Detecting Phishing Websites by using Neural Network Models

Dominika Zurawska[1]

[1]Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, POLAND

Abstract

In the article is presented the problem of classifying domains that may be phishing by using parameters and information extracted from sample pages. Presented tests are using various ML classifications models which we used from open libraries in selected programming language. Presented methods are implemented in simple way just to test selected models and compare them in standard metrics. To my tests i have selected neural networks, decision tree, svm, logistic regression and random forest. I have tested their effectiveness to select the best option for phishing.

Keywords

neural network, classification, phishing, security domain

1. Introduction

Machine learning methods are very popular in last years [1, 2, 3, 4]. In the development of It we can see that many applications use such methods to improve working toward some important aspects. In [5], [6], and [7] there are several application of neural networks in image processing. The model presented in [8, 9] show that neural networks are very good extractors of potential dangerous situation on the internet. Tests on classifiers for IoT environments show that both neural networks and fuzzy systems have very good application [10, 11].

Phishing attacks attempt to gain sensitive, confidential information such as usernames, passwords, credit card information, network credentials and more [12]. By posing as a legitimate individual or institution via phone or email, cyber attackers use social engineering to manipulate victims into performing specific actions—like clicking on a malicious link or attachment or willfully divulging confidential information. Both individuals and organizations are at risk; almost any kind of personal or organizational data can be valuable, whether it be to commit fraud or access an organization's network. In addition, some phishing scams can target organizational data in order to support espionage efforts or state-backed spying on opposition groups. Very interesting comments on this model can be found directly in online resources of https://www.antiphishing.org/resources/apwg-reports/.

To properly classify our domains, we decided to check and compare different classifiers to see if there are any significant differences between the results and which one is best suited to this problem. And also check if we can extract the features of a given domain that are most important in the classification of phishing.

ICYRIME 2021 @ International Conference of Yearly Reports on Informatics Mathematics and Engineering, online, July 9, 2021

domizur257@student.polsl.pl (D. Zurawska)

2. Phishing Websites Features

In this project, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. We classified our domain based on features such as: having IP Address, URL Length, Shortening Service, having At Symbol, double slash redirecting, Prefix Suffix, having Sub Domain, SSLfinal State, Domain registration length, Favicon, port, HTTPS token, Request URL, URL of Anchor, Links in tags, SFH, Submitting to email, Abnormal URL, Redirect, on mouseover, RightClick, pop up window, Iframe, age of a domain, DNSRecord, web traffic, Page Rank, Google Index, Links pointing to the page, Statistical report, Result.

3. Main decision parameters

The features that matter the most in the context of phishing websites detect.

3.1. SSL final State

The Subject Common Name of the certificate has to match the hostname of the phishing site that returned it. Some sites will return the hosting company's certificate when requested over HTTPS. As most modern browsers display warnings when a non-matching certificate is encountered, such certificates only serve to make the user more suspicious instead of increasing the perceived security of the site.

3.2. URL of Anchor

An anchor is an element defined by the <a> tag. This feature is treated exactly as "Request URL". However, for

this feature we examine:

1. If the <a> tags and the website have different domain names. This is similar to request URL feature.
2. If the anchor does not link to any webpage, e.g.:
   a) <a href="#">
   b) <a href="#content">
   c) <a href="#skip">
   d) <a href="JavaScript::void(0)">

Rule:
% of URL Of Anchor <31% → Legitimate
% of URL Of Anchor $\geq 31\%$ And $\leqslant 67\%$ → Suspicious
Otherwise→ Phishing

3.3. Links in tags

Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta> tags to offer metadata about the HTML document; <Script> tags to create a client side script; and <Link> tags to retrieve other web resources.

3.4. Prefix Suffix

The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example http://www.Confirme-paypal.com.

Rule:
Domain Name Part Includes (-) Symbol → Phishing
Otherwise → Legitimate

3.5. Web traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database. Furthermore, if the domain has no traffic or is not recognized by the Alexa database, it is classified as "Phishing". Otherwise, it is classified as "Suspicious".

