Proceedings of the Federated Conference on Computer Science and Information Systems pp. 27–30
DOI: 10.15439/2020F140
ISSN 2300-5963 ACSIS, Vol. 21

Data Mining-Based Phishing Detection

Jan Bohacik
Department of Informatics, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia
Email: Jan.Bohacik@fri.uniza.sk

Ivan Skula
Department of Informatics, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia
Email: skula@dobraadresa.sk

Michal Zabovsky
University Science Park, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia
Email: Michal.Zabovsky@uniza.sk

***Abstract*—Webpages can be faked easily nowadays and as there are many internet users, it is not hard to find some becoming victims of them. Simultaneously, it is not uncommon these days that more and more activities such as banking and shopping are being moved to the internet, which may lead to huge financial losses. In this paper, a developed Chrome plugin for data mining-based detection of phishing webpages is described. The plugin is written in JavaScript and it uses a C4.5 decision tree model created on the basis of collected data with eight describing attributes. The usability of the model is validated with 10-fold cross-validation and the computation of sensitivity, specificity and overall accuracy. The achieved results of experiments are promising.**

I. Introduction

PHISHING is understood to be a criminal attack on obtaining personal information, such as passwords and payment card information, through webpages or e-mails [13]. Webpage creators can easily make fake pages which are virtually identical to the original ones, so people can easily fall victim to them. An alarming sign is the availability of guides about how to make fake web pages directly on the internet, e.g. [6]. At the same time, online payments are increasingly being used and many other activities are being moved to the internet. For example, the transaction value of digital payments is expected to show a growth rate of 17.0 percent between 2020 and 2024 [11]. The number of internet users has grown 1,187 percent since 2000 and there are 4,648,228,067 internet users at this moment, which is 59.6 percent of the world population [2]. Therefore, it is very important for internet users to be able to detect phishing webpages. This is recognized by the Anti-Phishing Working Group that reported 165,772 phishing sites detected in the first quarter of 2020 in its Phishing Activity Trends Report published on 11 May 2020 [1]. As it is outlined in this Report, a recent trend has been the use of the COVID-19 pandemic for phishing attacks. For example, a fake site claiming to be an official registration for the immediate withdrawal of money from a compensation fund of the Brazilian government was disseminated in Brazil via WhatsApp in the first quarter of 2020. According to [1], in the first quarter of 2020, the most targeted phishing sectors were SAAS/webmails (33.5 percent), financial institutions (19.4 percent), payments (13.3 percent), social media (8.3 percent), e-commerce/retail (6.2 percent), and others.

According to [8], there are anti-phishing approaches based on: a) heuristic; b) content; c) blacklist; d) knowledge discovery in data; and e) hybrid combination of several previously mentioned approaches. The most complex and potentially most effective is an approach based on knowledge discovery in data and its merger with other approaches in a hybrid combination. This paper is focused especially on the collection of phishing data, the data mining step of knowledge discovery and the creation of a Chrome plugin for data collection and phishing detection. The interest of academics in the data mining step for the purposes of phishing detection has been shown in several papers [5], [9], [14]. One of the most popular algorithms for the data mining step is the C4.5 algorithm creating an easily interpretable decision tree for classification [15], [7]. In the three referenced academic papers regarding the data mining step for the purposes of phishing detection, the results of decision trees were shown to be promising. The C4.5 algorithm uses training data consisting of instances (webpages) described by defined describing attributes and classified into the class attribute with possible values legitimate and phishing. It is a recursive algorithm which associates the available describing attribute with the highest normalized information gain at each node of the decision tree. That eventually leads to the splitting of the instances into subsets enriched in value legitimate or phishing. Each leaf node of the decision tree is associated with a possible value of the class attribute. The Chrome plugin is written in JavaScript and it contains a created decision tree for the performance of the detection. It is used for the collection of instances with a manual assignment of value legitimate or phishing on the basis of an expert inspection of the webpage.

The following organization of the paper is used. The data collected for the creation of the data mining-based phishing detection is described and analyzed in Section II. In Section III, the developed plugin detecting phishing webpages in the Chrome browser and its decision tree model made with the collected data are presented. The results achieved in employed experiments with 10-fold cross-validation are in Section IV. And finally, Section V concludes the paper.

