Zararlı URL Veri Seti Oluşturulup Özellik Çıkarımı Yapılması - Creating Phishing URL Data Set and Doing Feature Extraction

İlkem İnan Ak, Helim Doğuş Toygur Kukul
Computer Engineering Department
Yıldız Technical University, 34220 Istanbul, Turkey
ilkeminan1@gmail.com, doguskukul@gmail.com

***Özetçe* —Oltalama amaçlı sitelerin varlığı sürekli artmaktadır. Kullanıcılar saldırganların hazırladığı oltalama amaçlı sitelere kişisel bilgilerini girerek zarara uğrayabilmektedir.**

Oltalama amaçlı siteler URL'lere bakılarak tespit edilebilir. Bu bazen yeterli bilgiye sahip bir kullanıcının URL'e bakmasıyla anlaşılabilir. Bazense saldırganlar URL'i taklit edilen sitenin URL'ine oldukça başarılı bir şekilde benzettiğinden anlaması zor olabilmektedir. Peki bilgisayarlar bir URL'in oltalama amaçlı ya da temiz olduğunu nasıl anlayabilir? URL'leri genellikle oltalama amaçlı ya da temiz olarak sınıflandırmaya yarayan belirli özellikler vardır. Bilgisayarın başarılı tahminlerde bulunabilmesinin yolu yeterince anlamlı özelliklerle eğitilmesidir.

Bu projede ilk önce temiz ve oltalama amaçlı URL'ler elde edilerek veri seti oluşturulmuştur. Bu veri seti bir ön işleme aşamasından geçip özellik çıkarımı yapılmıştır. Daha sonra sınıflandırma için daha etkili özellikleri belirlemek için özellik seçimi yapılmıştır. Son olarak da makine öğrenmesi algoritmalarıyla modeller oluşturularak sınıflandırmalar yapılmıştır. Makine öğrenmesi algoritmalarından alınan sonuçlar karşılaştırılmıştır.

Anahtar Kelimeler—Url, oltalama, temiz, veri setinin oluşturulması, özellik çıkarımı, özellik seçimi, makine öğrenmesi

***Abstract*—Phishing websites are growing problem worldwide. Users can be damaged by entering their personal informations to phishing websites which are created by attackers.**

Phishing websites can be detected from the URL. Sometimes it can be detected by the users who has enough knowledge. But sometimes it can be hard to detect. Because attackers may liken their URL to imitated URL. How can computers predict that whether a url is phishing or not? Usually URLs can be classified from specific features. The way computers can make successful predictions is to be trained by good features.

Firstly, we obtained phishing and legitimate URLs and created the data set. After doing a preprocessing, we did feature extraction. Later, we used feature selection algorithms in order to get most significant features. Finally, we classified the URLs with machine learning algorithms. We compared the results of the machine learning algorithms.

Keywords—Url, phishing, legitimate, creating data set, feature extraction, feature selection, machine learning

I. Introduction

The widespread use of the Internet has led to a large increase in the number of websites. But all of this websites are not legitimate. Attackers can harm users with the phishing technique. Phishing is based on the theft of user information by imitating the services of existing websites. Phishing attacks can be done in various ways such as e-mail, sms or pop-ups. By using these methods, the link leading to the user is shown. If the user clicks on the link, website of the attacker, which is very similar to the imitated site, is opened. If the user enters their personal information to the form screen on the website, this information is passed into the attacker and user is harmed. Phishing attacks can cause serious damage due to some weakness and carelessness of users. Detecting phishing websites is very important to reduce these damages.

In this study, it is aimed to make phishing url classification with using url based features. We created the data set by building Crawlers for Google, Dmoztools and Phishtank. We used 17 features and selected the most efficient features with feature selection algorithms. We used the machine learning algorithms to classify phishing URLs. The machine learning algorithms which we applied, are Logistic Regression, Naive Bayes, K Nearest Neighbors, Support Vector Machines, Decision Tree, Random Forest and AdaBoost.

