Phishing Detection: A Recent Intelligent Machine Learning Comparison based on Models Content and Features

Neda Abdelhamid
Information Technology
Auckland Institute of Studies
Auckland, New Zealand

Fadi Thabtah
Applied Business and Computing
Nelson Marlborough Institute of Technology
Auckland, New Zealand

Hussein Abdel-jaber
Faculty of Computer Studies
Department of IT and Computing
Arab Open University
Saudi Arabia

***Abstract*— In the last decade, numerous fake websites have been developed on the World Wide Web to mimic trusted websites, with the aim of stealing financial assets from users and organizations. This form of online attack is called phishing, and it has cost the online community and the various stakeholders hundreds of million Dollars. Therefore, effective counter measures that can accurately detect phishing are needed. Machine learning (ML) is a popular tool for data analysis and recently has shown promising results in combating phishing when contrasted with classic anti-phishing approaches, including awareness workshops, visualization and legal solutions. This article investigates ML techniques applicability to detect phishing attacks and describes their pros and cons. In particular, different types of ML techniques have been investigated to reveal the suitable options that can serve as anti-phishing tools. More importantly, we experimentally compare large numbers of ML techniques on real phishing datasets and with respect to different metrics. The purpose of the comparison is to reveal the advantages and disadvantages of ML predictive models and to show their actual performance when it comes to phishing attacks. The experimental results show that Covering approach models are more appropriate as anti-phishing solutions, especially for novice users, because of their simple yet effective knowledge bases in addition to their good phishing detection rate.**

Keywords— Computer Security; Phishing Detection; Machine Learning, Web Threat

I. INTRODUCTION

Phishing can be defined as developing websites that are fake and replicate trusted websites in order to deceive online users by illegally gaining access to their login information for the possibility of stealing their financial assets [5]. Traditionally, there are common approaches to combat phishing, such as legal, educational and awareness programmes [17]. Legislators in countries like the USA, UK, Canada and Australia have approved legislative bills that incriminate phishing, in which phishers can expect to face serious jail sentences. However, the legal actions are not highly effective in reducing phishing, since a phishing website has a short life span – normally about two days – which helps the phisher to disappear quickly once the fraud has been committed. On the other hand, while educating users may positively affect the global efforts of combating phishing, this approach demands high monetary costs besides necessitating users to be equipped with computer security knowledge [2].

To overcome the limitations of traditional anti-phishing approaches, computer security experts developed toolbar visualisation techniques, such as Ebay_Guard [10], Netscape [23], Netcraft [22], McAfee Site Advisor [16], Spoof Stick [28] among others. A security toolbar is usually embedded within a web browser and its role is to reveal certain security information to the end- user about potential online risks, such as phishing attacks. For instance, Spoof Stick toolbar displays the website's domain name and the Ebay_Guard toolbar shows an icon to the end-user. When this icon is green, this is an indication that it is safe to browse and, when the icon is red, this indicates that the website does not belong to Ebay or Paypal. Despite the wide spread of visualisation methods available, their phishing detection rate is low [17]. For instance, [33] contrasted three toolbars using security indicators and showed that all of them were unable to prevent phishing activities. The authors also showed that pop-up messages seems a more favourable approach to combat phishing than toolbars, since pop-up messages reveal additional security signs to the novice users.

Machine learning (ML) is an intelligent multidisciplinary approach primarily used in supervised learning to construct predictive models [29]. This approach seems appropriate to the problem of phishing detection, since this problem can be transformed into a typical classification task. A ML technique can build models based on labelled historical websites and then these models can be integrated into the browser to detect phishing activities. When the user browses a new web page, ML models guess the type of the website in real time and then communicate the outcome to the end-user. The key to success in developing automated anti-phishing ML models is the website's features in the input dataset and the availability of enough websites to create reliable predictive models.

