Técnicas de Perceção de Redes
Network Awareness

Entity Profiling

(Clustering, Anomaly Detection, and Classification)

Python references: *SciPy* – SciPy.org, http://www.scipy.org/

NumPy – NumPy Routines, https://docs.scipy.org/doc/numpy/reference/routines.html

matplotlib.pyplot - http://matplotlib.org/api/pyplot_api.html

scikit-learn, Machine Learning in Python - http://scikit-learn.org/stable/

Data Analysis

The Base data files 'YouTube.dat', 'Browsing.dat', 'Mining.dat' contain the download and upload data (byte counts per 1 second interval) for YouTube video playing, user web browsing, and one active crypto-miner data streams, respectively. These will be three distinct traffic classes.

Note: The traffic data streams were captured with a single application running at a time, and capturing all traffic from and to a specific machine using a specific port: (TCP 443 for YouTube using automatic video playing at 720p), (TCP 443 and 80 for browsing multiple web pages with Firefox), and (TCP port 3357 for the miner).

Use the provided script 'baseProfileClass.py' as base code for the following tasks.

Feature Extraction (time-independent and time-dependent)

2 Use the provided script 'basePktFeaturesSilenceExt.py' to extract the mean, median and standard deviation of the upload/download packet/byte counts over time as time-independent features, and also the silence periods features (number of periods, mean and standard-deviation of the duration) as 6 additional (time dependent) features.

```
python basePktFeaturesSilenceExt.py -i YouTube.dat -m 2 -w 300 -s 20
python basePktFeaturesSilenceExt.py -i Browsing.dat -m 2 -w 300 -s 20
python basePktFeaturesSilenceExt.py -i Mining.dat -m 2 -w 300 -s 20
```

Rename the output files, respectively, to 'YouTubeAllF.dat', 'BrowsingAllF.dat', and 'MiningAllF.dat'.

Make 2-D plots (or log-log plots) using different pairs of features. Try different feature combinations, and try to find find possible discriminators between applications.

Note: you may consider alternative silence period thresholds (e.g., 256 bytes, ~4 small packets).

Feature Datasets for Training and Test

3. From the extracted features construct three different datasets of features (and respective class label vectors):

(i) A training dataset for anomaly detection, with the first 50% of the observations, from the inferred features using only YouTube and Browsing test datasets (Mining traffic will be the anomaly);

(ii) A training dataset for traffic classification, with the first 50% of the observations, from the inferred features using all three classes of test datasets;

(iii) A test dataset for anomaly detection and traffic classification, with the last 50% of the observations, from the inferred features using all three classes of train datasets.

Clustering

4. Apply the K-Means clustering algorithm to the 3 classes train dataset, assuming that you know that data can be divided in 3 clusters.

>> Analyze the labels of each observation and compared with the known label of that observation.

>> Test the algorithm with a higher/lower number of clusters.

>> Test the algorithm applying normalizing the input values (e.g., Standard Scaler). Analyzed the relative results.

>> Propose a classification algorithm for the test dataset, assuming the previous knowledge of the training dataset observation labels.

5. Apply the DBSCAN clustering algorithm to the 3 classes train dataset, you must assume that the number of classes are unknown. **Data inputs must be normalized.** Assume that each cluster must have at least 10 observations (`min_samples` argument), and a maximum distance between two samples to be considered as belonging to same cluster of 0.5. (`eps` parameter).

>> Analyze the labels of each observation and compared with the known label of that observation.

>> Test the algorithm with a higher/lower `eps` values.

>> Test the algorithm without applying normalizing the input values (e.g., Standard Scaler). Explain the poor results.

>> Propose a classification algorithm for the test dataset, assuming the previous knowledge of the training dataset observation labels.

6. Test alternative clustering algorithms. See: https://scikit-learn.org/stable/modules/clustering.html .

Anomaly Detection (Statistical Analysis)

7. Consider Browsing and YouTube as licit traffic and Crypto-Minning traffic as an anomaly. Using all features from the two classes train dataset (only YouTube and Browising data), calculate the centroid of each class of licit traffic. Using the tree class test dataset calculate for each observation calculate the Euclidean (relative) distance to each centroid. Identify abnormal observations of the test dataset based on the distances above a specific threshold to the licit classes centroids (consider first a value of 10).

>> Analyze the labels of each test observation and compared with the known label of that observation.

>> Test the algorithm with higher/lower values for the anomaly threshold.

>> Test the algorithm applying normalized input values (e.g., Max Abs Scaler or Standard Scaler). Explain the obtained results.

Anomaly Detection (Machine Learning with OneClass SVM)

8. Considering again only the two classes train dataset (with only YouTube and Browising data), and using one-class support vector machines (OneClass-SVM) with linear, RBF and, polynomial (degree 2) kernels develop a simple anomaly identifier for the test dataset.

>> Analyze the labels of each test observation and compared with the known label of that observation.

>> Test the model with different mu values.

>> Test the algorithm applying normalized input values (e.g., Max Abs Scaler or Standard Scaler). Explain the obtained results.

>> Proposed a decision process based on an ensemble methodology.

9. Test alternative anomaly/outlier detection algorithms: https://scikit-learn.org/stable/modules/outlier_detection.html .

Namely try, Local Outlier Factor (LOF) and Isolation Forest (IF).

Classification (Machine Learning)

10. Using the features from the three classes train dataset (with all three classes of data), and using support vector machines (SVM) with linear, RBF and, polynomial (degree 2) kernels develop a simple classifier for the test dataset.

>> Analyze the labels of each test observation and compared with the known label of that observation.

>> Test the algorithm applying normalized input values (e.g., Max Abs Scaler or Standard Scaler). Explain the obtained results.

>> Proposed a decision process based on an ensemble methodology.

11. Using Neural Networks develop a simple classifier for the test dataset traffic that outputs the class identifier (from all inferred features or PCA reduced features, both **normalized**). Start with a single 20 nodes hidden layer, and test other hidden layer sizes.

>> Analyze the labels of each test observation and compared with the known label of that observation.

>> Test the model with different sizes of the hidden layer of the neural network.

>> Test the algorithm without applying normalizing the input values. Explain the results.

>> Proposed a decision process based on an ensemble methodology.

12. Test alternative classification (supervised learning) algorithms:

https://scikit-learn.org/stable/supervised_learning.html#supervised-learning .

Namely the ones based on Decision Trees (DTs) and Random Forests (ensemble of randomized DTs).

Evaluation of Classification/Anomaly Detection Results

13. Calculate the main evaluation metrics for the previous results, namely, accuracy, precision, recall and the F1-Score. Infer also the confusion matrices.

>> Compare the obtained results.

Overall Testing and Evaluation

14. **Go back to point 2, and change the observation windows type and duration (e.g., 120 seconds windows, 300 seconds with 60 seconds slide, etc…). Redo all subsequent tests.**

>> Compare the overall obtained results.

>> Discuss the impact of the different observation types/duration.