II. Related Work

There are many academic studies in the literature for detecting phishing URLs. Different methods such as artificial neural networks and machine learning have been used in these studies.

Rami Mustafa Mohammad, T.L. Mccluskey and Fadi Thabtah [1] used artificial neural networks in their study in order to detect the phishing URLs. In the method they use, artificial neural networks automatically form themselves. They used 600 legitimate URLs and 800 phishing URLs as the data set. They used the data set with 17 features (such as the '@' symbol in the URL, URL length, etc.) for the URLs. Using these 17 features, the system configures itself. They have received approximately 92.18% successful prediction results.

Mustafa Kaytan and Davut Hanbay [2] used extreme learning machines to identify the phishing websites. In their study, they used 30 features for a total of 11055 URLs. They achieved approximately 95.05% successful prediction rate.

Santhana Lakshmi and Viyaja [3] made a study to detect phishing URLs with supervised machine learning algorithms. They used 17 features in their studies. They used 100 legitimate URLs and 100 phishing URLs as the data set. Their algorithms were 'Multilayer Perceptron', 'Decision Tree Induction' and 'Naive Bayes'. 'Multilayer Perceptron' algorithm had 97% success rate whereas 'Decision Tree Induction' algorithm had 98.5% success rate and 'Naive Bayes' algorithm were 93.5% successful.

Daisuke Miyamato, Hiroaki Hazeyama, and Youki Kadobayashi [4] have made a study of detecting phishing URLs with machine learning methods. They used 'AdaBoost', 'Bagging', 'Support Vector Machines (SVM)', 'Logistic Regression', 'Classification and Regression Trees (CART)', 'Random Forest', 'Neural Network', 'Naive Bayes' and 'Bayesian Additive Regression Trees (BART)' algorithms. They used 1727 phishing URLs and 1273 legitimate URLs as data set. As a result of their studies, the most successful algorithm was the 'AdaBoost' algorithm with a ratio of 85.81%. This algorithm was followed by 'Neural Network' (85.70%), 'SVM' (85.62%), 'BART' (85.55%), 'Random Forest' (85.54%), ' Logistic Regression (85.48%), 'Naive Bayes' (85.47%), 'Bagging' (85.27%) and 'CART' (83.84%).

III. Methods

A. Creating Data Set

Data is obtained in different ways to create the data set. Crawlers are written to collect datas. Legitimate URLs are collected from Google and Dmoztools, while phishing URLs are collected from Phishtank. Python's Requests and BeautifulSoup modules are used to collect data. Requests module sends requests to web pages. With BeautifulSoup module, page sources of web pages are taken.

Google's search results are used to collect URLs from Google. When a keyword is entered into Google, the URLs in the results are collected. The biggest obstacle to Google's data collect is Google's limit on data that is collected. With this limit, a certain number of data can be extracted at a certain time. When checking this, requests from the same 'user agent' are checked. To prevent this, Python's fake-useragent module is used. With this module, it can be seen that the request is sent from different 'user agents' and the data collect limit can be exceeded.

In Dmoztools, URLs are kept in categories (such as Arts, Business, Computers) and subcategories of these categories. With BeautifulSoup, the page resources are accessed to collect the URLs found in these categories.

The Phishtank website has a 'Phish Search' section. When the 'Validation property' in this section is searched for 'Valid phishes', URLs that are determined as phishing are listed by filtering. On this page, the 'ID' of the URL in the Phishtank database, the content of the URL, the 'validity' property (the search result is all 'valid phish'), and 'online' properties are showed on table. Each page lists 20 URLs. The search type and page number appear in the URL. By looping the page number, the request is sent to the filtered URL and the BeautifulSoup module accesses the page source. The contents of the table are accessed from the page source. The biggest obstacle to extracting data from Phishtank is that long URLs in the table have '...' at the end. Length is often one of the most important features that make a URL phishing. Therefore, only short URLs should not be collected. As a solution, the 'ID' of the URL in Phishtank is collected instead of URL. After that, these 'ID's are searched in the 'phish detail' section of Phishtank. A request are sent to these URLs that have 'IDs' by loop and the page source is received with the BeautifulSoup module. Exact URLs can be obtained from these page sources.

