2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Detecting Phishing Websites and Targets Based on URLs and Webpage Links

Huaping Yuan[1,2], Xu Chen[1,3], Yukun Li[1,3], Zhenguo Yang[1,2]*, Wenyin Liu[1,2]*

[1]Web Identity Security Lab; [2]School of Computer Science and Technology; [3]School of Automation
Guangdong University of Technology, Guangzhou, P.R. China
Email: yuanhuaping@outlook.com; csx6688@gmail.com; gdutkelvin@outlook.com; zhengyang5-c@my.cityu.edu.hk; liuwy@gdut.edu.cn

***Abstract*—In this paper, we propose to extract features from URLs and webpage links to detect phishing websites and their targets. In addition to the basic features of a given URL, such as length, suspicious characters, number of dots, a feature matrix is also constructed from these basic features of the links in the given URL's webpage. Furthermore, certain statistical features are extracted from each column of the feature matrix, such as mean, median, and variance. Lexical features are also extracted from the given URL, the links and content in its webpage, such as title and textual content. A number of machine learning models have been investigated for phishing detection, among which Deep Forest model shows competitive performance, achieving a true positive rate of 98.3% and a false alarm rate of 2.6%. In particular, we design an effective strategy based on search operator via search engines to find the phishing targets, which achieves an accuracy of 93.98%.**

Keywords—phishing detection; phishing target; Deep Forest

I. INTRODUCTION

Phishing is the attempt to spoof users to leak their sensitive information, such as usernames, passwords, bank accounts, and credit card numbers. The number and sophistication of phishing attacks have been growing alarmingly in recent years. According to the APWG Global Phishing Survey [1], the total number of phishing attacks in 2016 was 1,220,523 with an increase rate of 65% compared with 2015. Phishing attacks are growing rapidly, making people increasingly concerned about how to prevent phishing attacks.

Phishing detection is a challenging task, which has attracted a lot of research attention from anti-phishing researchers to seek for effective solutions. The list-based anti-phishing approaches (blacklist or whitelist) [2] store URLs in the database, which is used to match the stored URLs with the URLs entered by users in browsers. These approaches perform quickly but fail to detect newly created phishing URLs as they have not been included in the database. Heuristic-based methods [3] usually extract textual features to detect phishing websites, which can detect newly created URLs. However, certain textual features extracted from the textual content of webpages cannot be used to detect phishing websites in other languages. Some researchers propose that similarity-based approaches [4] should be used to compare with the similarity between the given suspicious webpages and legitimate webpages under attack, i.e., phishing targets, which should be known in advance. Lately, phishing targets can be detected automatically [5], unfortunately, its approach is very slow since it needs to obtain and analyze a large number of webpages to form a parasitic community.

In this paper, we propose a machine learning based method for phishing detection, which extracts statistical features and lexical features from URLs and the links inside the webpages. Given the URL representation in vector form, Deep Forest [6] and a number of existing machine learning models, such as GBDT and XGBoost, can be applied seamlessly for phishing detection. The proposed approach works regardless of webpages in different languages. The method is efficient and effective, as shown in our experiments. We also propose an effective method based on search operator[1] to detect phishing target, i.e., the legitimate websites under attack. The main contributions of this paper are summarized as follows:

1. We propose to extract URL features and the statistical features of the links in the webpages to detect phishing webpages, which are effective for phishing detection.

2. We propose a method based on search operator to find the phishing targets of the detected phishing websites, allowing specified matching between query keywords and corresponding sections of the webpages of the phishing target candidates.

3. We investigate and evaluate quite a few machine learning based classification algorithms for phishing detection, among which Deep Forest achieves the highest performance.

This paper is structured as follows: Section II discusses related works. Section III introduces the proposed phishing detection and target detection approaches. Section IV shows the experiments to evaluate the proposed approaches and the baselines. Finally, we conclude this work in Section V.

