Applying Machine Learning Techniques to Detect and Analyze Web Phishing Attacks

Alfredo Cuzzocrea
University of Trieste & ICAR-CNR
Trieste, Italy
alfredo.cuzzocrea@dia.units.it

Fabio Martinelli
IIT-CNR
Pisa, Italy
fabio.martinelli@iit.cnr.it

Francesco Mercaldo
IIT-CNR
Pisa, Italy
francesco.mercaldo@iit.cnr.it

ABSTRACT

Phishing is a technique aimed to imitate an official websites of any company such as banks, institutes, etc. The purpose of phishing is to theft private and sensitive credentials of users such as password, username or PIN. Phishing detection is a technique to deal with this kind of malicious activity. In this paper we propose a method able to discriminate between web pages aimed to perform phishing attacks and legitimate ones. We exploit state of the art machine learning algorithms in order to build models using indicators that are able to detect phishing activities.

ACM Reference Format:
Alfredo Cuzzocrea, Fabio Martinelli, and Francesco Mercaldo. 2018. Applying Machine Learning Techniques to Detect and Analyze Web Phishing Attacks. In *20th International Conference on Information Integration and Web-based Applications & Services (iiWAS '18), November 19–21, 2018, Yogyakarta, Indonesia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3282373.3282422

1 INTRODUCTION AND RELATED WORK

Phishing is a method to imitating a official websites or genuine websites of any organization such as banks, in- stitutes social networking websites, etc. Mainly phishing is attempted to theft private credentials of users such as username, passwords, PIN number or any credit card details etc. Phishing is attempted by trained hackers or attackers. Phishing is mostly attempted by phishy e-mails. This kind of Phishy e-mails may contains phishy or duplicate link of websites which is generated by attacker. By click- ing these kinds of links, it is redirected on malicious website and it is easily to theft your personal credentials. Phishing Detection is a technique to detecting a phishing activity. there are various methods are given by so many researchers . Among them Data Mining techniques are one of the most promising technique to detect phishing activity. Data mining is a new solution to detecting phishing issue. So data mining is a new research trend towards the detecting and preventing phishing website.

Websites can be categorised using sophisticated techniques in light of specific features such as , URL length, prefix_suffix, sub_domain, and so forth. Researchers in [14] 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 [1], [16] , [7]. For example, authors in [2] and [15] 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. Advancements in rules for decision making have been developed in [1] in which the authors utilised a computational intelligence method on a bigger phishing dataset gathered from numerous sources.

On the other hand, the general problem of detecting and analytzing web phishing attackas has also tight connections with advanced web applications and systems (e.g., [5, 6, 20]), as such attacks can also inject into the inner layers of underlying (web) frameworks, hence dealing with such aspects will play a relevant role in the near future.

Starting from these considerations and in order to overcome the performances obtained bu current literature, in this paper we propose a machine learning based method able to identify whether a web page is able to perform phishing activities.

2 DECISION-TREE ALGORITHMS FOR WEB PHISHING DETECTION

In this section we describe the method we propose for web phishing attacks detection.

Table 1 shows the features considered in the following study.

Variable	Feature
F1	URL Anchor
F2	Request URL
F3	Server form handler
F4	URL Length
F5	Having IP address
F6	Prefix/Suffix
F7	IP
F8	Sub Domain
F9	Website traffic
F10	Domain Age

Table 1: The feature set involved in the study

In order to collect data, we consider the PhishTank dataset [1]: PhishTank is a free community site where anyone can submit, verify, track and share phishing data. This dataset is in the form of .csv file format.

[1] https://archive.ics.uci.edu/ml/datasets/Website+Phishing

iiWAS '18, November 19–21, 2018, Yogyakarta, Indonesia

ACM ISBN 978-1-4503-6479-9/18/11…$15.00
https://doi.org/10.1145/3282373.3282422

The evaluation consists of two different stages: (i) we provide hypotheses testing, to verify whether the features vector exhibit different distributions for attacks and normal messages populations; and (ii) decision-tree machine learning analysis in order to assess if the eight features are able to discriminate between attacks and normal messages.