B. Preprocessing

Most of the URLs in Dmoztools have 'http'. But contrary to what is seen in Dmoztools, the protocol of most of these URLs is actually 'https'. Having 'https' protocol is often one of the most important features of legitimate URLs. Therefore, URLs should not be used same as that are collected from Dmoztools. A preprocessing is required for correcting incorrect URLs. In order to correct URLs, a request is sent to the URLs with Python's Requests module. The result of the request can be kept in a variable named 'response'. When the 'url' property of this 'response' variable is received, the corrected states of the URLs are obtained. This can correct incorrect URLs that collected from Dmoztools.

C. Feature Extraction

URLs have features that make them often phishing or legitimate. These features can be of various types, such as URL-based, content-based, HTML-based. Only URL-based features are used in this project. Numerous academic studies in the literature are examined to determine URL-based features. The URL-based features used in these studies are determined. The most commonly used features are selected to use in this project. The features obtained for each URL are added to the data set. The features that are used in this study are below.

1) Domain Name in IP Adress Form: If the domain name contains an IP address, it is classified as phishing.

2) Using the Https Protocol in the URL: If the protocol is 'https', it is classified as a legitimate URL and if the protocol is 'http', it is classified as phishing.

3) Existence of 'SPAM' in the URL Instead of 'http': If the URL contains 'SPAM', it is classified as phishing.

4) Number of Dots in the URL: If the URL contains more than two dots, it is classified as phishing.

5) Number of Slashes (/) in the URL: If the URL contains more than six slashes, it is classified as phishing.

6) Number of Numbers in the URL: If the URL contains more than six numbers, it is classified as phishing.

7) Existence of Sensitive Words in the URL: Phishing URLs usually contain certain specific words such as 'secure', 'account' and 'login'. If the URL contains these words, it is classified as phishing.

8) Existence of Uppercase Letters in the URL: If the URL contains uppercase letter, it is classified as phishing.

9) URL Length: If the URL contains more than 100 characters, it is classified as phishing.

10)Existence of Suspicious Characters in the URL: If the URL contains certain specific characters such as '@', '&', '!', '?', '=', '$', '*' and '+', it is classified as phishing.

11)Prefix-Suffix: If the URL contains '-' character, it is classified as phishing.

12)Number of TLDs in the URL: If the URL contains more than one TLD (Top Level Domain), it is classified as phishing.

13)Entropy: If the entropy of the URL is greater than 4.8, it is classified as phishing.

14)Existence of Brands in the URL: If the URL contains certain specific brands such as 'Facebook', 'Google', 'Apple' and 'Amazon', it is classified as phishing.

15)Misuse of 'www' in the URL: If the URL contains 'www' in the wrong place, it is classified as phishing.

16)Existence of 'www' in the URL: If the URL contains 'www' after protocol and '://', it is classified as legitimate.

17) Extraction of Extensions: If the URL contains extensions such as '.jpg', '.rar' and '.css', it is classified as phishing.

D. Feature Selection

Some features are more effective than other features in distinguishing whether a URL is phishing or legitimate. In order to obtain better machine learning results, the determined features should be selected. Results are obtained and compared from some feature selection algorithms to determine the most meaningful feature groups.

In this study, we used Chi-Square, F-Test, Mutual Information, Logistic Regression, Random Forest and L1-based Feature Selection algorithms. The reason for selecting these algorithms is that they give the same kind of results. As a result, these algorithms give a list of 'True' and 'False' contents. They give 'True' for selected features and 'False' for non-selected features. The features that take the most 'True' values are selected to use in machine learning.

E. Machine Learning

Models are created for 7 different supervised machine learning algorithms. We used Logistic Regression, Naive Bayes, K Nearest Neighbors, Support Vector Machines, Decision Tree, Random Forest and AdaBoost algorithms. 70% of the data set is used for training and 30% is used for testing. The results of each machine learning algorithms are compared.