II. RELATED WORK

A. Phishing Detection

In the last few years, phishing website detection has received much attention in both academia and the industry. Rule-based approach found hidden information about URLs or HTML content and the links between websites [7]. Liu et al. [4] proposed to calculate the visual similarity of suspicious webpages and protected webpages (which are actually potential

*Corresponding authors
[1]http://www.googleguide.com/advanced_operators_reference.html

phishing targets known in advance) for phishing detection. However, the list of protected webpages cannot cover all the webpages that need to be protected, and it suffers from high false alarm rate in practice. Zhang et al. [8] proposed a content-based anti-phishing technique, called CANTINA, which is based on the TF-IDF algorithm and heuristics. Marchal et al. [9] proposed an efficient phishing URL detection system, which relied on URL lexical analysis and leveraged search engine query. However, sending queries over the network and storing large amounts of data lead to high time and space costs. James et al. [10] detected phishing websites based on lexical features, host properties, and importance properties of webpages. This method relied on handcrafted features, which cannot be applied to deal with large-scale datasets. Mohammad et al. [11] proposed a rule-based method to extract multiple features. However, the effectiveness on complicated phishing websites with multiple identities cannot be guaranteed. Srinivasa Rao et al. [12] exploited the behaviors of phishing webpages for phishing detection, such as observing the login status or return results by automatically submitting fake credentials. However, there exist certain restrictions about the login system, e.g., some websites allow users to enter wrong passwords within three times only. In addition, certain webpages' login boxes cannot be correctly detected and hence the fake credentials cannot be submitted automatically. Verma et al. [13] designed a series of lexical URL features, including the frequency of characters in URLs. However, it may not work if a phishing URL does not contain any spelling mistakes. Gowtham et al. [14] proposed an efficient anti-phishing system based on 15 heuristic features with a pre-filtering mechanism. However, the strict login page filtering mechanism relies on the accuracy of the login window detection.

B. *Phishing Target Discovery*

Finding a phishing target benefits to the analysis of the attacking behaviors, and helps users navigate to the legitimate webpages. Phishing webpages are usually associated with their phishing targets. Liu et al. [5] proposed an anti-phishing method by collecting webpages with either direct or indirect association with a given suspicious webpage. However, it requires a quantity of external resources, which leads to high computational and bandwidth cost. Ramesh [15] et al. proposed to group the hyperlinks of the suspicious webpages according to the domains they are associated with. However, it requires to analyze large amounts of links, and phishing target candidate sets may not be in the hyperlinks groups. Liu et al. [16] detected phishing targets of the suspicious webpages based on construction and reasoning of their Semantic Link Network (SLN). The proposed method can detect phishing webpages and find their targets effectively. However, the computational cost of building an SLN is also quite high.

III. THE PROPOSED METHOD

In this section, we first introduce the statistical features and lexical features of URLs and links, and then present the strategies of phishing website detection and phishing target detection, respectively. The overview of the proposed method is shown in Fig. 1.

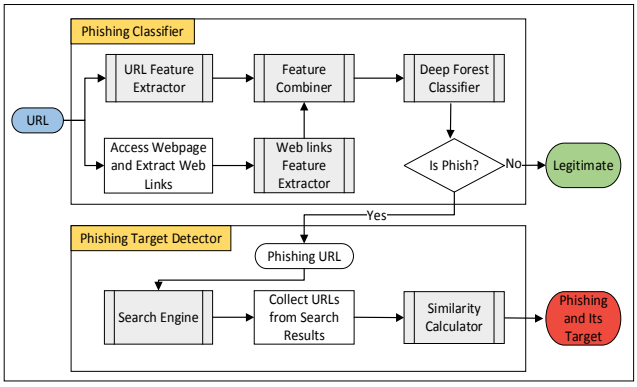

Fig. 1. The overview of the proposed method

A. *Statistical Features of URLs and Links*

In order to mimic legitimate websites, many phishing websites are created quickly by using the data of their targeting websites, such as, links, css. Therefore, we propose to extract two-fold features from URLs and links, i.e., statistical ones and lexical ones. In terms of statistical features, we incorporate certain basic features of a given URL [17,18] with statistic features of the links in its webpage. In addition, we extract a small number of lexical features from the given URL, links, and webpage content such as title, meta-words, and textual content. The basic features of URLs used in our work are detailed as follows:

1) **IP Address**: Legitimate URLs usually have their own domain names, while the phishing ones may use IP addresses directly in their URLs Therefore, we use a binary value to represent this feature.

2) **Suspicious Characters:** Phishers usually use some special characters (e.g., '@', '&', '-', and '_', etc.) in addition to alpha-numeric characters to trick users. The more special symbols are used, the more likely phishing websites are. To this end, we use the number of special characters as a feature.