978-1-5090-6727-5/17/$31.00 ©2017 IEEE

There have been several studies on the adaptation of ML approaches to minimise the risk of phishing, i.e. [1, 4, 6, 7, 11, 19, 20]. Most of these studies dealt with phishing as a binary classification problem and used one or more ML algorithms on a number of websites. Moreover, most of these studies contrasted a number of ML algorithms with respect to classification accuracy without paying high attention to the content of the predictive models generated. More importantly, there was too little attempt made on investigating the predictive models content to answer questions such as

- What are the key rules that can serve novice users in combating phishing?
- How the amount of knowledge might be useful for novice users in regards to making decisions?

In this paper, we critically analyse recent studies related to phishing in the research literature based on ML techniques. We show how these ML approaches derive the classification models and their advantages and disadvantages. More importantly, we investigate in-depth eight ML techniques on real datasets related to phishing and perform thorough comparisons of these techniques. The aim of the comparisons is to determine a suitable approach that may serve as an anti-phishing tool, based on the model content as well as the detection rate of phishing activities. In particular, the models' content of the type of ML called Covering approach is investigated to discover key rules that can empower end-users in combating phishing.

This paper is structured as follows: Section 2 presents phishing as a predictive task in ML. Sections 3 reviews common ML anti-phishing techniques, including Covering, neural networks, and associative classification among others. Section 4 is devoted to the experimental analysis of eight ML techniques and the phishing dataset. Finally, conclusions are given in Section 5.

II. Phishing as a Problem

ML techniques have proven to be powerful data analysis tools in many application domains such as medical diagnosis, market basket analysis, weather forecasting and events processing [3]. This is because ML techniques usually reveal concealed meaningful information from large datasets so they can be utilised in management decisions related to development, planning, and risk management [2]. ML can be seen as an automated and intelligent tool embedded within management information systems to guide decision-making processes in both business and scientific domains. Common tasks or problems that ML handles are clustering, association rule discovery, regression analysis, classification, pattern recognition, time series analysis, trends analysis, and multi-label classification [32].

One of the frequent tasks of ML is the forecasting of a target variable within datasets based on other available variables [31]. This forecasting process occurs in an automated manner using a classification model, normally named the classifier, which is derived from a labelled training dataset. The

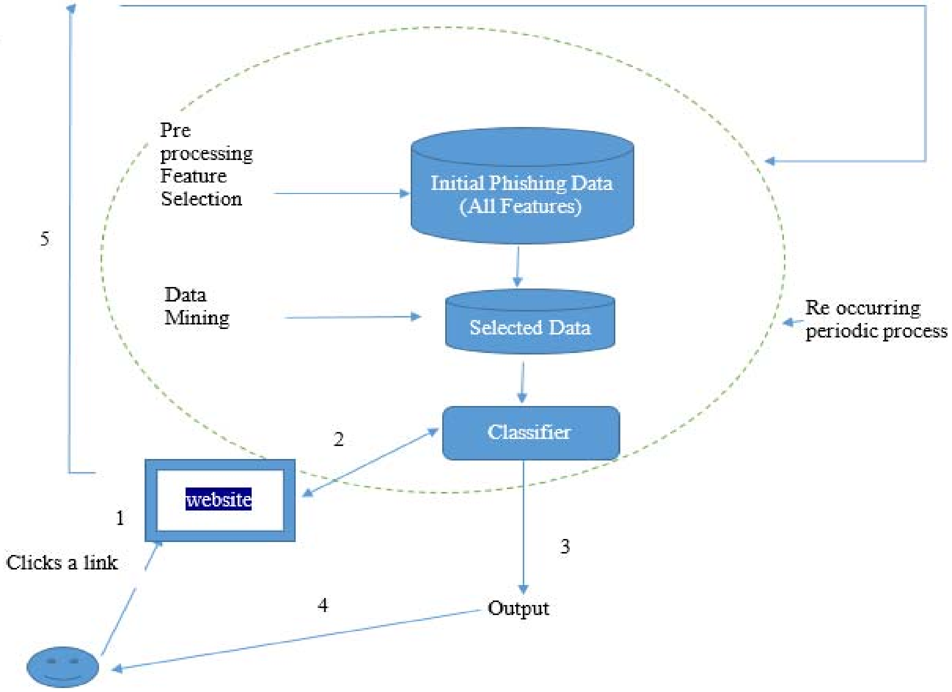

Fig. 1. Phishing as a classification process

goal of the classifier is to “guess” the value of the target variable in unseen data, referred to as the test dataset, as accurately as possible. This task description falls under the umbrella of supervised learning and is known as classification. [3] defined classification as the ability to “accurately” predict class attributes for a test instance using a predictive model derived from a training dataset.