Machine learning is a type of artificial intelligence able to provide computers with the ability to learn without being explicitly programmed [13].

Machine learning tasks are typically classified into two categories, depending on the nature of the learning available to a learning system:

- *Supervised learning*: the computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. It represents the classification: the process of building a model of classes from a set of records that contains class labels.
- *Unsupervised learning*: no labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Figure 1 shows the supervised machine learning classification process adopted in the following paper.

The supervised learning approach is composed of two different steps:

(1) **Learning Step**: starting from the labeled dataset (i.e., where each feature is related to a class. In our case, the class is represented by the driver), we filter the data in order to obtain a feature vector. The feature vectors, belonging to all the web pages involved in the experiment with the associated labels, represent the input for the machine learning algorithm that is able to build a model from the analyzed data. The output of this step is the model obtained by the labeled dataset.

(2) **Prediction Step**: the output of this step is the classification of a feature vector belonging to the legitimate or to a phishing web page. Using the model built in the previous phase, we input this model using a feature vector without the label: the classifier will output with their label prediction (i.e., phishing web pages or legitimate one).

The algorithms considered are supervised decision tree-based i.e., they use a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the target of the items value (represented in the leaves). These algorithms (i.e., *J48*, *HoeffdingTree*, *RandomForest*, *RetTree*, *LMT* and *DecisionStump*) are the most widespread to solve data mining problems [13] for instance, from malware detection [3, 4, 10, 12] to pathologies classification [11].

The used algorithms are described in the following:

- *J48* [18]: it is the an open source Java implementation of the C4.5 decision tree algorithm. It is an statistical classifier based on id3 algorithm [8]. C4.5 generates a decision tree where each node splits the classes based on the gain of information. The attribute with the highest normalized information gain is used as the splitting criteria. Basically it computes the potential information for every considered attribute, given by a test on the attribute and the gain in information is calculated that would result from a test on the attribute. It works on the concept of information entropy. The training data is a set of classified samples having p-dimensional vectors defining the attributes of the sample;
- *HoeffdingTree* [19]: A Hoeffding tree (VFDT) is an incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams, assuming that the distribution generating examples does not change over time. Hoeffding trees exploit the fact that a small sample can often be enough to choose an optimal splitting attribute. This idea is supported mathematically by the Hoeffding bound, which quantifies the number of observations (in our case, examples) needed to estimate some statistics within a prescribed precision (in our case, the goodness of an attribute);
- *RandomForest* [17]: it operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set;
- *RepTree* [9]: it is considered a fast decision tree learner, it builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning. It only sorts values for numeric attributes once, while missing values are dealt with by splitting the corresponding instances into pieces (i.e., as in C4.5);
- *LMT* a logistic model tree (LMT) is a classification model with an associated supervised training algorithm that combines logistic regression (LR) and decision tree learning. Logistic model trees are based on the earlier idea of a model tree: a decision tree that has linear regression models at its leaves to provide a piecewise linear regression model (where ordinary decision trees with constants at their leaves would produce a piecewise constant model);
- *DecisionStump*: it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. Sometimes they are also called 1-rules. Depending on the type of the input feature, several variations are possible. For nominal features, one may build a stump which contains a leaf for each possible feature value or a stump with the two leaves, one of which corresponds to some chosen category, and the other leaf to all the other categories: for binary features these two schemes are identical.

We consider in this work five different machine learning algorithms In order to enforce the conclusion validity.

With regards to the hypotheses testing, the null hypothesis to be tested is:

H_0 : 'phishing and legitimate web pages exhibit similar values of the considered features'.

