Internship Report for June - July 2018

Yash Saboo

2015-2019 Batch

T.Y.C.S.

15070121170

**MORE DETAILS THE PROJECT**

**Problem Statement:**

To develop a Google Chrome Web Extension which can detect phishing/imitation websites and alert users about the same based on previously browsed websites.

**Technology Used and Technical Details:**

In the month of June, all my tasks involved implementation of machine learning techniques to predict if a website is Phishing or Legitimate.

Following concepts were used:

1. *Models:* DBSCAN (Clustering technique)
2. *Learning:* Semi-supervised
3. *Data:* Unbalanced
4. *Metrics:* Confusion Matrix

These technologies were learnt and implemented, as and when required, using various resources available on Internet.

**TASKS DONE**

**Task 1: Hyperparameter Tuning**

As mentioned in the previous report, I was supposed to perform hyperparameter tuning on the XGBoost Model. I did the tuning, but the results didn’t improve significantly. So, I moved on to a new model implementation as the AUC and accuracy (mentioned in the previous report) was sufficient enough to assert the proof of concept (POC) for that particular model.

**DBSCAN Clustering Technique to classify websites into Phishing or Legitimate**

As mentioned in the last report, I had to find threshold of Hamming Distance which would help in classifying Phishing and Legitimate websites using the algorithms designed for Layer 1 and 2, we used clustering technique to find it. It was implemented as semi-supervised learning. In semi-supervised learning, labeled data is used to help identify that there are specific groups of webpage types present in the data and what they might be. The clustering algorithm is then trained on unlabeled data to define the boundaries of those webpage types.

**Task 2: Data Collection and Processing for Clustering**

* As the data was already downloaded and processed from previous implementation of Layer 1 and 2, so this time more focus was on modelling aspects.
* Also, this time, the screenshots had the label which signifies as to which domain’s phishing it belonged to.

**Task 3: Training and Testing of the Model**

* Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. We used this technique to cluster similar images so that phishing of one domain can fall into one cluster.
* To cluster our data, we used density-based clustering. These models search the data space for areas of varied density of data points in the data space. It isolates various density regions and assign the data points within these regions in the same cluster.
* We used a very popular and efficient density-based clustering algorithm called density-based spatial clustering of applications with noise (DBSCAN).

To understand DBSCAN better, consider a set of points in some space to be clustered. For DBSCAN clustering, the points are classified as core points, (density-) reachable points and outliers, as follows:

* A point p is a core point if at least minPts points are within distance ε (ε is the maximum radius of the neighborhood from p) of it (including p). Those points are said to be directly reachable from p.
* A point q is directly reachable from p if point q is within distance ε from point p and p must be a core point.
* A point q is reachable from p if there is a path p1, ..., pn with p1 = p and pn = q, where each pi+1 is directly reachable from pi (all the points on the path must be core points, with the possible exception of q).
* All points not reachable from any other point are outliers.

Now if p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge", since they cannot be used to reach more points.

The inner details of training and testing cannot be mentioned due to confidentiality clause. I have mentioned the statistics though and they are as follows:

*Training:*

* minPts: 1
* ε: 25
* Number of Images: 1358
* Number of Training Images: 909
* Number of Testing Images: 449
* Number of Clusters: 242

*Testing:*

* Threshold: 25
* True Positive: 364
* True Negative: 61
* False Positive: 0
* False Negative: 85
* True Positive Rate: 81.07%
* False Positive Rate: 0.00 %

Thus, the optimum threshold of Hamming Distance for Layer 1 and 2 was found to be 25.

The documentation of all the tasks done, in last six months, were published on “Confluence” page of Symantec, which is basically Wiki of the corporation.

*This marked the end of my internship at Symantec (18th June, 2018). All these files are uploaded regularly on the Bitbucket of Symantec using Git. Thank You.*