Since the problem of website phishing involves automatic categorisation of websites into a predefined set of class values (legitimate, phishy) based on features (variables), this problem can be considered a classification task. To be more specific, the training dataset will consist of a set of predefined features and the class attribute and instances are basically the websites’ feature values. These instances can be extracted from different sources, such as Phishtank and online directories. The aim will be to build an anti-phishing classifier that can predict the type of website based on models derived from the training dataset during the data processing phase. Usually the reliability of the classifier is measured using accuracy, which primarily relies on the correlations of the features and the class [17]. Figure 1 shows phishing as a classification problem from the ML prospective.

III. LITERATURE REVIEW: COMMON ML ANTI-PHISHING TECHNIQUES

Websites can be categorised using sophisticated techniques in light of specific features such as , URL length, prefix_suffix, sub_domain, and so forth. [21] created distinctive learning bases utilising space understanding to recognise phishing sites and real sites. Lately, there have been different studies for acquiring automated rules to separate genuine and phishing sites utilising statistical analysis [2, 19, 26]. For example, [4] and [20] characterised various intelligently derived rules in light of different website features by using frequency counting of websites (instances) gathered from various sources, including Phishtank and Yahoo directory (Yahoo, 2011). Advancements in rules for decision making have been developed in [2] in which the authors utilised a computational intelligence method on a bigger phishing dataset gathered from numerous sources.

Phishing was explored using C4.5, decision tree, and other approaches which include Random Forest, Support Vector Machine and Naïve Bayes by [11]. (PILFER), "Phishing Identification by Learning on Features of Email Received", was developed as an anti-phishing method and then investigated on a set of 860 phishy and 695 ham cases. The results were different features for recognising instances as phishy or ham, i.e. IP URLs, time of space, HTML messages, number of associations inside the email, JavaScript and others. Hence, the authors explained that PILFER can improve the clustering of messages by joining all ten features found in the classifier beside " Spam filter output ".

In order to enhance both false negatives and false positives, an evaluation of Random Forest algorithm n against 2000 examples of messages was conducted in [6] . After experimentations with a fifteen feature dataset , the outcomes uncovered a reduction in error rate when utilising Random Forest and, thus, the utilisation of this method for phishing classification seemed fitting. Specifically, the models revealed to be more dominant over those of [11] with respect to detection rate.

Another attempt to accurately classify websites based on features was conducted in [4]. The authors manually categorised features into six criteria and then loaded them into an environment for analysis on WEKA [14]. During which various experiments ran using four classification algorithms against 1006 instances from Phishtank. The evaluation measure to determine the applicability of these features was the classification accuracy. The outcomes uncovered that decision tree algorithms detected on average, 83% of the phishing sites. Accordingly, the authors’ proposed that with appropriate pre-processing the results would improve.

Enhanced Dynamic Rule Induction (eDRI), is one of the first Covering algorithms that has been applied as an anti-phishing tool [30]. This Covering algorithm processes datasets by using two main thresholds, frequency and Rule Strength. eDRI scans the training dataset and only stores “strong” features if their frequency passes the minimum frequency threshold. Consequently, these features become part of the rule while all other values are removed during the initial scan. Once a rule is derived, eDRI removes its training instances and updates the strong features frequency to reflect the removal of its instances. Hence, eDRI somehow naturally prunes features and leads to a more controllable models. As part of the experiments, 11,000 websites were collected from multiple sources to evaluate eDRI’s reliability. In compassion with decision tree algorithm, the results obtained showed eDRI superiority to other Covering and decision tree approaches with respect to phishing detection rate.