IV. Experimental Results

We used True-Positive Rate, False-Positive Rate, F Score and Accuracy as performance metrics.

True-Positive Rate: It is used to calculate the correct estimation rate of the selected class (1).

$$TPRate = \frac{TP}{TP+FN} \tag{1}$$

False-Positive Rate: It is used to calculate the wrong estimation rate of the selected class (2).

$$FPRate = \frac{FP}{FP+TN} \tag{2}$$

F Score: It (5) is calculated as the harmonic mean of Precision(P) (3) and Recall(R) (4) values.

$$P = \frac{TP}{TP+FP} \tag{3}$$

$$R = \frac{TP}{TP+FN} \tag{4}$$

$$F = 2 * \frac{P*R}{P+R} \tag{5}$$

Accuracy: It is ratio of correctly predicted observation to the total observations. (6)

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{6}$$

Logistic Regression (LR), Naive Bayes, K Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Tree, Random Forest and AdaBoost algorithms were tested and compared. Since the splitting of the data set for training and testing is random, different results can be obtained in each trial. For this reason, each algorithm was tried 5 times and the average of 5 trials were taken. For each algorithm, TP (True-Positive) Rate, FP (False-Positive) Rate, F Score and Accuracy are shown in Table 1.

Table 1 Comparison of machine learning algorithms

Algorithms	TP Rate	FP Rate	F Score	Accuracy
LR	87.08%	17.90%	84.98%	84.59%
Naive Bayes	85.96%	22.74%	82.39%	81.61%
KNN	85.31%	13.96%	85.63%	85.67%
SVM	87.62%	14.76%	86.60%	86.43%
Decision Tree	87.53%	14.62%	86.61%	86.45%
Random Forest	87.52%	14.61%	86.61%	86.45%
AdaBoost	87.54%	14.66%	86.60%	86.44%

In this study, the algorithms with the highest accuracy were Decision Tree and Random Forest with accuracy rate of 86.45%. These algorithms were followed by AdaBoost (86.44%), SVM (86.43%), KNN (85.67%), Logistic Regression (84.59%) and Naive Bayes (81.61%). The most successful algorithms for the F score were Decision Tree and Random Forest with a success rate of 86.45%.

Decision Tree and Random Forest algorithms give very close results in all performance metrics. This is because they are basically similar (tree-based) algorithms.

The true positive rate of the Naive Bayes algorithm is 85.96%, whereas the true positive rate of the KNN algorithm is 85.31%. Although the Naive Bayes algorithm has not the lowest true positive rate (KNN algorithm has the lowest true positive rate.), it has the lowest accuracy rate. This is because Naive Bayes has the highest false positive rate with the rate of 22.74%. The KNN algorithm has the lowest false positive rate with the rate of 13.96%. Logistic Regression has the highest false positive rate after the Naive Bayes algorithm with 17.90% false positive rate.

The effect of the features used for machine learning on performance criteria varies. Features that do not contribute a great deal alone, when they come together, become more efficient. To better illustrate this, the results obtained when each feature is used alone is shown in Table 2.

Table 2 Comparison of the features

Features	TP Rate	FP Rate	F Score	Accuracy
IP form	99.97%	97.64%	67.27%	51.21%
https	82.65%	43.92%	72.92%	69.34%
SPAM	100.0%	99.79%	66.66%	50.05%
Dot count	94.77%	67.73%	72.26%	63.56%
'/' count	94.13%	77.49%	69.33%	58.33%
Number count	91.18%	69.59%	69.96%	60.82%
Sensitive words	98.72%	66.12%	74.49%	66.25%
Uppercase letter	87.73%	63.91%	69.76%	61.93%
Length	94.41%	79.86%	68.91%	57.33%
Suspicious char	99.53%	86.31%	69.53%	56.50%
Prefix-suffix	63.16%	52.71%	58.48%	55.21%
TLD count	99.69%	94.00%	67.89%	52.85%
Entropy	98.63%	84.92%	69.57%	56.85%
Brands	92.61%	76.74%	68.74%	57.91%
'www' misuse	99.70%	97.92%	67.04%	50.94%
'www' existence	67.81%	13.88%	74.66%	76.95%
Extensions	88.23%	57.80%	71.70%	65.19%