II. Collected Data

The data characterized here contains descriptions of 1000 webpages visited through the Chrome browser with a developed plugin described in Section III. The plugin was used for saving data about these webpages and values legitimate or phishing were assigned to them manually on the basis of an expert inspection. Let us have a defined set $\boldsymbol{W}$ with 1000 webpages (instances). Let them be described by a defined set $\boldsymbol{B}$ with eight describing attributes and let them be classified into one class attribute D. The attributes in $\boldsymbol{B}$ and attribute D are presented in Table I. Describing attributes $\boldsymbol{B} = \{B_1; \ldots; B_k; \ldots; B_8\}$. If B_k is a numerical attribute and its value is v for a webpage $\boldsymbol{w} \in \boldsymbol{W}$, mark $B_k(\boldsymbol{w}) = v$ is used. Mark $B_k = \boldsymbol{P}$, B_k is a numerical attribute, $\boldsymbol{P}$ is a set of numerical values, means that $\boldsymbol{P}$ contains possible numerical values of B_k. If B_k is a categorical attribute and its categorical value is $b_{k,l}$ for a webpage $\boldsymbol{w} \in \boldsymbol{W}$, mark $B_k(\boldsymbol{w}) = b_{k,l}$ is used. Mark $B_k = \{b_{k,1}; \ldots; b_{k,l}; \ldots; b_{k,l_k}\}$ where B_k is a categorical attribute and $b_{k,1}, \ldots; b_{k,l}, \ldots, b_{k,l_k}$ are categorical values means $b_{k,1}, \ldots; b_{k,l}, \ldots, b_{k,l_k}$ are possible categorical values of categorical attribute B_k.

Table I.
Defined attributes

Attribute	Type of values	Possible values	Used units
AtInURL (B_1)	Categorical	*absent* ($b_{1,1}$)	N/A
		present ($b_{1,2}$)	
HyphenInURL (B_2)	Categorical	*absent* ($b_{2,1}$)	N/A
		present ($b_{2,2}$)	
SubdomainsInURL (B_3)	Numerical	1, 2, 3, ...	count
IPAddressInURL (B_4)	Categorical	*absent* ($b_{4,1}$)	N/A
		present ($b_{4,2}$)	
URLLength (B_5)	Numerical	4, 5, 6, ...	count
RatioOfLinksToOther Domains (B_6)	Numerical	[0;100]	%
RatioOfObjectsFrom OtherDomains (B_7)	Numerical	[0;100]	%
HTTPSProtocol (B_8)	Categorical	*trusted* ($b_{8,1}$)	N/A
		untrusted ($b_{8,2}$)	
ClassAttribute (D)	Categorical	*legitimate* (d_1)	N/A
		phishing (d_2)	

Attribute $B_1 = AtInURL = \{b_{1,1}; b_{1,2}\} = \{absent; present\}$ indicates if the URL address of some webpage $\boldsymbol{w}$ contains the @ symbol ($B_1(\boldsymbol{w}) = present$) or $\boldsymbol{w}$ does not contain it (i.e., $B_1(\boldsymbol{w}) = absent$). Normally, anything that is placed prior the @ symbol is ignored by the internet browser and redirection to what is typed after the @ symbol is performed. Attribute B_2 indicates if the URL address of some webpage $\boldsymbol{w}$ contains the '–' symbol ($B_2(\boldsymbol{w}) = present$) or $\boldsymbol{w}$ does not contain it ($B_2(\boldsymbol{w}) = absent$). This symbol may be used for creating a fake domain similar to the original one. *SubdomainsInURL* (B_3) contains the number of sub-domains in the URL address of the webpage. For example, fri.uniza.sk is a sub-domain of uniza.sk. $B_3 = \{1, 2, 3, ...\}$. Adding sub-domains to the URL address is another possible way for creating a fake domain and so a higher number of sub-domains is suspicious. Attribute B_4 indicates if the URL address contains an IP address. IP addresses may be used for hiding the real domains from the user. *URLLength* (B_5) contains the number of characters in the URL address. $B_5 = \{4, 5, 6, ...\}$. URL addresses with many characters may be used for hiding some information from the user. *RatioOfLinksToOtherDomains* (B_6) contains the ratio of links to other domains to all links in the webpage. $B_6 = [0;100]$. Too many links to other domains in the webpage might be indicative of a fake webpage. Attribute B_7 contains the ratio of objects such as images and videos from other domains to all objects in the webpage. $B_7 = [0;100]$. Similarly, too many objects from other domains might mean it is a fake webpage. *HTTPSProtocol* (B_8) indicates if some webpage $\boldsymbol{w}$ uses a trusted HTTPS protocol. Since HTTPS protocols are used for safe transfer of sensitive data, HTTPS protocols issued by unsound issuers or no HTTPS protocols at all are suspicious. Analysis of the collected data is provided in Table II.