A ML method that has been highly criticised due to the time involved in tuning its parameters is trial and error Neural Networks [21]. This method usually needs a domain expert available during the parameter tuning stage. A NN anti-phishing model by [31] proposed the elimination of trial and error and aimed for a more self-structuring classification. The authors designed the self-structured approach by updating several parameters, like the learning rate dynamically before adding a new neuron to the hidden layer. So, the process of updating the NN features is performed while building the classifier in the network environment. The purpose of applying the dynamic NN model was to detect phishy instances from a real dataset found in the UCI data repository [18] using different epoch sizes (100, 200, 500, 1000). The results revealed promising predictions when compared to Bayesian networks and decision trees.

Since phishers endlessly update their deceptive methods, [19] developed an anti- phishing NN model that relies on constantly improving the learnt predictive model, based on previous training experiences. The aim was to cope with the aggressive efforts by phishers that keep updating deceptive methods, so developing a self-structuring NN classification algorithm that deals with the vitality of phishing features was proposed by the authors. This self-structuring NN algorithm employs validation data to keep track on the performance of the built network model and involves appropriate intelligent decisions based on the outcomes acquired against the validation dataset. For example, when the attained error against the network is less than the minimum achieved error so far, the algorithm saves the networks' weights and continues the training process. On the other hand, when the achieved error is larger than the minimum achieved error so far, the algorithm continues the training process without saving the weights. Also if necessary, updates on other important network parameters occur during the construction of the classifier without having to wait until the model has been entirely built. As part of the experimentation on a number of features dataset revealed that the self-structuring NN model was able to generate highly predictive anti-phishing models compared to traditional classification approaches, such as C4.5 and probabilistic approaches.

IV. Data and Experimental Results

The dataset used in the experiments contains large numbers of websites and thirty features [20]. To be exact, the size of the security dataset includes more than 11000 examples, in which each example represents a website that can be either legitimate or phishy, and therefore the majority of the existing features are either binary or multi values, i.e. ternary. The available website's features in the dataset were extracted using a PHP script that was embedded in the web browser and applied on websites sources. The data instances have been collected from Phishtank [24] and Millersmiles [9] repositories. Before collecting the websites, and for each feature, a handcrafted rule was designed and then coded in PHP. For example, for the IP Address, a rule was designed that maps a website to phishy class when the IP address appears in the URL. Another rule examines the URL length and assigns a website phishy status when the URL length exceeds 70 digits. Further details on the complete description of features and their handcrafted rules can be found in [20].

In deriving the ML predictive models, the cross validation procedure has been adopted to compute the evaluation measures during constructing the classifiers. All experiments have been run on a Core i5 machine with a 3.1 GHz processor and 8.0 GB RAM. Moreover, Weka [14] ML tool was employed to run the experiments. Eight ML algorithms have been used to measure their effect on detecting phishing activities, namely Bayes Net [8], C4.5[27], SVM [25], AdaBoost [12], eDRI [30], OneRule [15], Conjunctive Rule[32], and RIDOR [13]. These ML algorithms have been selected because

1) They employ a variety of learning mechanisms

2) They are common algorithms that have been evaluated before on other applications data and showed good performance

3) They are implemented in Weka tool

The minimum frequency and rule strength thresholds for eDRI have been set in all experiments to 1% and 50% as suggested by its prospective authors. The key criteria used for comparison are

- Classification accuracy
- Content for the models

Figure 2 shows the classification accuracies generated by the predictive models against the phishing dataset. It is obvious from the figure that decision tree algorithms and eDRI produced the highest predictive models in regards to accuracy and both have outperformed the remaining classifiers. Particularly, eDRI achieved 0.83%, 4.79%, 4.79%, 0.69, 0.07%, 0.06% and 1.49% higher percentages of accuracy than Ridor, OneRule, Conjunctive Rule, Bayes Net, SVM-SMO and Ada Boost algorithms respectively. Only C4.5 algorithm outperformed eDRI on the phishing dataset with higher 2.20% and 3.09% respectively on the phishing dataset yet generated far more rules than eDRI (see Figure 3). The increase in the accuracy by C4.5 was because these algorithms utilise information gain metric to choose the root variable in a repetitive manner. Since we have several variables with ternary values in the input dataset, when decision tree algorithms evaluate these variables a branch is made and each path from the root node to any leaf denotes a rule. Hence, models generated by C4.5 on the phishing dataset are very large with respect to the numbers of rules derived when compared with the remaining classifiers, which may limit their use in real domain applications, including phishing. The fact that eDRI was able to achieve consistently higher accuracy than powerful classic algorithms (Naïve Bayes, SVM-SMO, Simple Logistic) as well as rule based classifiers (OneRule, Ridor, Conjunctive Rule, PRISM), is a clear evidence that this algorithm learning strategy has indeed improved the classifier's predictive power. In fact, eDRI ability to limit the use of the default class has increased the classifier's overall accuracy.

