COMP3217 Security of Cyber-Physical Systems
23/24 Coursework 2

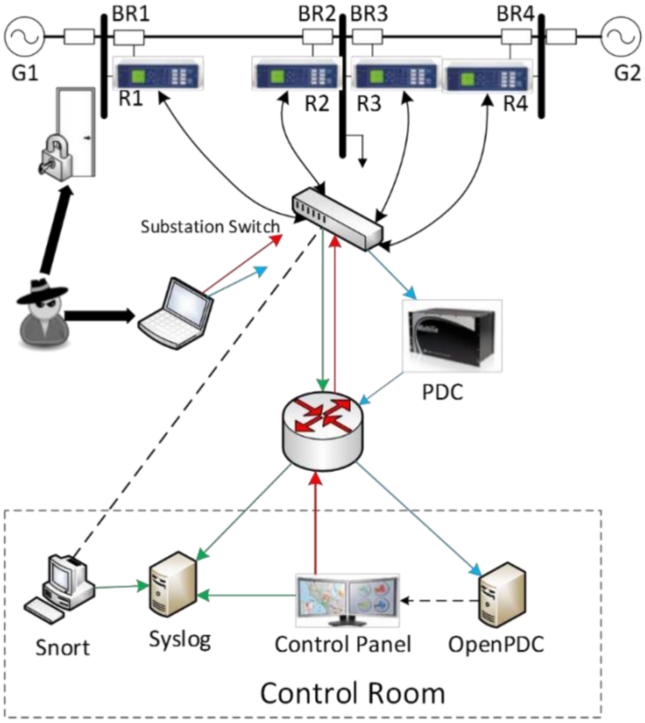
Detection of Attacks on Power System Grids

The coursework is worth 50% of the total mark for this course. It involves writing programs for the tasks described below, and you can choose any major programming language such as C/C++, Python, etc. The purpose of this coursework is to understand the power system attacks on electric grids CPS, understand the type of attacks, develop detection techniques for such attacks, and get familiar with some cyber-physical system security programming skills.

Release date – 19th March
Due date – 6th May 4 PM

Scenario

Consider a power system framework configuration as shown in the figure below:



In the network diagram we have several components, firstly, G1 and G2 are power generators. R1, R2, R3, and R4 are Intelligent Electronic Devices (IEDs) that can switch the breakers on or off. These breakers are labelled BR1 through BR4. We also have two lines. Line One spans from breaker one (BR1) to breaker two (BR2) and Line Two spans from breaker three (BR3) to breaker four (BR4). Each IED automatically controls one breaker. R1 controls BR1, R2 controls BR2 and so on accordingly. The IEDs use a distance protection scheme which trips the breaker on detected faults. Operators can also manually issue commands to the IEDs R1 through

R4 to manually trip the breakers BR1 through BR4. The manual override is used when performing maintenance on the lines or other system components.

Three different types of scenarios can occur in such a system:

1. Natural faults and service events: Natural faults include short circuit faults, for example a short circuit in a power line. Service event pertains to events related to servicing the grid. For example, line maintenance where one or more relays are disabled on a specific line to do maintenance for that line.
2. Normal event: Regular operation with no issues.
3. Remote cyber-attacks: This includes (a) data injection - imitate a valid fault by changing values to parameters such as current, voltage, sequence components etc. This attack aims to blind the operator and causes a black out, and (b) Remote tripping command injection - this is an attack that sends a command to a relay which causes a breaker to open. It can only be done once an attacker has penetrated outside defences.

This course work has two parts:

PART A

Input data -

- You are given a set of 6,000 system traces (TrainingDataBinary.csv), where half of them is labelled as 0 – normal events and the other half are labelled as 1 – abnormal data injection attack events. There are 128 numbers in each trace. These 128 numbers represent features. There are 29 types of measurements from each phasor measurement units (PMU). A phasor measurement unit (PMU) or synchrophasor is a device which measures the electrical waves on an electricity grid, using a common time source for synchronization. In our system assume there are 4 PMUs which measure 29 features totalling to **116** PMU measurement columns total. The index of each column is in the form of "R#-Signal Reference" that indicates a type of measurement from a PMU specified by "R#". The signal references and corresponding descriptions are listed below. For example, R1-PA1: VH means Phase A voltage phase angle measured by PMU R1 etc. After the PMU measurement columns, there are **12** columns for control panel logs, Snort alerts and relay logs of the 4 PMU/relay (relay and PMU are integrated together). The last column – 129th -is the label for the event – normal/abnormal. 0 indicates normal event while 1 indicates data injection attack. These 6,000 rows are basically the training data.
- You are also given 100 system traces without labels, which are the testing data (TestingDataBinary.csv), where there are 128 numbers in each row.

You are expected to -

- You need to design and implement a machine learning technique to model the training data and compute the labels for all testing data (i.e., 0 or 1 for each system trace).
- You need to submit your output file (TestingResultsBinary.csv) in the same format as the training data. You need to put the computed labels for all of the testing data – in the same order as was given to you.
- You are required to submit your source code.
- You also need to submit a report, where you should give a clear description of the problem, and the machine learning techniques used to compute the labels. You should also analyse and discuss your results in terms of training error and training accuracy. The accuracy on testing data followed by accuracy on the training data will be an important factor in determining your performance on this coursework.

PART B

Input data -

- You are given a set of 6,000 system traces (TrainingDataMulti.csv), where data for three types of events is evenly distributed: normal events, abnormal data injection attack events, and abnormal command injection attack events. There are 128 features in each trace and the last column – 129th - is the label for the event. 0 indicates normal event, 1 indicates data injection attack, and 2 stands for command injection attack. These 6,000 rows are basically the training data.
- You are also given 100 system traces without labels, which are the testing data (TestingDataMulti.csv), where there are 128 numbers in each row.

You are expected to -

- You need to design and implement a ML technique to model those training data and compute the labels for all testing data (i.e., 0 or 1 or 2 for each system trace).
- You need to output a file (TestingResultsMulti.csv) with the same format as the training data. You need to put the computed labels for all the testing data – in the same order as was given to you.
- You are required to submit your source code.
- You also need to submit a report, where you should give a clear description of the problem, and the machine learning techniques used to compute the labels. You should also analyse and discuss your results in terms of training error and training accuracy. The accuracy on testing data followed by accuracy on the training data will be an important factor in determining your performance on this coursework.

For both parts:

- You must use Google Colab for your coding.
- Your code should contain sufficient comments. It must be compilable. **Non compilable code will get 0 marks for the whole assignment.**
- Your code should be laid out neatly with consistent indentation.

Assessment criteria
Correctness of classification results for part A: 40%
Correctness of classification results for part B: 40%
Code – 10%
Report – 10%

Total marks: 100

Submission

Due Date: 6th May 2024, 4 PM
You are required to submit the

- A report – no more than 5 pages describing solution for both the parts. Any report more than 5 pages will be penalized. 2 marks will be deducted for each additional page. This must go to Handin.
  Note - It is acceptable for you to experiment with many ML methods and just mention that briefly in your report. However, for the purpose of the report only the method yielding the best results is required to be presented at minimum.
- Two output files into Handin. The names of the files must be TestingResultsBinary.csv and TestingResultsMulti.csv. The order of the data should be the same as provided to you.
- A link to compilable and well commented code in Google Colab. The link should be mentioned in your report. Your code must output the same two result files as you have submitted to Handin. Please make sure that the link accessible to the instructor. If you have tried multiple ML methods, then you must have them in your code in addition to your best performing method.