Table II.
Analysis of the data collected with the Chrome plugin

Attribute	Particular value	Frequency of the value	Median	Mode
B_1	*absent* ($b_{1,1}$)	999	*absent*	*absent*
	present ($b_{1,2}$)	1		
B_2	*absent* ($b_{2,1}$)	874	*absent*	*absent*
	present ($b_{2,2}$)	126		
B_3	N/A	N/A	2	2
B_4	*absent* ($b_{4,1}$)	995	*absent*	*absent*
	present ($b_{4,2}$)	5		
B_5	N/A	N/A	46	22
B_6	N/A	N/A	95.24	100
B_7	N/A	N/A	100	100
B_8	*trusted* ($b_{8,1}$)	559	*trusted*	*trusted*
	untrusted ($b_{8,2}$)	441		
D	*legitimate* (d_1)	829	*legitimate*	*legitimate*
	phishing (d_2)	170		

III. Created Plugin and Decision Tree Model

Since no realistic data set for the creation of the decision tree model had been found, data about phishing and legitimate webpages were prepared first. Although there are some available data sets about webpages, they contain a very high percentage of data about phishing webpages, which is not the case in the real world and models created with this type of data might have issues. For example, the Phishing Websites Data Set from the UCI Repository of Machine Learning Databases [3] has only 40.5 percent