Rule:
Website Rank<100,000 → Legitimate
Website Rank>100,000 →Suspicious
Otherwise → Phish

3.6. Links pointing to page

The number of links pointing to the webpage indicates its legitimacy level, even if some links are of the same domain. In our datasets and due to its short life span, we find that 98% of phishing dataset items have no links pointing to them. On the other hand, legitimate websites have at least 2 external links pointing to them.

Of Link Pointing to The Webpage = 0 → Phishing
Of Link Pointing to The Webpage > 0 and <= 2 →Suspicious
Otherwise → Legitimate

4. Classifications Algorithms

In this work some selected models were tested. Presented results are from open libraries that were available for student tests in online services. Data for the experiments were collected from https://archive.ics.uci.edu/ml/datasets.php.

4.1. Logistic regression

Logistic regression developing the concept of a perceptron using a nonlinear activation function and updating the weights with the logistic regression cost function,. In experiments i have used model from https://machinelearningmastery.com/logistic-regressionfor- machine-learning/. This model can be extended with regularization to prevent too high variance. Using the sigmoid activation function, model returns the probability of class, in our case we use the tanh activation function because it returns values from -1 to 1 and this is exactly how it is presented in our dataset.

4.2. SVM

SVM is very similar to logistic regression, but it uses a different method of determining the decision boundary, it consists in finding such a boundary whose distance to, samples of different classes, is as large as possible. Applied model is from https://paperswithcode. com/method-/svm. This algorithm also has the ability to correct variations with the help of the C parameter (expanded regularization) as well as solving problems with classes, linearly non-separable with the help of kernel functions, by increasing the dimensions and finding a hyperplane.

4.3. Decision tree

The next classifier is the decision tree [13], its activity is about creating a decision boundary by asking questions

about the data, answering them assigns the data to the next branches of the tree (there may be a lot of them, but in practice, it is usually divided into two sub-trees). In theory, such a tree can distribute data until there is only one class in each leaf, which means that the classifier will be over-trained and will not cope with the new data to prevent such a high variance when pruning the tree at a given height. In our case, the tree will work great because the features in our data set are binary ages and represent answers to the questions about domain metadata. The model of this section is from https://www.geeksforgeeks.org/decision-tree-implementation-python/.

4.4. Random forest

Random forest is the use of many decision trees and averaging their results, thanks to this solution we can use very tall trees and thus with a large variance (too high accuracy), because after averaging with other trees it ceases to be a problem and the model is efficient and accurate

4.5. Neural network

The multi-layer neural network, a model of a neural network with a layer of input neurons with an amount equal to the number of features of our data, layers of neurons which in our case is 50 and output neurons with the number of our classes

5. Accuracy measure parameter

After creating all classifications, it turns out that all models have proven themselves. However, they had different operating times and unfortunately for some real-time learning is not possible. But, for example, thanks to a decision tree, we can visualize the decision-making process and check based on the features it is made.

Formulas for determining parameters for model evaluation:

TP - true positive (the phishing sample is classified as phishing)
FN - false negative (the phishing sample is classified as non-phishing)
FP - false positive (the non-phishing sample is classified as phishing)
TN - true negative (the non-phishing sample is classified as non-phishing)

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{FN + PP} \tag{2}$$

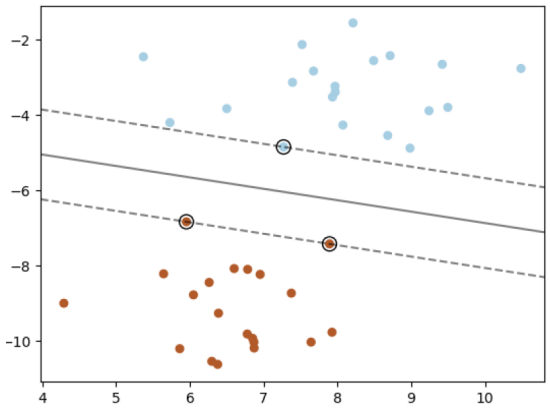

Figure 1: Results of SVM classification on input data.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

6. Results of tests

Our own implementation of logistic regression obtained a decent result of 0.9, while the error matrix shows that the model has a tendency to falsely classify websites as phishing: Accuracy: 0.9, Precision: 0.87, Recall: 0.974, F1: 0.919.

