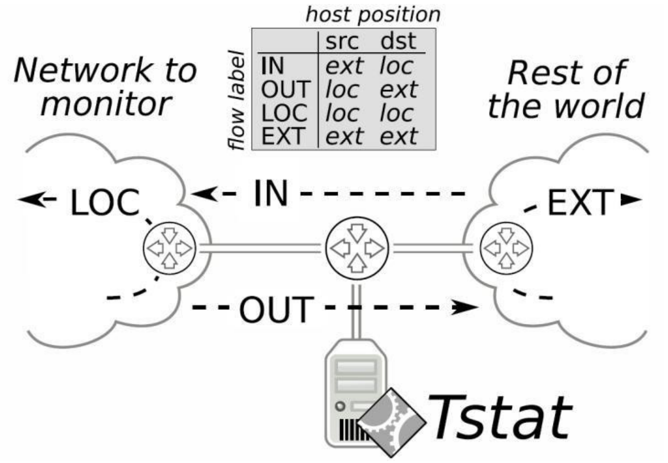
ML4N - Group Project 3

Patterns in encrypted web traffic HTTPS

Clarifications for this project can be asked to Luca Vassio: luca.vassio@polito.it

This project will need different aspects of Machine Learning applied to Internet measurements. In particular, you will use classification, clustering and regression techniques. As input data, we provide a network log trace file with TCP connections generated by thousands of users while browsing the web. For each network TCP connection, you have to identify and classify the service (e.g., Google, Amazon) that generated the traffic.
Thn, you will cluster the domain names observed in the trace given the traffic they produce. Finally, you should perform a regression prediction for the number of bytes transmitted and the RTT. For all parts, you will need to perform a complete machine learning pipeline.

In response to the ease of passively monitoring network traffic, the Internet has moved massively to encrypted protocols. The ease of new methods for obtaining and renewing server certificates, improvements in Web protocols that now support or mandate the use of encrypted channels, and tighter security policies in Web browsers have led to popular services moving to secure implementations. These implementations are robust against in-network surveillance and eavesdropping, increasing overall privacy and confidentiality. The most notable omission from the initial efforts to improve confidentiality was the domain name, which is still exchanged in plain-text in various situations. The presence of plain-text domain names in traffic gives eavesdroppers access to rich information, such as the content the user is interested in. This aspect gives the eavesdropper the ability to obtain a dataset of traffic flow statistics that contains the associated domain names as labels. Eavesdroppers can therefore create lists of websites visited by users, which clearly has an impact on user privacy.



You will use a dataset collected in July 2019 with statistics from the traffic using a tool called Tstat (http://tstat.polito.it/). Tstat is a passive traffic monitor that exports flow records, i.e., a single entry for each TCP/UDP stream in the network. Each flow record is composed of a rich set of statistics.

The basic objects that passive monitoring tools capture are the packets that are transmitted on the monitored link. We can group packets in flows if they share key IP header information. A common choice is to consider:

Flow = (Protocol, IP Source Address, Source Port, IP Destination Address, Destination Port)

Tstat automatically aggregates by flow the traffic (packets) captured.

Beside classical flow-level fields, such as IP addresses, port numbers, packet and byte-wise counters, Tstat extracts the domain names, which we use to label the TCP flows. The final dataset is composed of 125 features, 122 numerical described with an initial “_” in the featurename, plus the client IP address, the time when this flow was generated and the label. We removed the information about the server IP.

You can find more information regarding the features in the file readme.txt.

The data provided is already divided in 2 parts for training (https_training.csv) and test (https_test.csv).

Section 1 – Data exploration and pre-processing

Explore the dataset and learn about feature behavior at different levels, e.g., flow, domain name. Use various data visualization techniques and statistical analysis

Produce different visualizations and statistical analysis grouping the data at the flow level (rows of the dataset), at the IP level (clients) and domain name level (labels).

1. Distributions of features (EPDF or ECDF) per flow, domain name or IP
2. How are you merging the features at the domain name and IP levels?
3. Find most correlated features (at the 3 levels)
4. Plot statistics on number of bytes transmitted and the round trip time (RTT).

5. Apply PCA and t-SNE to the features at the 3 levels

Section 2 – Supervised learning – classification

In this section you will develop a supervised machine learning method that classifies the flows to predict the domain names visited by clients with high accuracy, using attributes derived from network traffic characteristics.

1. Choose at least 3 ML methods, and perform the model training, with default parameter configuration, evaluating the performance on both training and test set. Output the confusion matrix and classification report. Do you observe overfitting or under-fitting?
2. Tune the hyper-parameters of the models through cross-validation. How do performance vary? Which model generates the best performance?
3. Investigate the False Positive and False Negative. Can you draw considerations about the misclassification in terms of features? Report your analysis and findings for the ones you consider the most notable samples.

Section 3 – Unsupervised learning – clustering

In this task you will **cluster domain names** that produce similar patterns . The clustering will be done in an unsupervised fashion, independently of the labels used in Section 2. (clustering, unsupervised task).
The goal of this section is to develop an unsupervised machine learning method for grouping domain names. This will involve checking to see if there are “families” of domain name (e.g.,video streaming, news, etc.).

Choose at least 2 Clustering Algorithms, and for each of them:

1. Find a representation for each domain name (features)
2. Determine the number of clusters: This can be done using methods like the elbow method or silhouette analysis. Explain your reasoning.
3. Find the best hyper-parameters, if any
4. Evaluate the clusters through clustering metrics and performance indicators
5. Report a coarse analysis of the detected clusters (e.g. ECDF of domain names per cluster).
6. When analyzing domain names in the clusters, can you identify ”families” of domain names? I.e., domain names that offer similar services.

Section 4 – Regression – Estimate bytes transmitted and round trip time

In this section you need to perform regression of the following two variables:

- s_bytes_all. It represents the number of bytes from server to client transmitted in the payload, including retransmissions. It is measured in bytes.
  Note: in order to predict this feature you cannot use the following column: s_bytes_uniq
- s_rtt_avg. It represents the average RTT from server to client computed measuring the time elapsed between the data segment and the corresponding ACK. It is measured in milliseconds (ms). Remove values equal to 0.
  Note: in order to predict this feature you cannot use the columns related to other measures of RTT

For each of the two regression tasks:

1. Choose at least 3 ML methods, and perform the model training, with default parameter configuration, evaluating the performance on both training and test set. Output the regression metrics results. Do you observe overfitting or under-fitting?
2. Tune the hyper-parameters of the models through cross-validation. How do performance vary? Which model generates the best performance?
3. Which of the two problems gives a higher Mean Absolute Error? What is the error measured in (unit)? Which of the two problems is easier to solve?

Project aknowledgement: Danilo Giordano - danilo.giordano@polito.it