Cyber Data Analytics Assignment 3

INTRODUCTION

Most datasets from software or hardware systems do not fit into memory. Data stream mining is a machine learning framework that aims to store only the data parts that are relevant for learning, e.g., by sampling and hashing.

Profiling and fingerprinting are two key techniques (in addition to anomaly detection) for threat discovery. Profiling builds an overall picture, typically a probability distribution, from training data and matches this against new data. Fingerprinting instead looks for very specific patterns that only occur when a threat is present.

In this exercise, you will apply the techniques taught in class to build approximations of a large network data streams on-the-fly and evaluate the quality of the obtained approximations. In addition, you will apply fingerprinting/profiling to the problem of botnet detection in computer networks and compare it with flow classification. In the bonus, you can try to fool these detectors by crafting adversarial examples.

LEARNING OUTCOMES

After completing this assignment, you will be able to:

- *Use sampling and hashing to create correct approximations of large data streams.*
- *Build methods for fingerprinting and profiling behaviours.*
- *Successfully detect botnets in network data.*

INSTRUCTIONS

Sampling task – < 1 A4

Pick any of the CTU-13 datasets (Malware capture 42 to 54). Download the *unidirectional* netflows. *DO NOT DOWNLOAD THE VIRUS THAT WAS USED TO GENERATE THE DATA UNLESS USING A VM OR OTHER SANDBOX.* The flows are collected from a host in the network. Its IP address should be obvious from the data sample. We are interested in the other addresses the host connects with.
Estimate the distribution over the other IP_addresses, what are the 10 most frequent values? Write code for RESERVOIR sampling, use it to estimate the distribution 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). Use a range of reservoir sizes. What are the 10 most frequent IP-addresses and their frequencies when sampled? Use the theory to explain any approximation errors you observe.

Sketching task – < 1 A4

Build code for computing a COUNT-MIN sketch, play with different heights and widths for the Count-Min sketch matrix. Compare it to the RESERVOIR sampling strategy. Is it more space-efficient/accurate? What about run-time? Use the theory to explain any differences you observe.

Flow data discretization task – 1 A4

We aim to learn a sequential model from NetFlow data from an infected host (*unidirectional netfows*). Consider scenario 10 from the CTU-13 data sets (see paper 4 from below resources). Remove all background flows from the data. You are to discretize the NetFlows. Investigate the data from one of the infected hosts. Select and visualize two features that you believe are most relevant for modeling the behavior of the infected host. Discretize these features using use any of the methods discussed in class (combine the two values into a single discrete value). Do you observe any behavior in the two features that could be useful for detecting the infection? Explain. Apply the discretization to data from all hosts in the selected scenario.

Botnet profiling task – 1 A4

Choose a probabilistic sequential model (Markov chain, n-grams, state machines, HMMs, …). Code for HMMs is available in many packages, or you can use our own Python code from Brightspace. For state machines you may use our code for state machine learning from https://bitbucket.org/chrshmmmr/dfasat. Use a sliding window to obtain sequence data. Learn a probabilistic sequential model from the data of one infected host and match its profile (as discussed in class) with all other hosts from the same scenario. Evaluate how many new infections your method finds and false positives it raises (as in paper 4). Can you determine what behaviour your profile detects?

Flow classification task – 1 A4

Study paper 3 and construct a classifier for detecting anomalous behavior in individual NetFlows (every flow is a row, ignoring sequences). 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 3), and on the host level (as in paper 4). Do you prefer using a sequential model or a classifier for detecting botnets? Explain why.

Bonus: Adversarial examples – 1/2 A4

Study the blog-post on adversarial machine learning. Construct adversarial examples for the flow classifier and the botnet profiler from the previous tasks. Show that it fools the detectors and argue why the examples are (close to) malicious.

RESOURCES

Slides from Lectures 5, 6, 7

Study:

1. https://stratosphereips.org
2. In particular the CTU-13 data: https://www.stratosphereips.org/datasets-ctu13/
3. Garcia, Sebastian, et al. "An empirical comparison of botnet detection methods." *computers & security* 45 (2014): 100-123.
4. Pellegrino, Gaetano, et al. "Learning Behavioral Fingerprints From Netflows Using Timed Automata."
5. Cormode, Graham, Count-min sketch (notes)
6. https://medium.com/alfagroup-csail-mit/robust-detection-of-evasive-malware-part-1-312efc1bccc3

Links on Brightspace to online tutorials.

Code samples available on Brightspace.

PRODUCTS

A small report (max 4 pages, 5 including bonus), and the code used to obtain the results. Both will be assessed using the below criteria.

ASSESSMENT CRITERIA

The assignment will be assessed by peer review. The form will be made available directly after the assignment deadline.

Knockout criteria (will not be evaluated if unsatisfied):
Your code needs to execute successfully on computers/laptops of your fellow students (who will assess your work). You may assume the availability of 4GB RAM and a Linux operating system, possibly a virtual machine. Please test your code before submitting. In addition, the flow from data to prediction has to be highlighted, e.g., using inline comments.
Your report needs to satisfy the page limit requirements for the different parts. When working in a data analysis notebook, you have to copy and paste the text and results into a printable document satisfying the requirements.
Submissions submitted after the deadline will not be graded.

The report/code will be assessed using these criteria:

Criteria	*Description*	*Evaluation*
Sampling	*Implemented correctly, explanation accurate*	*0-5 points*
Sketching	*Implemented correctly, analysis, and evaluation are sound*	*0-5 points*
Discretization	*The discretization is sound. The relevance and behavior are analyzed and explained clearly.*	*0-5 points*
Profiling	*Probabilistic model is correctly implemented and evaluated. Interesting behavior is discovered in the flows coming from the infected machine.*	*0-5 points*
Classification	*Correctly implemented machine learning method and sound comparison and conclusion.*	*0-5 points*
Bonus	*Created adversarial examples, misclassification and maliciousness are explained and sound.*	*0-5 points*
Report and code	*The data-detection flow is clearly described, including preprocessing and post-processing steps.*	*0-5 points*

Your total score will be determined by summing up the points assigned to the individual criteria, and averaging to account for the number of peer reviews. In total 35 points can be obtained in each course assignment, the total number of obtained points will be divided by 90 to determine the final grade. In case one of the reviews is significantly worse than (at least 10 points difference) the others, this review score is not taken into account.

You will receive a penalty of 5 points for each peer review not performed. Significantly different reviews will be subject to investigation. If deemed badly done by the teacher or TA, you will also receive 5 penalty points.

SUPERVISION AND HELP

We use Mattermost for this assignment. Under channel Lab1, you may ask questions to the teacher, TAs, and fellow students. It is wise to ask for help when encountering start-up problems related to loading the data or getting a machine learning platform to execute. Experience teaches that students typically answer within an hour, TAs within a day, and the teacher the next working day. When asking a question to a TA or teacher, your questions may be forwarded to the channel to get answers from fellow students. Important questions and issues may lead to discussions in class.

There is no separate lab session hosted at the university, it is your own responsibility to start and finish on time.

SUBMISSION AND FEEDBACK

Submit your work in Brightspace, under assignments, and in the peer reviewing system. Within a day after the deadline, you will receive several (typically two) reports to grade for peer review as well as access to the online peer review form. You have 5 days to complete these reviews. You will then receive the anonymous review forms for your groups report and code.

There is the possibility to question the amount of points given to your work, up to one week after receiving the completed forms. You should do so via a private message to the teacher and TA in Mattermost.