3) **Network Protocol:** Most phishing URLs use the HTTP protocol while some legitimate URLs use the HTTPS protocol. We use a binary value to indicate the protocol being adopted.

4) **Alex Ranking:** Most of the domain names of the legitimate websites can be found in the Alex ranking list. We use a binary value to show whether the domain name of an URL is in the list or not.

5) **Length of the Entire URL:** Phishing URLs usually are longer than legitimate URLs. We count the number of characters in a URL as one feature.

6) **Length of Host Name:** The lengths of host names of the phishing websites usually are longer than those of legitimate websites. Therefore, we use the number of characters in the host name as one feature.

7) **Length of Main Domain Name:** The main domain name of a phishing website is generally arbitrary, while the main domain name of a legitimate URL usually reflects its brand name (e.g., "baidu" is the brand and the main domain name of the host name "baidu.com"), and we use the number of charactors of the main domain name as a feature.

8) **Number of Dots in Host Name:** The dots in a URL divides the host name into sub-domains. Some phishers may use many sub-domains to make the URLs similar to the legitimate URLs. The number of dots in host name as a feature indicates the number of sub-domains.

9) **Number of Dots in URL Path:** Some phishing URLs may include the hosts of their targets in the URL paths to confuse users. The number of dots in URL path is used as a feature.

10) **URL Token Count:** The number of tokens in an URL.

11) **Host Name Token Count:** The number of tokens in the host name.

12) **Searching Result:** Phishing webpages usually keep alive in a short period of time or show low visibility, hence they usually cannot be found by search engines. We use a binary value to indicate whether the host name can be returned by search engines.

Furthermore, we extract the aforementioned features for the links (in URL form) in the webpages. According to the different domain names, we divide the links of webpages into four categories: *internal resource links*, *external resource links*, *internal webpage links*, and *external webpage links*. Resource links refer to those links direct to image or other type of files, and webpage links refer to those links direct to other webpages. Firstly, we calculate the percentage of the links using the network protocol of HTTPS in each of these four categories as a feature. Secondly, seven of the above basic features (referring to No. 5-11) are extracted for links in each of the four categories links and obtain four feature matrices, one for each of the four categories, respectively. The statistical features (i.e., mean, median, and variance) are calculated for each dimension of the feature matrices. Finally, the basic features of URLs, the ratio of HTTPS used, and the statistical features are concatenated as one feature vector.

B. Lexical Features of URLs and Links

Usually, the terms in URLs or links often reflect the brand names or characteristics of legitimate webpages while phishing webpages do not. For a given URL, we obtain 8 sets of keywords by extracting keywords from its host name, its path, the host names of external resource links in its webpage, the paths of external resource links in its webpage, the host names of external webpage links in its webpage, the paths of external webpage links in its webpage, the titles of its webpage, and its webpage content, respectively. For the keywords extracted from hostname, paths of links or title, we split it into terms and select the length longer than three keywords, respectively. For the keywords extracted from webpage content, we use the well-known TF-IDF scoring function. Furthermore, we conduct the pairwise intersection operation on these 8 keyword sets, achieving a number of 28 intersections. We count the number of elements in each pairwise intersection and obtain a 28-dimensional vector as the lexical feature of the given URL. Finally, we concatenate the lexical features and the feature vector obtained in Section III.A into one feature vector.

C. Phishing Website Detection

The features extracted in Sections III.A and III.B are merged into one vector, which can be used by any machine learning models to identify whether the URL is phishing or not. In our experiments, we investigate a number of machine learning models on phishing detection, including k-Nearest Neighbor, Logistic Regression, Random Forest, Decision Tree, Gradient Boosting Decision Tree, XGBoost, Deep Forest [6], etc., among which Deep Forest achieves competitive performance.