The SVM model with complementary variable and regularization be scored very high: Accuracy: 0.97, Precision: 0.97, Recall: 0.99, F1: 0.98.

Tree decision tree obtained very good results at a depth of about 15, increasing by higher values will not improve much, and reaching very high values caused that the accuracy was decreasing - it resulted from the previously discussed too large variance: Accuracy: 0.964, Precision: 0.966, Recall: 0.969, F1: 0.968.

Presentation of an example tree with a depth of 3 so that it is relatively clear, such a tree also achieves a satisfactory result of about 93

The Random Forest, achieves practically the same result as properly trimmed random tree: Accuracy: 0.965, Precision: 0.966, Recall: 0.969, F1: 0.968.

The multi-layer neural network obtained a very high result as one might expect: Accuracy: 0.975, Precision: 0.971, Recall: 0.984, F1: 0.977.

7. Conclusion

Most classifiers work well in predicting whether a given domain is a phishing attack, the effectiveness of prediction is at a very, level, and thanks to algorithms such as the decision tree, we are able to extract from the model

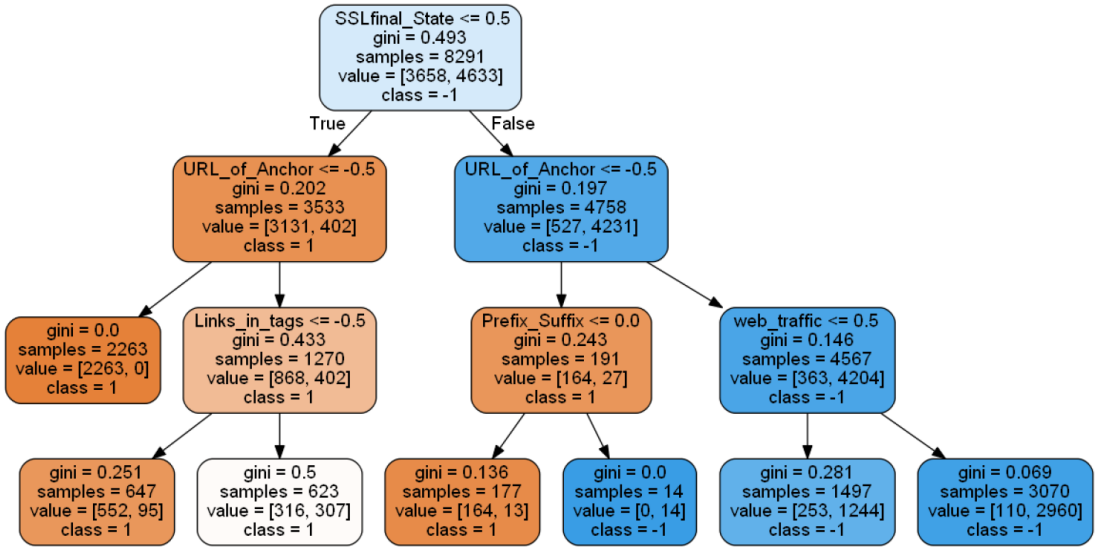

Figure 2: Decision tree diagram, created on the basis of our classifier, you can use it to see how the decision-making process is progressing. A precise description of the most important parameters according to which the decision was made in the fifth paragraph.

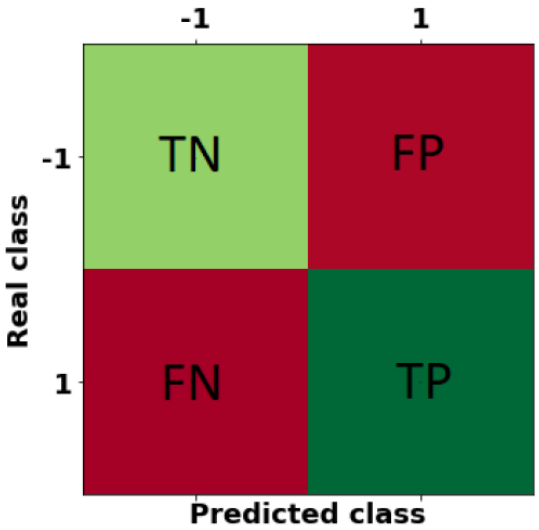

Figure 3: Sample error matrix of classification on input data. A pattern which was used in all further figures.