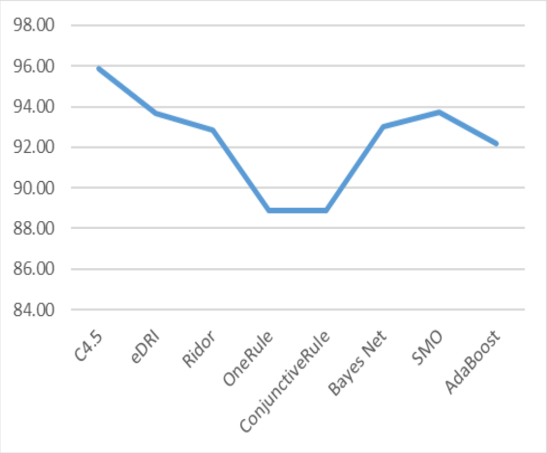

Fig. 2. Classification accuracy of the ML algorithms on the dataset

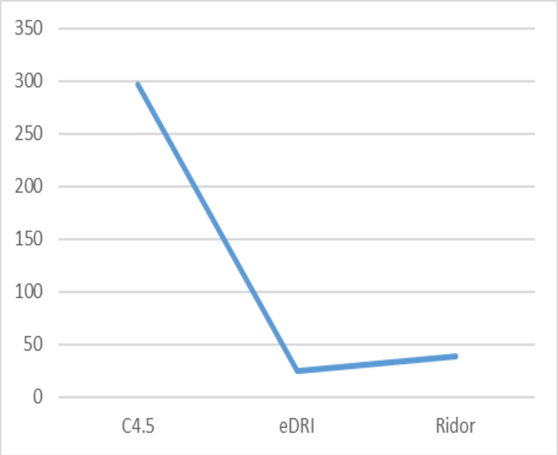

Fig. 3. Model size measured by the number of rules

We looked at the models' size resulting from the phishing dataset for C4.5, Ridor and eDRI algorithms. The results shown in Figure 3 illustrate that the eDRI algorithm substantially minimised the classifier size when contrasted with the considered rule-based algorithms. In fact, eDRI generated models with a lower number of rules than the remaining rule-based algorithms. For example, decision tree methods such as C4.5 generated 272 more rules than eDRI, without accomplishing any accuracy gain. Indeed, the majority of C4.5 rules cover a small portion of instances, which makes C4.5 models overfit the input dataset. Finally, OneRule and Conjunctive Rule algorithms seem to be inappropriate anti-phishing tools, since they produce very limited models with low phishing detection rates.

Ridor and eDRI algorithms have smaller models than decision trees which is an advantage for decision makers because a smaller number of rules are processed during decision making. These models work fine for applications such as medical diagnoses, where general practitioners can enjoy a concise set of rules for daily diagnoses of their patients. Overall, eDRI and Ridor consistently derived smaller classifiers than the rest of the classification algorithms and maintained steady predictive performance. Therefore, models that produce rule sets seem more suitable as anti-phishing tools for two primary reasons

1) They constantly produce high quality models in regards to accuracy
2) They provide rules sets that are simple to understand and manage by security experts. Thus, these sets of rules, when combined with human experience and knowledge, deliver more accurate decisions

Finally, phishing features were assessed based on the content of the models generated by C4.5, Ridor and eDRI to determine which were effective in detecting phishing activities. There are also two features in the top ranked rules within the models and these are:

- URL_Of_Anchor
- SSL_Final_State

For instance, these two features have been used to construct a rule that correctly cover 2041 instances by the Ridor algorithm. In addition, SSL_Final_State was part of 25 rules out of 39 rules of the Ridor model, i.e. 65% of the model's rules set used SSL_final_state feature. For the eDRI model, SSL_final_state feature was present in 21 out of 25 rules, which makes it the highest influential feature followed by URL_of_Anchor. These two features combined can replace the entire set of features of the dataset and, when processed, derive models with acceptable predictive accuracy. Specifically, when using only these two features the classification accuracy drops only 3% yet the data dimensionality as well as the models content become more manageable by the end-user.

V. Conclusions

There have been tremendous efforts to reduce phishing attacks by educating novice users, incriminating phishers and developing visualization techniques and toolbars that are integrated into web browsers. Despite the promising results of these approaches, they are still associated with low phishing detection rates, besides the costs associated with training users about phishing. Recently, a more effective approach to fight phishing that relies on machine learning techniques has emerged. In this approach, models extracted by a ML technique are used to classify websites either as legitimate or phishy, based on certain features. Thus in this paper, we deeply investigate a number of ML models on the hard problem of website phishing classification. The aim is to determine which ML method is more effective in detecting phishing attacks using a real dataset of 11,000 websites collected from Phishtank and other sources. To accomplish the aim, a large numbers of ML approaches (C4.5, OneRule, Conjunctive Rule, eDRI, RIDOR, Bayes Net, SVM, Boosting) have been contrasted with respect to different metrics, including predictive models' accuracies and rules. We have also taken features into consideration and its effect on the phishing detection rate. The experimental section demonstrates that the knowledge based approach presented by Ridor and eDRI algorithms seem to be an appropriate to combat phishing for two reasons

1) The classification accuracy of the models derived are highly competitive
2) The models contain easy to understand knowledge by novice users so they can utilise them in making decisions

Decision trees Bayes Net, and SVM achieved good detection rates. However, models extracted by decision trees showed very large amount of information which may overwhelm novice users and security experts, and thus will be hard to manage or understand. Moreover, Bayes Net and SVM showed good performance with respect to accuracy, yet their models are hard to understand by end-users.

In the near future, we intend to integrate a SVM within a web browser and conduct live experiments using large numbers of novice users in a pilot study.

References

[1] Abdehamid N. (2015) Multi-label rules for phishing classification. Applied Computing and Informatics 11 (1), 29-46.

[2] Abdelhamid N., Thabtah F., Ayesh A. (2014) Phishing detection based associative classification data mining. Expert systems with Applications Journal. 41 (2014) 5948–5959.

[3] Abdelhamid N., Thabtah F., (2014) Associative Classification Approaches: Review and Comparison. Journal of Information and Knowledge Management (JIKM). Vol. 13, No. 3 (2014) 1450027.

[4] Aburrous M.., Hossain M., Dahal K.P. and Thabtah F. (2010) Experimental Case Studies for Investigating E- Banking Phishing Techniques and Attack Strategies. Journal of Cognitive Computation, Springer Verlag, 2 (3): 242-253.

[5] Afroz, & Greenstadt, R. (2011) PhishZoo: Detecting Phishing Websites by Looking at Them. In Fifth International Conference on Semantic Computing (September 18- September 21). Palo Alto, California USA, 2011. IEEE.

[6] Akinyelu A. A. and Adewumi A. O. (2014) Classification of phishing email using random forest machine learning technique. Journal of Applied Mathematics, vol. 2014, Article ID 425731, 6 pages, 2014.

[7] Basnet R., Mukkamala S., Sung AH (2008) Detection of phishing attacks: A machine learning approach (2008) Soft Computing Applications Industry, pp. 373-383.

[8] Bouckaert, R. R., (2004). Bayesian network classifiers in Weka. (Working paper series. University of Waikato, Department of Computer Science. No. 14/2004). Hamilton, New Zealand: University of Waikato.