According to Table 2, the feature with the highest accuracy rate was 'www existence' with the accuracy rate of 76.95%. This feature was followed by 'https' (69.34%), 'Sensitive words' (66.25%), 'Extensions' (65.19%), 'Dot count' (63.56%), 'Uppercase letter' (61.93%), 'Number Count' (60.82%), '/ Count' (58.33%), 'Brands' (57.91%), 'Length' (57.33%), 'Entropy' (56.85%), 'Suspicious characters' (56.50%), 'Prefix-Suffix' (55.21%), 'TLD Count' (52.85%), 'IP Form' (51.21%), 'www misuse' (50.94%) and 'SPAM' (50.05%).

After the feature selection, the selected features were 'www existence, 'https', 'Sensitive words', 'Extensions', 'Dot count', 'Uppercase letter', 'Brands', 'Entropy' and 'Suspicious characters'. The 6 features with the highest accuracy were selected. However, 'Number count' and '/ count' could not be selected despite having higher accuracy rates than 'Brands'. Also, the 'Length' feature could not be selected despite the fact that it has a higher accuracy rate than the 'Entropy' and 'Suspicious characters' features. This is because the 'Number count', '/ count' and 'Length' features are less effective when used in combination with other features. In other words, URLs that these features classify as phishing are often classified as phishing by other features as well.

The feature with the lowest accuracy rate is 'SPAM' with the accuracy rate of 50.05%. Although this feature has the lowest accuracy rate, it also has the highest true-positive rate with the true-positive rate of 100%. A small number of URLs in the dataset replace 'http' with 'SPAM'. However, all URLs that contains 'SPAM' instead of 'http' in the dataset are phishing. Therefore, this feature has both the highest true-positive ratio (100%) and the highest false-positive ratio (99.79%).

V. Result

With the increase in the use of the Internet, the number of sites that are phishing is also increasing significantly. As a result of the researches, it has been found that phishing URLs increased by 400% between January 2019 and July 2019[5]. This situation shows the importance of the measures to be taken for this subject. The main goal of this project is to prevent this problem by detecting phishing URLs.

The first thing to do in this direction is to create a data set. In order to achieve successful results, different methods of collecting URLs were used. The features separating the phishing URL addresses from the legitimate URL addresses were determined by examining literature and the data set. In this project, URL based features were used. Because domain-based features are extracted over a long period of time and have some limitations. After the features were determined, the features selected by more algorithms were determined by using feature selection algorithms in order to select useful features. According to the results of these tests, the best algorithms were determined as Decision Tree and Random Forest. After these results were obtained, the machine learning algorithms were tested individually for each feature in order to determine which features were effective.

In our study, a larger data set was used, when compared to other studies on this subject. This increases the consistency in the result. The used features are not specifically prepared for the used data set, and can be used in other studies. Also, the effectiveness of each feature is shown in the comparative table, and it sheds light on the studies that will be done using less features. For these reasons, It is thought that our study can contribute to the literature.

References

[1] 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, 2014.

[2] M. Kaytan and D. Hanbay, "Effective classification of phishing web pages based on new rules by using extreme learning machines," *Anatolian Science-Bilgisayar Bilimleri Dergisi*, vol. 2, no. 1, pp. 15–36, 2017.

[3] V. S. Lakshmi and M. Vijaya, "Efficient prediction of phishing websites using supervised learning algorithms," *Procedia Engineering*, vol. 30, pp. 798–805, 2012.

[4] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, "An evaluation of machine learning-based methods for detection of phishing sites," in *International Conference on Neural Information Processing*. Springer, 2008, pp. 539–546.

[5] "Help Net Security phishing attempts increase 400%, many malicious urls found on trusted domains," https://www.helpnetsecurity.com/2019/10/09/phishing-increase-2019/.