The null hypothesis was tested with Wald-Wolfowitz (with the p-level fixed to 0.05), Mann-Whitney (with the p-level fixed to 0.05) and with Kolmogorov-Smirnov Test (with the p-level fixed to

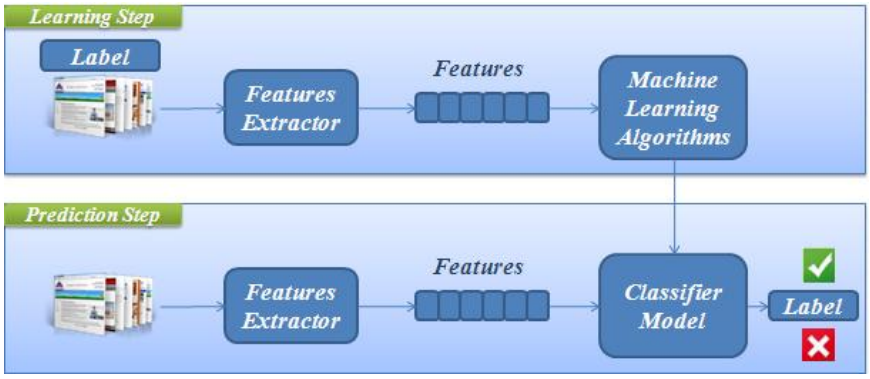

Figure 1: The Supervised learning approach adopted in this work with details about the Learning Step and the Prediction Step.

0.05). We chose to run three different tests in order to enforce the conclusion validity.

The purpose of these tests is to determine the level of significance, i.e., the risk (the probability) that erroneous conclusions be drawn: in our case, we set the significance level equal to .05, which means that we accept to make mistakes 5 times out of 100.

The analysis goal is to verify if the considered features are able to correctly discriminate between phishing and normal web pages.

These algorithms were applied to the full feature vector.

The classification analysis is performed using the Weka[2] tool, a suite of machine learning software, employed in data mining for scientific research.

3 EXPERIMENTAL ASSESSMENT AND ANALYSIS

3.1 Hypothesis Testing

The hypothesis testing aims at evaluating if the features present different distributions for the populations of phishing and normal web pages with statistical evidence.

We assume valid the results when the null hypothesis is rejected by the three tests performed.

Table 2 shows the results of hypothesis testing: the null hypothesis H_0 can be rejected for all the eight features. This means that there is statistical evidence that the feature vector is a potential candidate for correctly classifying between injected and normal messages.

This result will provide an evaluation of the risk to generalize the fact that the selected features produce values which belong to two different distributions (i.e., the one related of the four types of injected messages and the normal messages): the null hypothesis H_0 test confirms that the features can distinguish those observations. With the classification analysis we will be able to establish the effectiveness of the features in associating any feature vector to the phishing or to the legitimate web pages distribution.

3.2 Classification Analysis

We used five metrics in order to evaluate the results of the classification: Precision, Recall, F-Measure, MCC and RocArea.

[2]http://www.cs.waikato.ac.nz/ml/weka/

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$Precision = \frac{tp}{tp+fp}$$

where *tp* indicates the number of true positives and *fp* indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$Recall = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$F\text{-}Measure = 2 * \frac{Precision*Recall}{Precision+Recall}$$

MCC (i.e., the Matthews correlation coefficient) is a measure related to the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes:

$$MCC = \frac{tp*tn-fp*fn}{\sqrt{tp+fptp+fntn+fptn+fn}}$$

where *tn* is the number of true negatives.

The Roc Area is defined as the probability that a positive instance randomly chosen is classified above a negative randomly chosen.

The classification analysis consisted of building deep learning classifiers with the aim to evaluate the feature vector accuracy to distinguish between phishing and normal web pages.

For training the classifier, we defined T as a set of labeled messages (M, l), where each M is associated to a label $l \in \{IM, NM\}$.

Variable	Mann-Whitney	Kolmogorov-Smirnov	Wald-Wolfowitz	Test Result
F1	0,00000	$p < .001$	0,000	*passed*
F2	0,00000	$p < .001$	0,000	*passed*
F3	0,00024	$p < .001$	0,000	*passed*
F4	0,00000	$p < .001$	0,000	*passed*
F5	0,00000	$p < .001$	0,000	*passed*
F6	0,00000	$p < .001$	0,000	*passed*
F7	0,00000	$p < .001$	0,000	*passed*
F8	0,00000	$p < .001$	0,000	*passed*
F9	0,00000	$p < .001$	0,000	*passed*
F10	0,00000	$p < .001$	0,000	*passed*

Table 2: Results of the null hypothesis H_0 test.