[9] Bright, M. (2011) MillerSmiles. [Online] Available at: http://www.millersmiles.co.uk/ [Accessed 09 January 2016].

[10] eBay Toolbar Guard (n. d.) [Online] Available at: http://pages.ebay.com/securitycenter/ [Accessed June 15th 2016].

[11] Fette I., Sadeh N., Tomasic A. (2007) Learning to detect phishing emails. Proceedings of the 16th international conference on World Wide Web. 649-656.

[12] Freund Y. and Schapire R. E. (1997) A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139, 1997.

[13] Gaines, B.R., Paul Compton, J. (1995) Induction of Ripple-Down Rules Applied to Modeling Large Databases, Intell. Inf. Syst. 5(3):211-228

[14] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. (2009) The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[15] Holte, R.C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning, 11, pp 63-90.

[16] McAfee, Inc. (n. d.) McAfee SiteAdvisor. [Online] Available at: http://www.siteadvisor.com/. [Accessed: January 11, 2016].

[17] Mohammad R., Thabtah F., McCluskey L., (2015A) Tutorial and critical analysis of phishing websites methods. Computer Science Review Journal. Volume 17, August 2015, Pages 1–24 Elsevier.

[18] Mohammad R., Thabtah F., McCluskey L. (2015B) Phishing websites dataset. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites Accessed January 2016.

[19] Mohammad R., Thabtah F., McCluskey L., (2014A) Predicting Phishing Websites based on Self-Structuring Neural Network. Journal of Neural Computing and Applications, 25 (2). pp. 443-458. ISSN 0941-0643. Springer.

[20] Mohammad R., Thabtah F., McCluskey L., (2014B) Intelligent Rule based Phishing Websites Classification. Journal of Information Security (2), 1-17. ISSN 17518709. IET.

[21] Mohammad, R. M., Thabtah, F. & McCluskey, L. (2013) Predicting Phishing Websites using Neural Network trained with Back-Propagation. Las Vigas, World Congress in Computer Science, Computer Engineering, and Applied Computing, pp. 682-686.

[22] Netcraft Inc. (n. d.) Netcraft Anti-Phishing Toolbar. [Online] Available at: http://toolbar.netcraft.com/. [Accessed May 9th 2016].

[23] Netscape Communications (n. d.) [Online] Available at: netscape-navigator.soft32.com. [Accessed May 8th 2016].

[24] PhishTank, 2011. PhishTank. http://www.phishtank.com/ [Accessed January 16 2016].

[25] Platt J. (1998) Fast training of SVM using sequential optimization, (Advances in kernel methods – support vector learning, B. Scholkopf, C. Burges, A. Smola eds), MIT Press, Cambridge, 1998, pp. 185-208

[26] Qabajeh I., Thabtah F., Chiclana F. (2015) Dynamic Classification Rules Data Mining Method. Journal of Management Analytics.Volume 2, Issue 3, pp. pages 233-253. Wiley.

[27] Quinlan, J. (1993) C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann.

[28] Spoof Stick Toolbar (n. d.) [Online] Available at: http:// Spoofstick.com. [Accessed May 18th 2016].

[29] Tan C.L., Chiew K.L., Sze S.N. (2017) Phishing Webpage Detection Using Weighted URL Tokens for Identity Keywords Retrieval. In: Ibrahim H., Iqbal S., Teoh S., Mustaffa M. (eds) 9th International Conference on Robotic, Vision, Signal Processing and Power Applications. Lecture Notes in Electrical Engineering, vol 398. Springer, Singapore

[30] Thabtah F., Qabajeh I.., Chiclana F. (2016A) Constrained dynamic rule induction learning. Expert Systems with Applications 63, 74-85.

[31] Thabtah F., Mohammad R., McCluskey L. (2016B) A Dynamic Self-Structuring Neural Network Model to Combat Phishing. In the Proceedings of the 2016 IEEE World Congress on Computational Intelligence. Vancover, Canada.

[32] Witten I. H. and Frank E. (2005). Data Mining: Practical Machine Learning Tools and Techniques.