PHISHING DETECTION USING MACHINE LEARNING

By : Shubham Dutt

# Abstract

To create our phishing detection ML mode, datasets from UC Irvine’s Machine Learning repository were used for training, testing and evaluating the model to classify websites. Websites were classified based on a set of parameters. For instance, whether the website has an SSL certificate, uses HTTPs, amount of web traffic and it’s DNS record.

The model used was a decision tree classifier to assess parameters based on decisions taken and their relevance to make a final decision, whether a website it legitimate or phishing.

A decision tree classifier was chosen due to its success rate of 90.5% compared to other models.

# Introduction to phishing

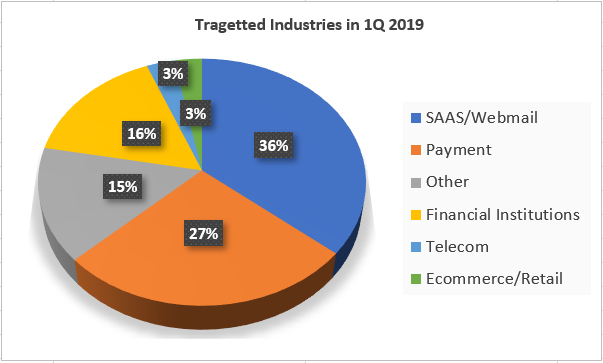
Phishing is a type of an identity-theft based cyberattack where attackers impersonate popular websites and email templates to manipulate customers into disclosing personal credentials. The main vulnerability in phishing lies in human’s ability to be deceived rather than the strength of a system’s defence.

Examples of phishing could be an email impersonating Amazon stating that your order has been cancelled, and asking you to login and view order status. Similarly, banks asking you to login to change certain details after an alleged incident.

Any credentials entered in these websites go straight to the attacker.

Such instances can occur at an unfortunate time, for example – receiving an Amazon-related phishing email right after ordering a package.

According to (APWG, 2019), a total of 791,766 unique phishing websites and 475,309 phishing emails were detected during the first quarter of 2019. The payments, SaaS/webmail industries were the most targeted.



## How are phishing weBsites detected ?

Common methods of detecting phishing websites include following a black-list containing phishing websites domain names and URLs for browsers to follow.

Manually inspecting the url for character differences, such as substituting a capital ‘i’ with a lower case “L”. Adding an extra character in the url name, for example – “Microsoftt” and manually inspecting website’s visuals.

[ML-based phishing detection](#_Implementation) classifies websites (and emails) after thorough training, testing and evaluation of [classification models](#_Implementation).

Phishtank, a database of phishing URLs, can be a go-to place to verify the authenticity of a website. Phishtank also acts as a platform for users to submit phishing URLs they have come across. The URLs are then verified to confirm they are actually phishing websites. Additionally, Phishtank also offers an API service to allow developers to incorporate its features in their applications.

## how can phishing be PREVENTED?

The most effective way to combat phishing is to address the central problem, which is the lack of user awareness. This can be accomplished by educating employees and users. A common strategy to test employee’s awareness is for a phishing website to be sent internal to the organization and checking who falls prey to it.

# IMPLEMENTATION – PHISHING DETECTION MODEL

Out of all the machine learning models available, phishing detection leverages classification-based models to differentiate/classify between a normal and phishing website.

Some examples of classification-based ML models include - logistic regression, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), decision trees and random forests.

For this use-case, decision tree classifiers are used because of its high success rate of 90.6%.

The first step is feature extraction. Features are a list of attributes extracted from websites and later analyzed for phishing detection based on certain criterias assigned to them. The exact list of features used in this research are –

|  |  |
| --- | --- |
| having\_IP\_Address | Domain\_Registeration\_Length |
| URL\_Length | Favicon |
| Shortening\_Service | Port |
| Having\_At\_Symbol | HTTPS\_Token |
| double\_Slash\_Redirecting | Request\_URL |
| Prefix\_Suffix | URL\_of\_Anchor |
| Having\_Sub\_Domain | Links\_in\_Tags |
| SSL\_Final\_State | SFH |
| Redirect | Submitting\_to\_Email |
| On\_Mouseover | Abnormal\_URL |
| RightClick | Age\_of\_Domain |
| PopUpWindow | DNS\_Record |
| Iframe | Web\_Traffic |
| Page\_Rank | Links\_Pointing\_To\_page |
| Google\_Index |  |

These features are originally extracted from 2456 websites, consisting of both phishing and legitimate websites. 2000 websites were used as the training data for detecting phishing websites. While the remaining 456 websites for testing the model.

Phishing websites are identified with a ‘-1’ label, while non-phishing websites are labelled with a ‘1’.

## Role of decision tree classifiers in phishing detection

Decision trees are a supervised machine learning model either used in regression or classification.

The primary goal of a decision tree is to start a decision process and narrow down to a final decision based on a series of decisions.

A decision tree starts with a root node, and consists of other nodes which each represent a decision on specific features from the dataset. A leaf node is the last node in the tree, representing a final decision.

In the phishing decision tree, the “SSL\_final\_state” feature is the root node since it has the highest information gain and lowest Gini impurity. Which means it has the highest chance of gaining meaningful information with the lowest chances of incorrect classifications, since a lower Gini impurity means purer splits.