For each M we built a feature vector $F \in R_y$, where y is the number of the features used in training phase ($y = 8$).

For the learning phase, we consider a k-fold cross-validation: the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset for testing the model, while the remaining $k-1$ subsets of the original dataset are used as training data. We repeated this process for $k = 10$ times; each one of the k subsets has been used once as the validation dataset. To obtain a single estimate, we computed the average of the k results from the folds.

We evaluated the effectiveness of the classification method with the following procedure:

(1) build a training set $T \subset D$;
(2) build a testing set $T' = D \div T$;
(3) run the training phase on T;
(4) apply the learned classifier to each element of T'.

Each classification was performed using 90% of the dataset as training dataset and 10% as testing dataset employing the full feature set.

The results that we obtained with this procedure are shown in table 3.

As shown in Table 1 the proposed method is able to obtain a precision equal to 0,923 and a recall equal to 0,916 in pishing attack detection using the J48 algorithm.

Figure 2 shows the trends related to the precision and the recall metrics for the classification algorithms involved in the evaluation.

The classification algorithms obtaining the best precision are J48 and RepTree, but considering also the recall metric, we highlight that the RepTree recall is lower if compared with the one obtained by the J48 classification algorithm: this is the reason why we confirm the J48 algorithm as the one obtaining the best performances in terms of precision and recall in order to detect web phishing attacks. As a matter of fact, the remaining algorithms (i.e., HoeffdingTree, RandomForest, LMT and DecisionStump) exhibit lower performances than J48 and RepTree in terms of precision and recall.

4 CONCLUSIONS AND FUTURE WORK

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.

In this paper we propose a method machine learning-based able to detect whether a web page exhibits pishing attacks. The proposed method is based on a feature vector easy to gather without require additional computation. In the evaluation we are able to obtain a precision equal to 0,923 and a recall equal to 0,916 in pishing attack detection using the J48 algorithm.

As future work, we plan to extend the proposed features in order to increase the detection accuracy, in addition we plna to apply formal methods with the aim to detect the code in which the malicious action happens.

ACKNOWLEDGMENTS

This work was partially supported by the H2020 EU funded project *NeCS* [GA #675320], by the H2020 EU funded project *C3ISP* [GA #700294].

REFERENCES

[1] N. Abdelhamid, A. Ayesh, and F. Thabtah. Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13):5948–5959, 2014.
[2] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah. Experimental case studies for investigating ebanking phishing techniques and attack strategies. *Cognitive Computation*, 2(3):242–253, 2010.
[3] P. Battista, F. Mercaldo, V. Nardone, A. Santone, and C. A. Visaggio. Identification of android malware families with model checking. In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISSP 2016, Rome, Italy, February 19-21, 2016.*, pages 542–547. SciTePress, 2016.
[4] G. Canfora, F. Mercaldo, C. A. Visaggio, and P. Di Notte. Metamorphic malware detection using code metrics. *Information Security Journal: A Global Perspective*, 23(3):57–67, 2014.
[5] M. Cannataro, A. Cuzzocrea, C. Mastroianni, R. Ortale, and A. Pugliese. Modeling adaptive hypermedia with an object-oriented approach and xml. In *In Second International Workshop on Web Dynamics*, 2002.
[6] A. Cuzzocrea. Combining multidimensional user models and knowledge representation and management techniques for making web services knowledge-aware. *Web Intelligence and Agent Systems*, 4(3):289–312, 2006.
[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.
[8] C. Jin, L. De-Lin, and M. Fen-Xiang. An improved id3 decision tree algorithm. In *Computer Science & Education, 2009. ICCSE'09. 4th International Conference on*, pages 127–130. IEEE, 2009.
[9] S. Kalmegh. Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news. *International Journal of Innovative Science, Engineering and Technology*, 2(2):438–46, 2015.
[10] F. Martinelli, F. Marulli, and F. Mercaldo. Evaluating convolutional neural network for effective mobile malware detection. *Procedia Computer Science*, 112:2372–2381, 2017.
[11] F. Mercaldo, V. Nardone, and A. Santone. Diabetes mellitus affected patients classification and diagnosis through machine learning techniques. *Procedia Computer Science*, 112(C):2519–2528, 2017.