D. Phishing Target Detection

To mimic legitimate websites, phishing websites generally use part of the resources of their targeting websites directly, such as titles, links, image resources, css, making them similar to their targeting websites. The keywords in the title, the domain names of the external resource links and the external webpage links are significant and critical for finding phishing targets. Therefore, we propose to detect phishing targets by matching the keywords extracted from the phishing websites with the ones in the phishing target candidates. More specifically, we extract a few keywords from phishing websites, such as title, domain name, etc., which are used as query keywords for search engines (e.g., Google) to retrieve a number of phishing target candidates. In particular, we use the search operator supported by Google-like search engines, which allows a query with a specified tag-word indicating where to match the query keyword. For example, given a query keyword with a search operator "intitle" (i.e., matching the query keyword with words in the title), the retrieval results merely return the webpages whose titles contain the query keyword exactly, reducing the scope of the phishing target candidates.

In reality, some phishers may deliberately remove the page titles to avoid the anti-phishing detection tools that rely on the features extracted from webpage titles. Hence, we propose to use the domain names of external resource links and external webpage links as query keywords, considering the external resource links and the external webpage links may link to their phishing targets. Similarly, a query keyword can be combined with search operator "inurl" (i.e., domain name) to retrieve the webpages whose URL contains the query keyword.

Finally, the top-5 links from the results returned by the aforementioned search operators are selected as phishing target candidates. Furthermore, title similarity and text content similarity between the phishing webpage and the phishing target candidates are calculated. Specifically, we extract the keywords from title and text content, and then obtain keywords vectors based on the word frequency. Furthermore, cosine similarity between the phishing webpage and the phishing target candidates on title and text content can be calculated according to the keywords vectors, respectively. Finally, the phishing target of a phishing website is identified as its most similar candidate.

IV. EXPERIMENT

In this section, we introduce the dataset and experiment settings, and present the performance evaluation of the phishing detection and the target detection methods, respectively.

A. Dataset

The dataset which is released on the github[2] consists of 2,892 phishing URLs and 3,305 legitimate URLs, as shown in Table I, where the phishing ones are obtained from PhishTank[3] from January 8 to 9, 2018, while the legitimate ones are obtained from the top one million on Alex rankings[4] and a network security

challenge[5]. The distributions of URL lengths are shown in Fig.2. As we can see from the figure, the numbers of characters are between 20 and 100 for most of the URLs.

TABLE I. DATASET

	Training Set	Test Set
#Phishing URLs	1703	1189
#Legitimate URLs	2015	1290

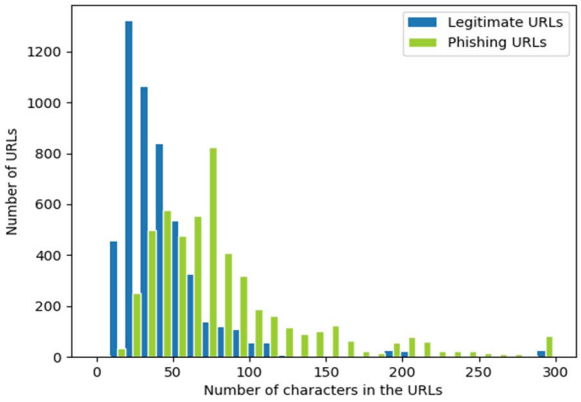

Fig. 2. Distribution of URL lengths

B. Performance Metrics

In our experiments, we use three metrics to evaluate the performance of the phishing detection approaches: false positive rate (*FPR*), true positive rate (*TPR*), and accuracy (*ACC*).

FPR in the following formula refers to the percentage of legitimate webpages that are misclassified to phishing webpages.

$$FPR = \frac{FP}{TN+FP} \quad (1)$$

where *FP* represents the number of legitimate webpages misclassified into phishing ones, and *TN* represents the number of legitimate webpages correctly classified as legitimate ones.

TPR refers to the percentage of phishing webpages that are correctly classified to phishing webpages.

$$TPR = \frac{TP}{FN+TP} \quad (2)$$

where *TP* represents the number of phishing webpages correctly classified as phishing webpages, and *FN* represents the number of phishing webpages that are misclassified to legitimate webpages.

ACC refers to the percentage of webpages that are correctly classified, including the phishing webpages correctly classified to phishing webpages and legitimate webpages are correctly classified to legitimate webpages.

$$ACC = \frac{TP+TN}{FN+TN+FP+TP} \quad (3)$$

[2]https://github.com/antiphish/deepforest
[3]http://www.phishtank.com
[4]http://stuffgate.com/stuff/website/
[5]https://www.kesci.com/apps/home/dataset/58f32a96a686fb29e425a567

C. Baselines

A number of popular phishing detection approaches have been included, which are specified as follows:

1) **Hybrid Phishing Detection Method (HPD):** Xiang et al. [19] proposed a method based on information extraction and information retrieval techniques. It used an identity-based component to detect phishing webpages by directly detecting the inconsistency between their identities and the identities they are imitating. The keywords-retrieval component utilized information retrieval algorithms via search engines to identify phishing websites.