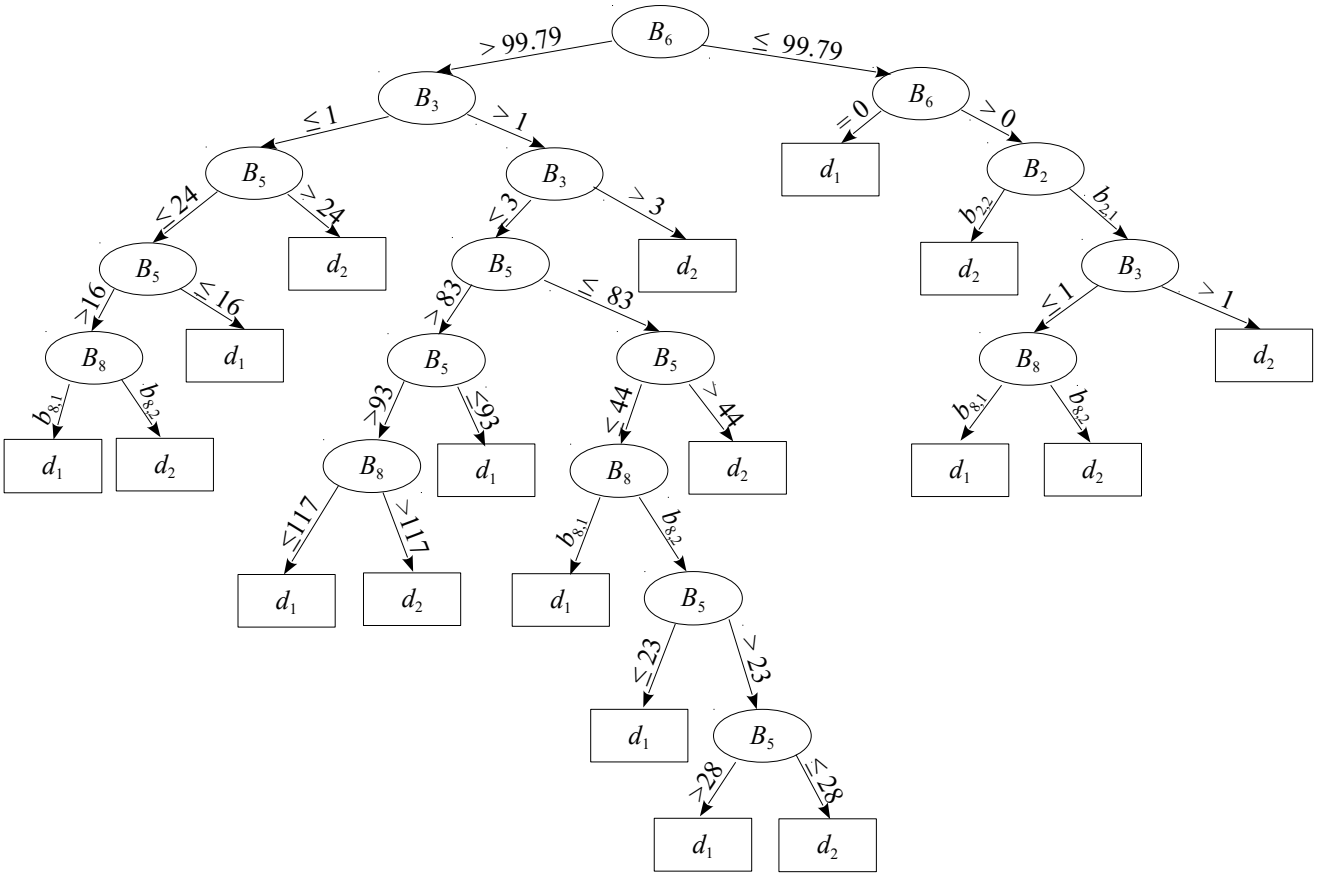

Fig 1. Decision tree model in the Chrome plugin

legitimate webpages. In addition, the functionality for the collection of data about a particular webpage is required in the Chrome plugin even after the decision tree model is created, as the collected data is used as the input of the model. The development of a Chrome plugin is similar to the development of a webpage because it consists of HTML, CSS and JavaScript codes hosted by the Chrome browser with the possibility to access some additional JavaScript APIs [10]. JavaScript APIs and a JavaScript code are used for the determination of the values for particular attributes in $\boldsymbol{B}$. The value for D is is set manually on the basis of an expert inspection. If the expert trusts some webpage $\boldsymbol{w}_1$, $D(\boldsymbol{w}_1)$ = *legitimate*, otherwise $D(\boldsymbol{w}_2)$ is set to *phishing*. Webpages $\boldsymbol{w} \in \boldsymbol{W}$ described by attributes in $\boldsymbol{B}$ and classified into D were given as the input to an implementation of the C4.5 algorithm in the Waikato Environment for Knowledge Analysis (Weka) [15]. The created decision tree shown in Fig. 1 is implemented into the plugin.

IV. Results of Employed Experiments

The results obtained in employed experiments with the C4.5 algorithm and with the collected data from Section II are described here first. It was important to see how the created decision tree model would perform potentially. All 1000 webpages from set $\boldsymbol{W}$ were loaded in Weka and then 10-fold cross-validation [4] was executed. In the validation, the real values for D and the detected values for D were compared and put into a confusion matrix [12] shown in Table III. There are 129 true positives, 27 false positives, 41 false negatives and 803 true negatives. Measures sensitivity, specificity and overall accuracy computed from the values in Table III are presented in Table IV. The achieved sensitivity is 0.7588, which means that 75.88 percent phishing websites were detected in the validation. The achieved specificity is 0.9675, which means that 3.25 percent websites generated false warnings about phishing activities in the validation. Finally, the achieved overall accuracy was 0.9320, which means that 6.80 percent of all websites were classified incorrectly. The results of 10-fold cross-validation show that the use of the decision tree model is promising. Several other data mining models were tried, but none of them gave significantly better results. In addition, it is simple to implement the decision tree with its use for detection in the Chrome plugin and its interpretability is high. Therefore, the decision tree shown in Fig. 1 was created on the basis of all webpages in $\boldsymbol{W}$ and implemented in the plugin. The plugin was tested on phishing and legitimate websites on the internet and the achieved results were similar to those in Table IV. The values of the describing attributes for three sample webpages are presented in Table V. When the correct leaf node for particular values of each website is found in the decision tree in Fig. 1, $\boldsymbol{w}_1$ is

phishing, $\boldsymbol{w}_2$ is *legitimate* and $\boldsymbol{w}_3$ is *phishing*. Some comprehensive analysis of the decision tree indicates that describing attributes *AtInURL* (B_1), *IPAddressInURL* (B_4), *RatioOfObjectsFromOtherDomains* (B_7) are not predictive when the combination of the other attributes from $\boldsymbol{W}$ is used. It is likely their unique use is not common nowadays.

TABLE III.
CONFUSION MATRIX AFTER 10-FOLD CROSS-VALIDATION

		Real	
		phishing	*legitimate*
Detected	*phishing*	129	27
	legitimate	41	803

TABLE IV.
MEASURES COMPUTED FROM THE CONFUSION MATRIX

Measure/Method	**Decision tree model**
Sensitivity	0.7588
Specificity	0.9675
Overall accuracy	0.9320

TABLE V.
SAMPLE OF DATA ABOUT WEBPAGES

Describing attribute	**Sample webpage**		
	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_3$
B_1	*present*	*absent*	*absent*
B_2	*absent*	*absent*	*absent*
B_3	5	1	2
B_4	*absent*	*absent*	*absent*
B_5	103	19	19
B_6	100	29	71
B_7	100	9	98
B_8	*untrusted*	*trusted*	*untrusted*