The decision-making process starts by deciding whether the website has a SSL certificate. Based on which, other attributes are measured based on likelihood. For instance, if a website has a SSL certificate, it examines the URL\_of\_Anchor first, which is an attribute woth investigating in a suspicious website. Followed by web traffic, sub domain and DNS record. Since a phishing website will not have as much web-traffic, and will not have very prominent records.

Else it chooses the Web\_traffic node as the first entry to the path exploring a legitimate website. This logic is carried out throughout the tree, where each condition determines the path of the split.

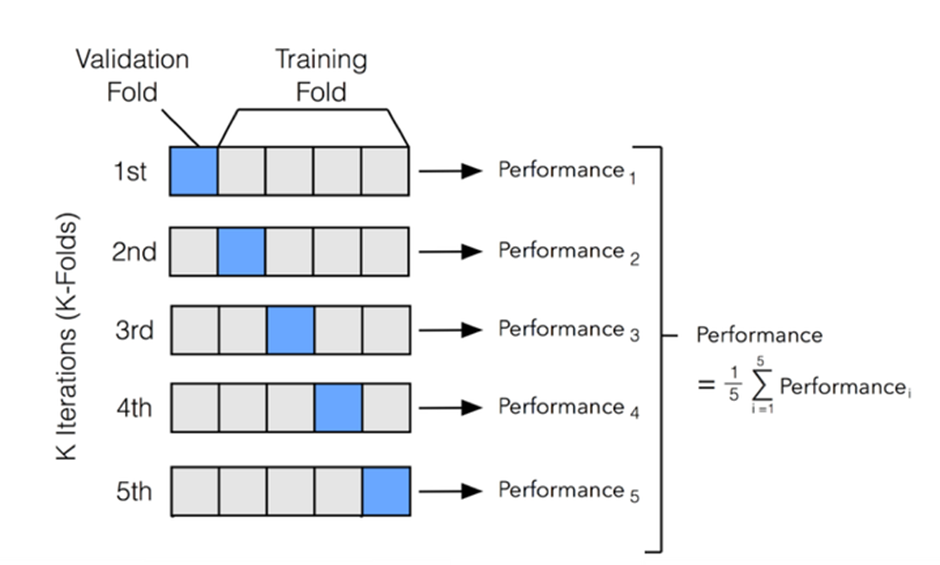
The Gini impurity of each node decreases as the split goes deeper, and reaches a value of “0.0” to indicate a pure node and it’s final decision about a website being phishing or legit.

The orange nodes indicate that a website is classified as phishing, while blue nodes indicate a regular safe website.

# Model evaluation

A technique called cross-validation is used to evaluate models. It evaluates the performance of a ML model on unseen data. It prevents relying solely on training data, else the model leads to overfitting – When the model performs really well with training data but not unseen data.

Each part of the unseen dataset is divided into subsets called “folds”. One fold out of the total is used as a validation set (subset of dataset that evaluates performance after training) while the rest are training datasets. This process is repeated with a different fold as the validation set each round. The results are then averaged to give an overview of the model’s performance.



[https://miro.medium.com/v2/resize:fit:940/1\*lOZqYqwmuW1lg6fitwqXxA.png](https://miro.medium.com/v2/resize:fit:940/1*lOZqYqwmuW1lg6fitwqXxA.png)

**Results -**

{'fit\_time': array([0.01200819, 0.0086987 , 0.00709391, 0.00710082, 0.00730062,

0.00698781, 0.00817919, 0.01255798, 0.00961423, 0.01052308]), 'score\_time': array([0.00367284, 0.00309205, 0.00304294, 0.00354981, 0.00299096,

0.00297928, 0.00375342, 0.00462461, 0.0043273 , 0.00460386]), 'test\_score': array([0.965, 0.945, 0.965, 0.975, 0.92 , 0.94 , 0.955, 0.945, 0.97 ,

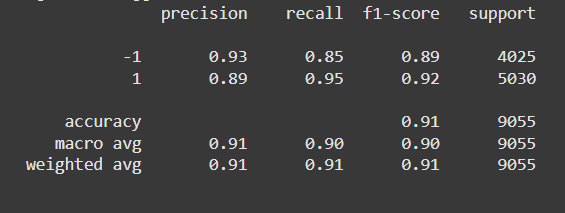
0.96 ])}

The fit\_time shows the time the model took to train in each of the 10 folds. Results of 0.003-0.013 seconds indicate a quick and efficient model training.

Score\_time is the time taken to evaluate the model on the validation test on each fold. Small time values indicate fast and efficient compute processing.

The test\_score is the evaluation score such as accuracy. The lowest score being 92% indicates a stable model.

**Classification Report -**



A classification report provides a details evaluation of the model’s performance in each class, and whether it performs better with one class over the other.

The ‘precision’ findings show that 93% of the predicted “-1” outcomes were correct, while 89% of the “1” outcomes were correct. High precisions means fewer false positives occurred.

The ‘recall’ results show better identification for ‘1’ value (phishing websites). High recall values indicate few false negatives were recorded.

F1-Score takes the average of precision and recalls. The support values show the number of true instances of each class in the dataset.

The overall model performed well by correctly predicted 91% of all instances.

References

<https://www.geeksforgeeks.org/cross-validation-machine-learning/>

<https://github.com/npapernot/phishing-detection/blob/master/README.md>

<https://www.statology.org/sklearn-classification-report/>

<https://scikit-learn.org/1.5/index.html>