2) **Tell-tale Lexical and Host-based Approach (TLH):** Ma et al. [20] proposed to detect phishing websites based on URL classification, using statistical methods to discover the tell-tale lexical and host-based properties of malicious URLs.

3) **Cantina** [8]**:** It is a content-based approach by using the TF-IDF algorithm to obtain keywords as unique signatures of webpages. Cantina infers the legitimacy of a webpage based on retrieving the signatures in search engines.

4) **Semantic Link Network (SLN):** Liu et al. [16] proposed to construct a semantic link network (SLN) of the suspicious webpages from the given suspicious webpages and their associated webpages for phishing target detection.

D. Comparison with Baseline Methods

The performance of the baselines and proposed approach are summarized in Table II. Note that the performance metrics of baselines are reported in their papers respectively, while they are tested on different datasets. The reasons are two-folded. On one hand, the datasets used by these approaches are unpublished and unreleased, make it impossible for us to test our approach on their datasets. On the other hand, some of the features used by the baselines rely on third-party services, which are difficult to obtain and some of them may be already inaccessible, making these methods hard to be tested on our dataset, too. In addition, TLH exploits URL features with no need to access webpage content, and the dataset is relatively large-scale. However, our method and some other baselines are content-based approaches, which have to access the content of webpages. As most of the phishing websites can only keep alive for quite a short period of time, even in a few seconds, it is hard to collect large-scale datasets consisting of both phishing URLs and their webpage content. In particular, HPD and Cantina exploit the content of the webpages to extract keyword information, which cannot deal with the webpages using only images. TLH relies on third-party services such as WHOIS, DNS MX record, etc., which may be affected by the performance of these services. SLN needs to analyze a large number of links and calculate the relations between the given webpages and links, which is computation-intensive. In contrast, our method achieves competitive performance by using statistical features and lexical features of URLs and links of webpages and identifies each website in no more than one second. Hence it can be used in practice.

TABLE II. COMPARISON OF THE PROPOSED METHOD WITH BASELINES (%)

	Test Set		FPR	TPR	ACC
	#Legitimate URLs	*#Phishing URLs*			
SLN	1000	1000	13.8	83.4	84.8
HPD	7906	3543	1.95	90.06	n.a.
TLH	15000	20500	0.1	n.a.	95.5
Cantina	100	100	6.0	97.0	96.9
Our method	1290	1189	**0.026**	**98.3**	**97.7**

E. Comparison with Existing Machine Learning Models

We summarize the performances of different machine learning algorithms on phishing detection, including k-Nearest Neighbor (KNN), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Gradient Boosting Decision Tree (GBDT), XGBoost (XGBST), and Deep Forest (DF). The performances of the approaches are shown in the Table III. Note that $N_{L\to L}$ and $N_{p\to P}$ represent the number of correctly classified webpages, i.e., the legitimate webpages classified as legitimate ones, and the phishing webpages classified as phishing ones, respectively, while $N_{L\to P}$ and $N_{p\to L}$ represent the number of misclassified webpages. From Table III, we can observe that the ensemble models, such as GBDT, XGBST, and DF achieve better performance than the single classification models, such as LR, RF, DT.

TABLE III. PERFORMANCE OF THE CLASSIFIERS

Algorithm	$N_{P\to L}$	$N_{P\to P}$	$N_{L\to P}$	$N_{L\to L}$	**FPR**	**TPR**	**ACC**
KNN	115	1074	72	1218	0.056	0.903	0.925
LR	71	1118	42	1248	0.033	0.940	0.954
RF	31	1158	69	1221	0.053	0.974	0.960
DT	53	1136	39	1251	0.030	0.955	0.963
GBDT	28	1161	45	1245	0.033	0.976	0.969
XGBST	23	1166	36	1254	0.030	0.981	0.971
DF	23	1166	32	1258	**0.026**	**0.983**	**0.977**

Considering the randomness of the ensemble models, such as GBDT, XGBST, and DF, we report the performance of these methods over ten separate runs. The results on different metrics are shown in Fig. 3, Fig. 4, and Fig. 5, from which we can observe that the performance is relatively stable, and DF tends to achieve a higher accuracy, a higher true positive rate, and a lower false positive rate.

