# Cybersecurity and AI Case Studies

# (MOD 006570, Element 010)

As part of the assessment for Element 010, you are expected to submit the following three files through a single Canvas link:

- Jupyter notebook of your coursework project
- Coursework project report
- Lab Logbook (logbook must contain GITHUB or sharable one drive link of all your source code)

# Part A (30 marks):

In this part of the assessment, you will implement and evaluate an AI model for intrusion detection using the Edge-IIoTset dataset. The link to the dataset is provided here:

Edge-IoT Dataset

1. You need to submit a functional code for cyber threat classification based on any AI model of your choice. Your code must present thorough experiments on the choice of your selected model(s) (5 marks).
2. Based on the code, you must provide details of your results and critically evaluate them in your report. The focus should be on intrusion detection and identifying malicious activities within the IoT network. (25 marks)

Note that your code and report must experimentally prove the rationale behind the selection and choice of your AI model.

Note that **you DO NOT have to write an Introduction, literature survey, or Conclusions.** You only need to provide your model choice, prove the rationale behind the selected model, and provide results.

# Part B (30 marks):

Imagine a scenario where a company intends to train and deploy the model that you created using the same dataset. However, an **insider threat** exists in the form of a dishonest employee who gains unauthorized access to your training code. This insider attack involves the

employee poisoning the training data by manipulating the labels in the training samples. For instance, in your dataset X_train and y_train, the employee maliciously flips the labels stored in y_train to undermine the integrity of the training process. The consequence of such manipulation is that the input data and their corresponding labels no longer align. For example, if a network traffic sample originally corresponds to a malicious attack, the insider alters the label so it is now classified as benign traffic, thereby compromising the intrusion detection system. One method to manipulate the labels involves taking a label and randomly replacing it with another label, which is also called random label flipping. Another method can be systematically replacing all attack labels with a benign label, effectively hiding cyber threats within the dataset and weakening the model's ability to detect intrusions.

1. You are required to provide functional code for manipulating the labels in your training set and observing its impact on the performance of the model developed in part A. You must compare the performance of the model before and after such data poisoning across various percentages of flipped labels in the training samples for your intrusion detection neural network model. For example, rather than manipulating all labels, manipulate a certain percentage of labels (5%, 10%, 15%, and so on) and graphically analyze the degradation of model performance in terms of its ability to detect attacks. While you could use a simple strategy like random label flipping for your analysis, higher marks will be awarded if you can propose a more sophisticated poisoning strategy exposing the potential vulnerabilities of AI model. The objective of your label manipulation strategy should be to **minimize the number of manipulated labels while maximizing the model's failure rate** in detecting cyber threats(15 marks).

2. You need to provide a **detailed report analyzing your results**, focusing on the your data poisioning strategy and its impact on threat detection AI model (15 marks).

# Part C (20 marks):

Provide a theoretical analysis based on existing literature and your own insights on how label manipulation in the training dataset can be prevented or detected using a cybersecurity-based approach (e.g., cryptography or any other method). Create and describe a block diagram of your model training and deployment pipeline, explaining where your proposed solution fits within the process. Summarize your ideas and findings theoretically, ensuring clarity and coherence. Higher marks will be awarded if you can propose a novel approach for mitigating the impact of such label manipulation (20 marks).

**Note: You are not required to provide code for this part (Part C). However, if you choose to include code, you will receive an extra 5 extra points based on the novelty of your solution.**

# Part D – Lab Logbook Submission (20 marks)

For each tutorial session, we will request specific details or summaries of the work you have conducted. For example, you may be required to include graphs or charts in your lab logbook for certain experiments.

You are expected to include the specified summaries for 10 tutorial sessions, as outlined on Canvas. To ensure you have the necessary information for each lab session, please refer to the Canvas page for each week, where you will find the specific instructions and details to be added in this section. Add the specific details or summaries in the relevant section of this document.

It's important to note that you should incorporate the specific details or summaries for each tutorial session into this singular logbook. There's no need to create separate logbooks for each lab; instead, you will maintain one logbook where you consistently record the details for each week's activities.

**Submission Requirement for this part:** In this part, you are required to submit a single lab logbook that you have maintained throughout the semester, including the specified summaries for 10 tutorial sessions. Within your lab logbook, provide the link to your GITHUB or a shareable OneDrive folder containing all your source code. The evaluation will primarily be based on your lab logbook, but we will also review your code to confirm that the logbook accurately represents the output from your code.