Técnicas de Perceção de Redes
Network Awareness

Entity Profiling

(Classification and Anomaly Detection)

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/

File 'baseProfileClass.py' contains the base code to use in this gide.

Feature Extraction (time-independent)

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).

1. Read data for all three files, and plot and analyze the download and upload traffic profiles over time.

2. Divide each dataset in (sequential) observation windows of 5 minutes (300 seconds), and divide (randomly) a set of observations for training and another for testing. Plot and analyze, separately, the train and test dataset for each application.

3. From the train observations, create a single data file with Time-independent Descriptive Statistics features and a vector with the known classification of each observation.

Note1: The best format is 2-D matrix, where each line contains the set of features for each observation. The classification vector should contain the known data classification (e.g., 0-YouTube, 1-Browsing, 2-Mining) of each observation, in the same order used in the features.

Note2: The descriptive statistical features (for upload and download) may be: mean, standard deviation, and percentiles for 75%, 90% and 95%.

4. Make 2-D log-log plots using different pairs of features, where each point will have a color dependent on the known classification. Try different feature combinations, and try to find find possible discriminators between applications.

Note: Example features pair: (mean upload, mean download).

Feature Extraction (time-dependent)

5. Using the data observations, extract (for each one) the download and upload silence duration and calculate its mean and variance as four additional features.

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

Note: you may consider a silence period when less than 256 bytes (~4 small packets) are transmitted.

6. Calculate and plot the scalogram for one YouTube, Browsing, and Mining observation (download traffic). Observe the differences and explain how these relate with inherent service characteristics.

7. Using the data observations, extract (for each one) the download and upload data Wavelet scalogram and use the "Energy" at predefined scales (e.g., [2,4,8,16,32,64,128,256]) as additional features.

Make 2-D plots using different pairs of the new scalogram based features. Try different feature combinations, and try to find find possible discriminators between applications.

8. Construct three different set of features:

- A training set for anomaly detection from the inferred features using only YouTube and Browsing test datasets (Minning traffic will be the anomaly);

- A training set for traffic classification from the inferred features using all three classes of test datasets;

- A test set for anomaly detection and traffic classification from the inferred features using all three classes of train datasets;

For each set infer time-independent and time-dependent features.

Feature Normalization

9. Normalize the features sets by scaling each feature by its maximum absolute value (Max Abs Scaler). Analyze the mean and standard deviation of the features after scaling.

The resulting scaling done to train data MUST BE be the same applied to test data. Therefore, four different set of normalized features must be inferred:

- A training set for anomaly detection from the inferred features using only YouTube and Browsing test datasets (Minning traffic will be the anomaly), normalize using the scaling inferred from its own data;

- A training set for traffic classification from the inferred features using all three classes of test datasets, , normalize using the scaling inferred from its own data;

- A test set for anomaly detection from the inferred features using all three classes of train datasets, normalize using the scaling inferred from its the test features set with just two classes;

- A test set for traffic classification from the inferred features using all three classes of train datasets, normalize using the scaling inferred from its the test features with all classes;

Feature Reduction

10. For the four set of normalized features, use Principal Components Analysis (PCA) to reduce each set of features to **3 main components**. Make 2-D plots using the 2 of the new features outputted by PCA, where each point will have a color dependent on the known classification. Try to find find possible discriminators between applications.

Note: The resulting feature reduction done to train data MUST BE be the same applied to test data.

Feature reduction and normalization are two distinct data pre-processing methods and can be used independently/isolated or combined.

Anomaly Detection (Statistical Analysis)

11. Consider Browsing and YouTube as licit traffic and Crypto-Minning traffic as an anomaly. Using all inferred normalized features (not PCA components) from the train dataset (with only YouTube and Browising data), calculate the centroid of each class of licit traffic. Using the test dataset calculate for each observation all the normalized features (the same used above) and calculate the Euclidean distance to each centroid. Identify abnormal observations of the test dataset based on the distances above a specific threshold to the licit classes centroids.

12. Consider now the features outputted by applying the PCA to the normalized features (3 components) of the train dataset (with only YouTube and Browising data) and repeat the previous point.

13. Considering again only the features outputted by applying the PCA to the normalized features (3 components) of the train dataset (with only YouTube and Browising data), infer the multivariate normal (Gaussian) distribution parameters for each class of licit traffic. Identify abnormal observations on the test dataset based on probability (returned by the respective distributions of each licit class) below a specif threshold.

Anomaly Detection (Machine Learning)

14. Considering again only the features outputted by applying the PCA to the normalized features (3 components) of the 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 (using all normalized features).

15. Considering now only the normalized features (not PCA components) from the 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 (using all normalized features).

>> What can you conclude about the usage of PCA component based features with SVM?

Classification (Statistical Analysis)

16. Using only the features outputted by applying the PCA to the normalized features (3 components) of the train dataset, calculate the centroid of all three classes. Using the test dataset calculate for each observation the 3 main components of the PCA to the normalized features and calculate the Euclidean distance to each centroid. Classify each observation of the test dataset based on the lowest distance to the respective class centroid.

17. Using only the features outputted by applying the PCA to the normalized features (3 components) of the train dataset, infer the multivariate normal (Gaussian) distribution parameters for each class. Classify each observation of the test dataset based on the highest probability returned by the respective distributions of each class.

Classification (Machine Learning)

18. Using all inferred normalized features (not PCA components) from the 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 (using all normalized features).

19. Consider now the features outputted by applying the PCA to the normalized features (3 components) of the train dataset (with all three classes of data) and repeat the previous point.

>> What can you conclude about the usage of PCA component based features with SVM?

20. 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.

>> What can you conclude about the usage of PCA component based features with Neural Networks?