V. CONCLUSIONS

A Chrome plugin with a decision tree model for the detection of phishing webpages was described in the paper. The decision tree model was created with the C4.5 algorithm on the basis of collected data about 1000 legitimate and phishing webpages. It checks hyphens in URLs of webpages, sub-domains, lengths of URLs, links to other domains and HTTPS protocols. The results of using the C4.5 algorithm on the collected data in 10-fold cross-validation were promising with achieved sensitivity 0.7588, specificity 0.9675 and overall accuracy 0.9320. The use of the decision tree model in the Chrome plugin during browsing the internet led to similar values of the observed measures to the performed 10-fold cross-validation. The decision tree model suggests the use of the @ symbol in the URL address, IP address and objects from other domains in the webpage does not appear to be very predictive when the combination of hyphens, sub-domains, lengths of URLs, links to other domains and HTTPS protocols is checked. In the future, more describing attributes might be included and more webpages might be collected for the model.

ACKNOWLEDGMENT

The authors thank Peter Rovnanik for working on JavaScript implementations of the Chrome plugin, on collecting data and on conducting experiments under the supervision of the first author Jan Bohacik within his studies related to the preparation of the final thesis.

REFERENCES

[1] Anti-Phishing Working Group, *Phishing Activity Trends Report, 1st Quarter 2020*. USA: Anti-Phishing Working Group, 2020, https://docs.apwg.org/reports/apwg_trends_report_q1_2020.pdf.

[2] E. d. Argaez, *Internet Usage Statistics: The Internet Big Picture*, Bogota, Colombia: Internet World Stats, 2020, https://www.internetworldstats.com/stats.htm.

[3] D. Dua and C. Graff, *UCI Machine Learning Repository*, USA: University of California, School of Information and Computer Science, 2019, http://archive.ics.uci.edu/ml.

[4] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. : Springer-Verlag, 2009, https://dx.doi.org/10.1007/978-0-387-84858-7.

[5] M. Karabatak, T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," in *Proc. of the International Symposium on Digital Forensic and Security*, IEEE, Turkey, 2018, pp. 1-5, https://dx.doi.org/10.1109/ISDFS.2018. 8355357.

[6] B. M. Lawrence, *How to Make Fake Web Pages*. : Techwalla, 2020, https://www.techwalla.com/articles/how-to-make-fake-web-pages.

[7] K. Pancerz, V. Levashenko, E. Zaitseva, J. Gomuła, "Experiments with classification of MMPI profiles using fuzzy decision trees," in *Proc. of the Federated Conference on Computer Science and Information Systems*, IEEE, Poland, 2018, pp. 125-128, https://dx.doi.org/10.15439/2018F111.

[8] S. Patil, S. Dhage, "A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework," in *Proc. of the International Conference on Advanced Computing & Communication Systems*, IEEE, India, 2019, pp. 588-593, https://dx.doi.org/10.1109/ICACCS.2019.8728356.

[9] Y. Pristyanto, A. Dahlan, "Hybrid resampling for imbalanced class handling on web phishing classification dataset," in *Proc. of the International Conference on Information Technology, Information Systems and Electrical Engineering*, IEEE, Indonesia, 2019, pp. 401-406, https://dx.doi.org/10.1109/ICITISEE48480.2019.9003803.

[10] J. Sonmez, *How to Create a Chrome Extension in 10 Minutes Flat*. Australia: sitepoint, 2015, https://www.sitepoint.com/create-chrome-extension-10-minutes-flat/.

[11] Statista, *Digital Payments: Worldwide*. Germany: Statista, 2020, https://www.statista.com/outlook/296/100/digital-payments/worldwide.

[12] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy", *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997, https://dx.doi.org/10.1016/S0034-4257(97)00083-7.

[13] L. Wenyin, G. Huang, L. Xiaoyue, X. Deng, and Z. Min, "Phishing web page detection," in *Proc. of the International Conference on Document Analysis and Recognition*, IEEE, South Korea, 2005, pp. 560–564, https://dx.doi.org/10.1109/ICDAR. 2005.190.

[14] R. Wahyudi, H. Marcos, U. Hasanah, B. P. Hartato, T. Astuti, R. A. Prasetyo, "Algorithm evaluation for classification 'phishing website' using several classification algorithms", in *Proc. of the International Conference on Information Technology, Information Systems and Electrical Engineering*, IEEE, Indonesia, 2018, pp. 265-270, https://dx.doi.org/10.1109/ICITISEE.2018.8720975.

[15] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. USA: Morgan Kaufmann, 2017, https://dx.doi.org/10.1016/C2015-0-02071-8.