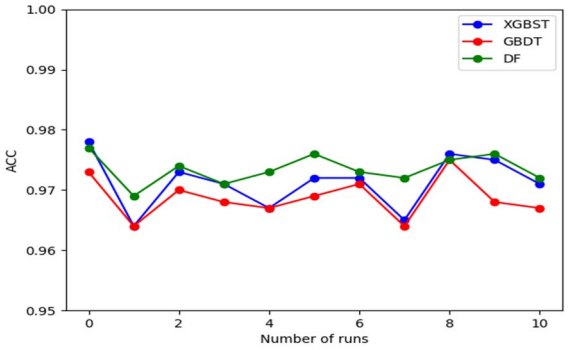

Fig. 3. Accuracy of different runs

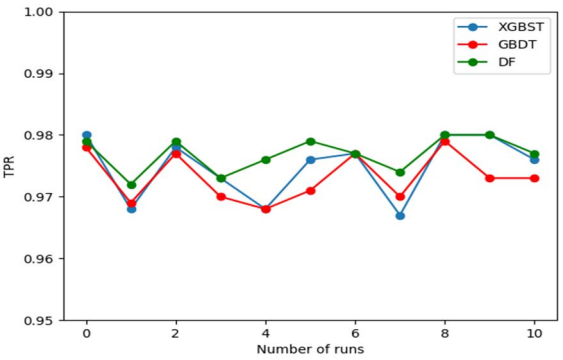

Fig. 4. True positive rate of different runs

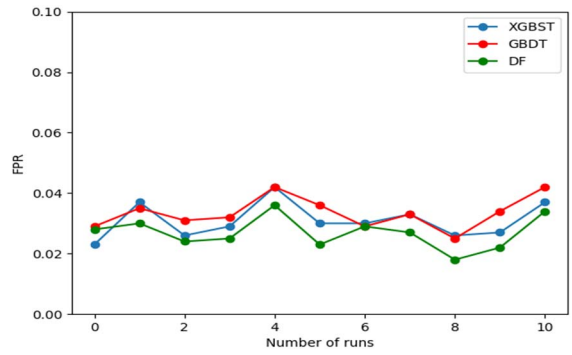

Fig. 5. False alarm rate of different runs

F. Evaluation on the Effectiveness of the Features

We evaluated the effectiveness of the combinations of the extracted features, i.e., a number of 12 raw basic features of URL (named as RAW) and statistical features of links, including mean (MEA), median (MED), and variance (VAR), and lexical features (LEX), etc. The performance of DF using different combinations of these features are shown in Fig. 6. From this figure, we can observe that the performance improves significantly by using more aforementioned features, indicating the effectiveness and significance of each of the extracted features.

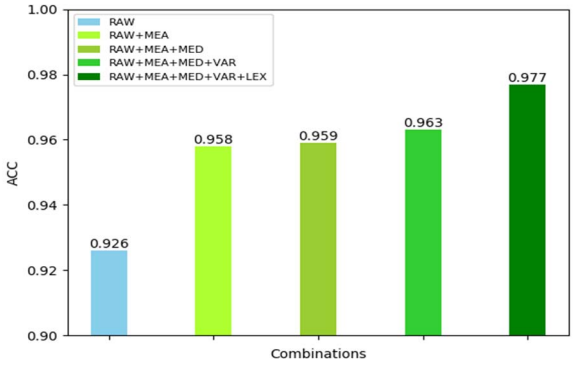

Fig. 6. Effectiveness of the features

G. Evaluation on Phishing Target Detection

To evaluate the performance of phishing target detection, we randomly selected 100 phishing URLs from PhishTank for evaluations. The proposed phishing target detection approach has found correct phishing targets for 78 phishing URLs, wrong targets for 5 phishing URLs, while the 17 remaining ones cannot be accessed. More specifically, the proposed search operator based strategy for phishing target detection achieves an accuracy of 93.98%. The experimental results indicate the effectiveness of the search operator based phishing target detection approach.

