Cyber Data Analytics Assignment 3

Study:

1. https://stratosphereips.org
2. In particular the CTU-13 data: https://mega.nz/#F!vdRmBA6D!yMZXx74nnu8GjhdwSF54Sw
3. And honeypot data available from: https://surfdrive.surf.nl/files/index.php/s/mOvvW5hZZ5PF86N
4. Garcia, Sebastian, et al. "An empirical comparison of botnet detection methods." *computers & security* 45 (2014): 100-123.
5. Pellegrino, Gaetano, et al. "Learning Behavioral Fingerprints From Netflows Using Timed Automata."
6. Cormode, Graham, and Marios Hadjieleftheriou. "Finding frequent items in data streams." *Proceedings of the VLDB Endowment* 1.2 (2008): 1530-1541.

Sampling task (5pt) – ½ A4

Download the Honeypot data (34 GB, but we only use the first IP address values). Load one of the files in your favorite editor and describe in a few lines what you see.
Count the number of distinct source IP_addresses, what are the 10 most frequent values? Write code for the FREQUENT algorithm, use it to count the amounts in one pass (no need to actually stream the data, you may store it in memory, or run every file separately but do store and load the intermediate results). Run FREQUENT using reservoirs of size 10, 100, and 1000. What are the 10 most frequent IP-addresses and their frequencies? Explain the differences with the true frequencies using the theoretical bounds.

Hashing task (5pt) – ½ A4

Split the honeypot data 50:50 into split (first half) and test (second half) maintaining the temporal order. Test for each source IP in the test set whether it occurs in the train set. Write code for a BLOOM filter (see, e.g., http://www.maxburstein.com/blog/creating-a-simple-bloom-filter/). Perform the same test using BLOOM filters of sizes 10, 100, and 1000. Explain the differences using the theoretical false positive rates.
Extend the BLOOM filter to a Count-Min sketch, play with different heights and widths for the CM sketch matrix. Does it provide better approximations than the FREQUENT algorithm? Use the theory to explain why (not).

Botnet classification task (5 pt) – 1 A4

Download the CTU-13 data, containing NetFlows from 13 scenarios of both benign and infected hosts. Study paper 4 and construct a classifier for detecting anomalous behavior in NetFlows. Do not forget to study and deal with properties of your data such as class imbalance. Evaluate your method in two ways: on the packet level (as in paper 4), and on the host level (as in paper 5). Very briefly describe the steps you employed to build/improve your classifier. Focus on your implementation being correct rather than excellent performing (classification is simply hard).

Botnet fingerprinting task (5 pt) – 1 A4

Learn a sequential model from data coming from an infected host, for the CTU-13 scenarios considered in paper 5. First cluster the NetFlows using a clustering method of your choice in order to discretize the data. Use the "ELBOW" method (https://en.wikipedia.org/wiki/Elbow_method_(clustering)) to determine the number of clusters. You are free to choose a sequential model (Markov chain, n-grams, state machine, HMM, Stratosphere…). For state machines you may use our code for state machine learning from https://bitbucket.org/chrshmmmr/dfasat. Stratosphere is available from https://stratosphereips.org/. Code for HMMs is available in many packages, or you can use slow code from BlackBoard. Simply use a sliding window to obtain sequence data. Evaluate how many new infections your method finds and false positives it raises (as in paper 5).