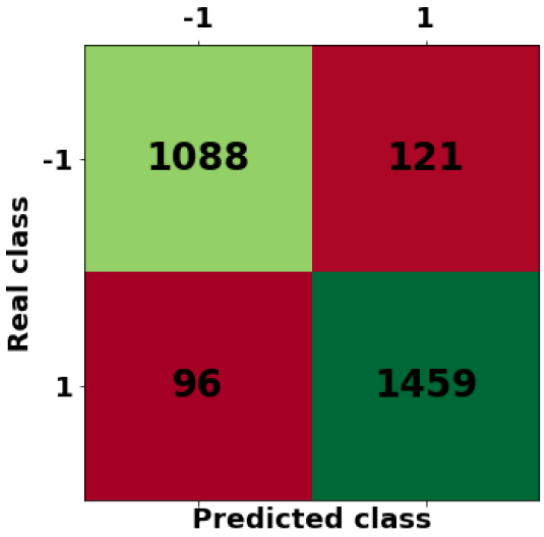

Figure 4: Sample error matrix of logistic classification on input data.

the features that are most important to recognize this type of attack, thanks to which you can defend yourself more effectively . And the learned model can be used in user protection programs. As the models obtained very similar results, making the choice appropriate to our needs should be based on the assessment of efficiency, flexibility for learning with new data and transparency of operation.

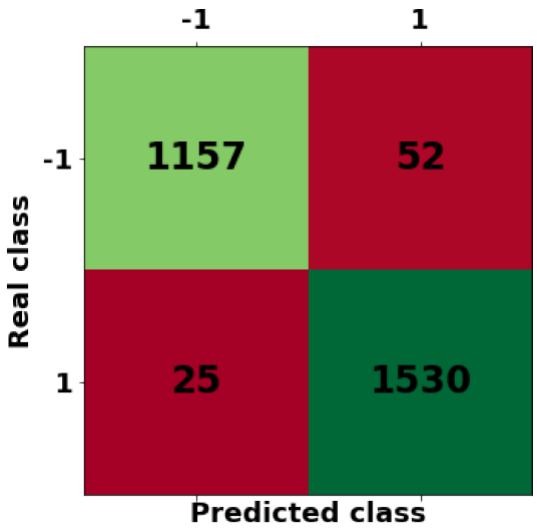

Figure 5: Sample error matrix of SVM classification on input data.

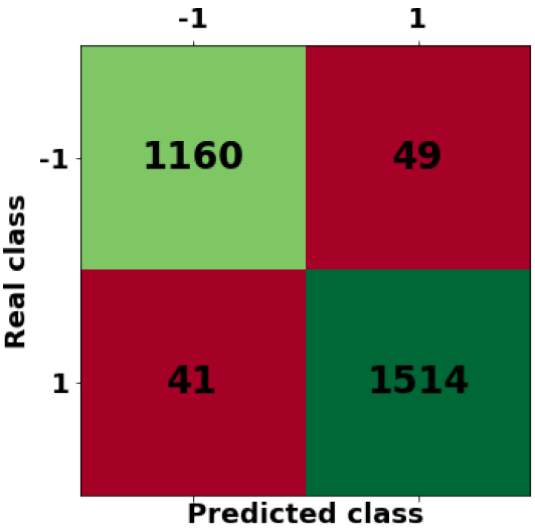

Figure 7: Sample error matrix of random forest classification on input data.

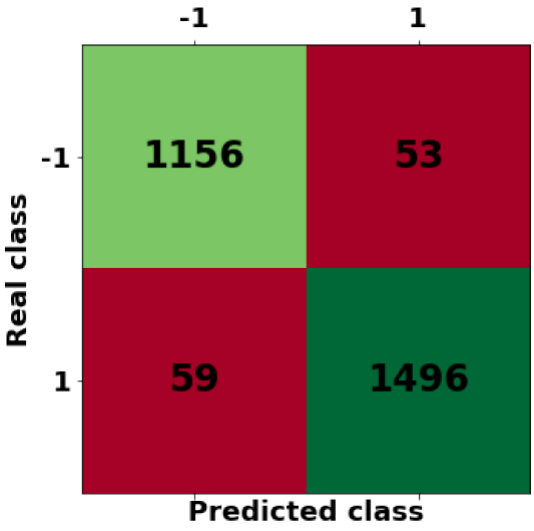

Figure 6: Sample error matrix of decision tree classification on input data.