V. Conclusion

In this work, we proposed to combine URL and webpage link features for phishing website detection. The achieved features can be used by various classification algorithms, among which DF shows competitive performance. In particular, we extract features from URLs and the links in the first-level webpages, and do not access the content of the second-level webpages. Therefore, the proposed approach performs quickly in practice and achieves a high accuracy. In addition, we proposed a search operator based method for phishing target detection, which has also achieved a relatively high accuracy.

VI. Acknowledgments

This work is supported by the Guangdong Innovative Research Team Program (no. 2014ZT05G157), and the National Natural Science Foundation of China (No.61703109).

References

[1] "APWG: Phishing Activity Trends Report," 2017. [Online]. Available: http://docs.apwg.org/reports/apwg_trends_report_h1_2017.pdf

[2] L. Li, E. Berki, M. Helenius, and S. Ovaska, "Towards a contingency approach with whitelist-and blacklist-based anti-phishing applications: what do usability tests indicate?" in *Behaviour & Information Technology*, vol. 33, no.11, 2014, pp. 1136-1147.

[3] M. Dunlop, S. Groat, and D. Shelly, "Goldphish: Using images for content-based phishing analysis," in *Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on*, IEEE, 2010, pp. 123-128.

[4] W. Liu, X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," in *IEEE Internet Computing*, vol. 10, no. 2, 2006, pp. 58-65.

[5] L. Wenyin, G. Liu, B. Qiu, and X. Quan, "Antiphishing through phishing target discovery," in *IEEE Internet Computing*, vol. 16, no. 2, 2012, pp. 52-61.

[6] Z. H. Zhou, and J. Feng, "Deep forest: Towards an alternative to deep neural networks," *arXiv preprint*, arXiv:1702.08835, 2017.

[7] W. Zhang, Q. Jiang, L. Chen, and C. Li, "Two-stage ELM for phishing Web pages detection using hybrid features," in *World Wide Web*, vol. 20, no. 4, 2017, pp. 797-813.

[8] Y. Zhang, J. Hong, and L. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *ACM Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 639-648.

[9] S. Marchal, J. François, R. State, and T. Engel, "PhishScore: Hacking phishers' minds," in *Network and Service Management (CNSM)*, IEEE, 2014, pp. 46-54.

[10] J. James, L. Sandhya, and C. Thomas, "Detection of phishing URLs using machine learning techniques," in *Control Communication and Computing (ICCC)*, IEEE, 2013, pp. 304-309.

[11] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," in *IET Information Security*, vol. 8, no. 3, 2014, pp. 153-160.

[12] R. Srinivasa Rao and A. R. Pais, "Detecting Phishing Websites using Automation of Human Behavior," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, 2017, pp. 33-42.

[13] R. Verma and K. Dyer, "On the character of phishing URLs: Accurate and robust statistical learning classifiers," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015, pp. 111-122.

[14] R. Gowtham, I. Krishnamurthi, "A comprehensive and efficacious architecture for detecting phishing webpages," in *Computers & Security*, vol. 40, 2014, pp. 23-37.

[15] G. Ramesh, J. Gupta, and P. G. Gamya, "Identification of phishing webpages and its target domains by analyzing the feign relationship," in *Journal of Information Security and Applications*, vol. 35, 2017, pp. 75-84.

[16] L. Wenyin, N. Fang, X. Quan, B. Qiu, and G. Liu, "Discovering phishing target based on semantic link network," in *Future Generation Computer Systems*, vol. 26, no. 3, 2010, pp. 381-388.

[17] R. Patil, B. D. Dhamdhere, K. S. Dhonde, R. G. Chinchwade, and S. B. Mehetre, "A hybrid model to detect phishing-sites using clustering and Bayesian approach," in *Convergence of Technology (I2CT), 2014 International Conference for*, IEEE, 2014, pp. 1-5.

[18] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical feature based phishing URL detection using online learning," in *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*, ACM, 2010, pp. 54-60.

[19] G. Xiang, and J. I. Hong, "A hybrid phish detection approach by identity discovery and keywords retrieval," in *Proceedings of the 18th international conference on World wide web*, ACM, 2009, pp. 571-580.

[20] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 1245-1254.