Algorithm	Precision	Recall	F-Measure	MCC	Roc Area	Class
J48	0,904	0,892	0,898	0,829	0,958	legitimate
	0,923	0,916	0,919	0,833	0,958	phishing
HoeffdingTree	0,840	0,892	0,865	0,770	0,948	legitimate
	0,882	0,916	0,899	0,786	0,953	phishing
RandomForest	0,891	0,892	0,892	0,818	0,968	legitimate
	0,917	0,912	0,914	0,822	0,966	phishing
RepTree	0,856	0,911	0,882	0,799	0,964	legitimate
	0,933	0,872	0,901	0,804	0,961	phishing
LMT	0,876	0,892	0,884	0,804	0,970	legitimate
	0,922	0,905	0,913	0,821	0,972	phishing
DecisionStump	0,794	0,849	0,820	0,692	0,835	legitimate
	0,836	0,913	0,873	0,726	0,845	phishing

Table 3: Classification results.

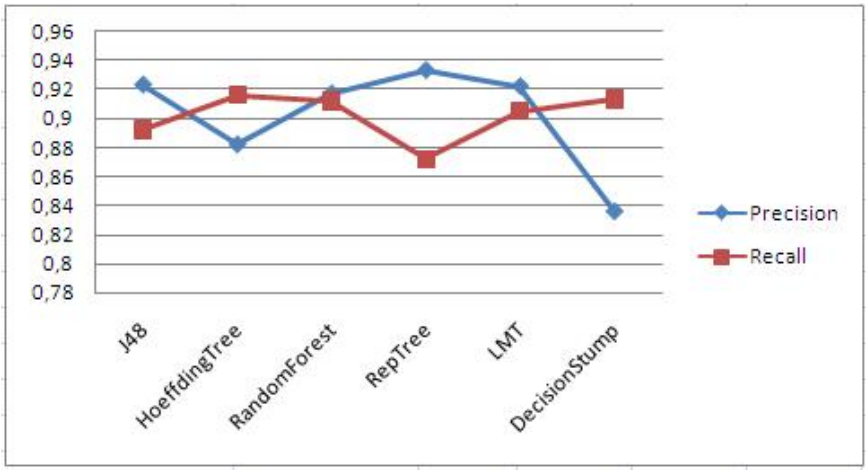

Figure 2: Precision and recall trends for the decision-tree classification algorithms involved in the experiment.

[12] F. Mercaldo, C. A. Visaggio, G. Canfora, and A. Cimitile. Mobile malware detection in the real world. In *Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on*, pages 744–746. IEEE, 2016.

[13] T. M. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.

[14] R. Mohammad, T. McCluskey, and F. A. Thabtah. Predicting phishing websites using neural network trained with backpropagation. In *World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2013.

[15] R. M. Mohammad, F. Thabtah, and L. McCluskey. Intelligent rulebased phishing websites classification. *IET Information Security*, 8(3):153–160, 2014.

[16] R. M. Mohammad, F. Thabtah, and L. McCluskey. Predicting phishing websites based on selfstructuring neural network. *Neural Computing and Applications*, 25(2):443–458, 2014.

[17] J. M. Pérez, J. Muguerza, O. Arbelaitz, I. Gurrutxaga, and J. I. Martín. Combining multiple class distribution modified subsamples in a single tree. *Pattern Recognition Letters*, 28(4):414–422, 2007.

[18] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[19] G. Webb. Decision tree grafting from the all-tests-but-one partition. San Francisco, CA, 1999. Morgan Kaufmann.

[20] Z. Wu, W. Yin, J. Cao, G. Xu, and A. Cuzzocrea. Community detection in multi-relational social networks. In *Web Information Systems Engineering - WISE 2013 - 14th International Conference*, pages 43–56, 2013.