed better than the SVM linear as well as SVM non-linear algorithm, on the training and the testing dataset of the second dataset as well.

The results for the second dataset are summarized in table 2.

TABLE II

ACCURACY OBTAINED ON DATASET II

Algorithm	Accuracy on training dataset (in %)	Accuracy on testing dataset (in %)
Linear SVM	83.23	85.19
Non-Linear SVM	88.78	89.63
Random Forest	**89.52**	**90.12**

V. CONCLUSION

URL phishing analysis is very useful in determining whether a certain URL is a legitimate URL or not and whether it

should be visited or not. This helps the users a lot in knowing which of the websites should be avoided. Thus, it prevents them in revealing their sensitive information to unknown or illegitimate sources. Features of the URL should be chosen carefully to get better results.

In our work, we made use of a powerful classifier, namely Random Forest, to perform the classification. We compared its classification performance with SVM (Linear as well as Non-linear) algorithm. The comparison was done by taking two different datasets for URL Phishing. The first dataset consisted of 31 features and it had around 2500 entries for phishing and non-phishing URLs. The second dataset consisted of just 10 features with 1353 entries of URLs. Random Forest performed better than SVM (linear as well as Non-linear) algorithm on both of these datasets. Another advantage that Random Forest offers is that it doesn't overfit the data, if the parameters are tuned and selected carefully. Thus, they are a suitable choice for use in URL Phishing datasets and in determining whether a URL is phishing or not.

REFERENCES

[1] Zuochao Dou; Issa Khalil; AbdallahKhreishah; Ala Al-Fuqaha; Mohsen Guizani, "Systematization of Knowledge (SoK): A Systematic Review of Software-Based Web Phishing Detection", IEEE Communications Surveys & Tutorials, 2017.

[2] Marco Cova, Christopher Kruegel, Giovanni Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code", Proceedings of the 19th International Conference on World Wide Web, pp. 281-290, 2010.

[3] Choon Lin Tan, Kang LengChiew, San Nah Sze, "Phishing Website Detection Using URL-Assisted Brand Name Weighting System", 2014 IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) December 1-4, 2014.

[4] R. B. Basnet, A. H. Sung, "Mining web to detect phishing urls", Proceedings of the International Conference on Machine Learning and Applications, vol. 1, pp. 568-573, Dec 2012.

[5] Mohiuddin Ahmed, AbdunNaserMahmood, Jiankun Hu, "A survey of network anomaly detection techniques", J. Netw. Comput. Appl., vol. 60, no. C, pp. 19-31, 2016.

[6] LuongAnh Tuan Nguyen, Ba Lam To, HuuKhuong Nguyen1 and Minh Hoang Nguyen, "A Novel Approach for Phishing Detection Using URL-Based Heuristic", 2014 International Conference on Computing, Management and Telecommunications (ComManTel), IEEE 2014.

[7] S. CarolinJeeva, Elijah Blessing Rajsingh, "Intelligent phishing url detection using association rule mining", Human-centric Computing and Information Sciences, vol. 6, no. 1, pp. 10, 2016.

[8] S. Duffner and C. Garcia, "An Online Backpropagation Algorithm with Validation Error-Based Adaptive Learning Rate," in Artificial Neural Networks – ICANN 2007, Porto, Portugal, 2007.

[9] R. M. Mohammad, F. Thabtah and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, no. 2, pp. 443-458, 2013-B.

[10] HibaZuhair, Ali Selamat, MazleenaSalleh, "Feature selection for phishing detection: a review of research", International Journal of Intelligent Systems Technologies and Applications, Vol. 15, No. 2, 2016

[11] Huang, H., Tan, J. and Liu, L. (2009) 'Countermeasure techniques for deceptive phishing attack', International Conference on New Trends in Information and Service Science (NISS'09), 30 June–02 July, 2009, China, pp.636–641.

[12] Mayuri, A. and Tech, M. (2012) 'Phishing detection based on visual-similarity', International Journal of Scientific and Engineering Research (IJSER), Vol. 3, No. 3, March, pp.1–5.

[13] Anvil, Search Engine Optimization Whitepaper, http://www.anvilmediainc.com/wpcontent/uploads/ami_seo_whitepaper_1104.pdf

[14] FadiThabtah, Rami M. Mohammad, Lee McCluskey, "A Dynamic Self-Structuring Neural Network Model to Combat Phishing", 2016 International Joint Conference on Neural Networks (IJCNN), 2016

[15] UCI Machine Learning Repository, https://archive.ics.uci.edu/ml

[16] http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/