1.1. PART 1 – IoT Case Study – Designing & Developing Data Science Solutions for IoT [Contributes 60% of your assessment]

1.1.1. Tasks Specification

To complete this part of the coursework, you will work through the tasks specified below. It is strongly recommended that you complete the tasks in the order suggested, as they build on one another, and later tasks may not make sense if the preceding tasks are not completed. You may like to return to a task after you have completed it to add more features or to increase the complexity of your solution/algorithm; more complete and challenging solutions may be awarded more marks.
You are going to develop a Pollution monitoring IoT solution based on the data available from the EU project CityPulse (http://iot.ee.surrey.ac.uk:8080/datasets/pollution/) and will carry out associated tasks. The website provides open data about smart-cities applications from different European cities. You are provided with a data dump of some of the data from this data source that you need to use to build a system. You are required to complete the following tasks that test your ability to develop a functional application/solution and will cover various aspects of the Advanced IoT module, including data science for IoT, IoT messaging protocols, data processing, and real-time streaming analytics.
You will submit your code and a report as part of your submission.

1.1.2. Task 1 – Data Preparation/Analysis

Kindly do data preparation and analysis highlighting the dataset and features

1.1.3. Task 2 – Time Series Analysis and Prediction

Utilise the data points (suggested: timestamp and Carbon Monoxide) for each of the datasets to:

- Divide them into test and train datasets.
- Create ARIMA based time series model with a train data frame.
- Apply this model to test the data frame for the prediction of Pollution.
- Discover the Mean Absolute Percentage Error (MAPE) for your model, and experiment with different (p, d, q) values to reduce the MAPE to the lowest possible value you can find.
- Draw plots to explain your final results.

In your report, write a few paragraphs (max 300 words) summarising your solution. Then, place your code for this task in a folder called 'TimeSeries'.

1.1.4. Task 3 – Use MQTT protocol for simulating real-time streams from the two stations

- Write an MQTT publisher programme named 'Arhus1'. Arhus1 publisher will publish two data points (timestamp, carbon monoxide) from your group's dataset file as its' payload every 10 seconds. Note: you can use the MQTT broker provided by Eclipse for this purpose or utilise Mosquito (https://mosquitto.org/) broker.
- Write an MQTT publisher programme named 'Aarhus2'. Choose the topic of your liking for publishing.
- Aarhus2 publisher will publish two data points (timestamp, carbon monoxide) from the pollutionData209960.csv file as its' payload at every 10 seconds.

- Write an MQTT subscriber programme called 'PollutionSubscriber'. This programme will subscribe to the topics of Arhus1 and Arhus2.
- Show the output of your publishers and subscribers working correctly.
- In your report, write a few paragraphs (max 300 words) summarising your solution and output. Then, place your code for this task in a folder called 'MQTT'.

1.1.5. Task 4 – Use Apache Flink for real-time stream processing and complex event processing (CEP)

Write a programme using Apache Flink framework to:

- Ingest the real-time streams from the two stations (as in Task 3, i.e. section 1.1.4; you can use CSV files as input or your transformed data from Part 1, section **Error! Reference source not found.**)
- Define a complex event processing pattern (Apache Flink Pattern class) for low Pollution (you can decide the threshold for what is a low Pollution event yourself in terms of setting the values)
- Apply this pattern on the real-time stream, and when the pattern is fired, generate an alert (Apache Flink Alert) by printing the alert on the screen.

In your report, write a few paragraphs (max 300 words) summarising your solution. Then, place your code for this task in a folder called 'CEP'.

1.2. Task 4 – Short report of about 500 words [5 Marks]

Here you write a short report summarising all the tasks you did in section 3.3. The report will be used to write a paper or technical report in part 3.