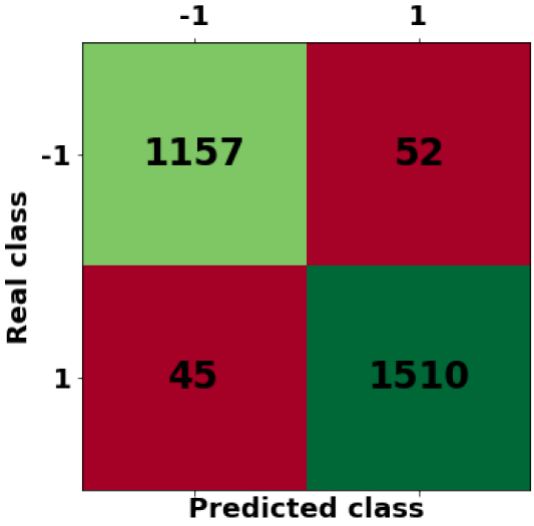

Figure 8: Sample error matrix of neural network classification on input data.

References

[1] G. Cardarilli, L. Nunzio, R. Fazzolari, D. Giardino, M. Matta, M. Patetta, M. Re, S. Spanò, Approximated computing for low power neural networks, Telkomnika (Telecommunication Computing Electronics and Control) 17 (2019) 1236–1241.

[2] G. Capizzi, G. Lo Sciuto, C. Napoli, M. Woźniak, G. Susi, A spiking neural network-based long-term prediction system for biogas production, Neural Networks 129 (2020) 271 – 279. doi:10.1016/j.neunet.2020.06.001.

[3] G. Capizzi, G. Lo Sciuto, C. Napoli, E. Tramontana, A multithread nested neural network architecture to model surface plasmon polaritons propagation,

Micromachines 7 (2016).
[4] G. Capizzi, C. Napoli, L. Paternò, An innovative hybrid neuro-wavelet method for reconstruction of missing data in astronomical photometric surveys, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7267 LNAI (2012) 21 – 29.
[5] R. Brociek, G. De Magistris, F. Cardia, F. Coppa, S. Russo, Contagion prevention of covid-19 by means of touch detection for retail stores, volume 3092, 2021, p. 89 – 94.
[6] D. Połap, M. Woźniak, Meta-heuristic as manager in federated learning approaches for image processing purposes, Applied Soft Computing 113 (2021) 107872.
[7] R. Avanzato, F. Beritelli, M. Russo, S. Russo, M. Vaccaro, Yolov3-based mask and face recognition algorithm for individual protection applications, volume 2768, 2020, p. 41 – 45.
[8] M. Wozniak, J. Silka, M. Wieczorek, M. Alrashoud, Recurrent neural network model for iot and networking malware threat detection, IEEE Transactions on Industrial Informatics 17 (2021) 5583–5594.
[9] X. Liu, S. Chen, L. Song, M. Woźniak, S. Liu, Self-attention negative feedback network for real-time image super-resolution, Journal of King Saud University-Computer and Information Sciences (2021).
[10] M. Woźniak, M. Wieczorek, J. Siłka, D. Połap, Body pose prediction based on motion sensor data and recurrent neural network, IEEE Transactions on Industrial Informatics 17 (2020) 2101–2111.
[11] M. Woźniak, A. Zielonka, A. Sikora, M. J. Piran, A. Alamri, 6g-enabled iot home environment control using fuzzy rules, IEEE Internet of Things Journal 8 (2020) 5442–5452.
[12] M. Silic, A. Back, The dark side of social networking sites: Understanding phishing risks, Computers in Human Behavior 60 (2016) 35–43.
[13] S. Russo, S. Illari, R. Avanzato, C. Napoli, Reducing the psychological burden of isolated oncological patients by means of decision trees, volume 2768, 2020, p. 46 – 53.