Internet of Things (IoT) Introduction

Module Leaders (CSC8112):
Prof. Rajiv Ranjan, Dr Tejal Shah, Dr. Tomasz Szydlo

CSC8112 Information

- Module web site (lecture notes, homework, policies): https://ncl.instructure.com/courses/50097
- Write to us: raj.ranjan@ncl.ac.uk (Module Leader)
  - tomasz.szydlo@ncl.ac.uk (Co-Module Leader)
  - tejal.shah@newcastle.ac.uk (Co-Module Leader)
- Module Lead Demonstrators:
  - Rui Sun, ruisun.ray@gmail.com
  - Zeynep Erdogan, z.y.erdogan2@newcastle.ac.uk
- Broadcast messages (mostly from me): *Canvas*

CSC8112 TOPICS

- Fundamentals
- Networking
- Application Programming Workflows
- Machine Learning
- Application Use cases
  - Healthcare 4.0
  - Smart Building
  - Environmental Risk Management (e.g. Flooding)

CSC8112 Canvas

https://ncl.instructure.com/courses/55159

Account
Dashboard
Courses
Calendar
Inbox
History
Commons
10
Help

CSC8112 (24/25)

View as

2024/25

- Home
- Announcements
- Syllabus
- Modules
- ReCap
- Assignments
- Library Reading List
- Grades
- Accessibility Report
- Zoom
- Item Banks
- NU Reflect
- Quizzes
- Discussions
- Pages
- Files
- People
- BigBlueButton
- Outcomes
- Rubrics
- Collaborations
- Smart Search
- NCL Voice: Let's Talk
- Settings

Recent announcements

Internet of Things - CSC8112

Assign to | Edit

Welcome to CSC8112 - Internet of Things

Syllabus

See the syllabus area for information about the module leader, contact hours, learning outcomes and assessments.

Online Lecture Reading List

Below articles discuss technical challenges related to devising a new Internet of Things (IoT) programming paradigm, which we developed at Newcastle University and is referred to as the Osmotic Computing, for managing lifecycle operations (e.g., configure, monitor, network, and reconfigure) of IoT applications (e.g., smart traffic management, smart city, flood disaster management, among many others) across heterogeneous computing infrastructure (Cloud, Edge, and IoT Devices).

- Osmotic (IoT-Edge-Cloud) Computing
- Osmotic Computing and Benchmarking

CSC8112 Canvas

Online Lecture Reading List

Below articles discuss technical challenges related to devising a new Internet of Things (IoT) programming paradigm, which we developed at Newcastle University and is referred to as the Osmotic Computing, for managing lifecycle operations (e.g., configure, monitor, network, and reconfigure) of IoT applications (e.g., smart traffic management, smart city, flood disaster management, among many others) across heterogeneous computing infrastructure (Cloud, Edge, and IoT Devices).

- Osmotic (IoT-Edge-Cloud) Computing
- Osmotic Computing and Benchmarking
- IoT and Blockchain
- Osmotic Computing Messaging
- Osmotic Computing and Artificial Intelligence
- Osmotic Computing and Cybersecurity
- Osmotic Computing and IoT Data Analysis Workflows
- Building IoT-based Sensor Infrastructure
- IoT and Data Integration: A Case Study from LandSlide Management

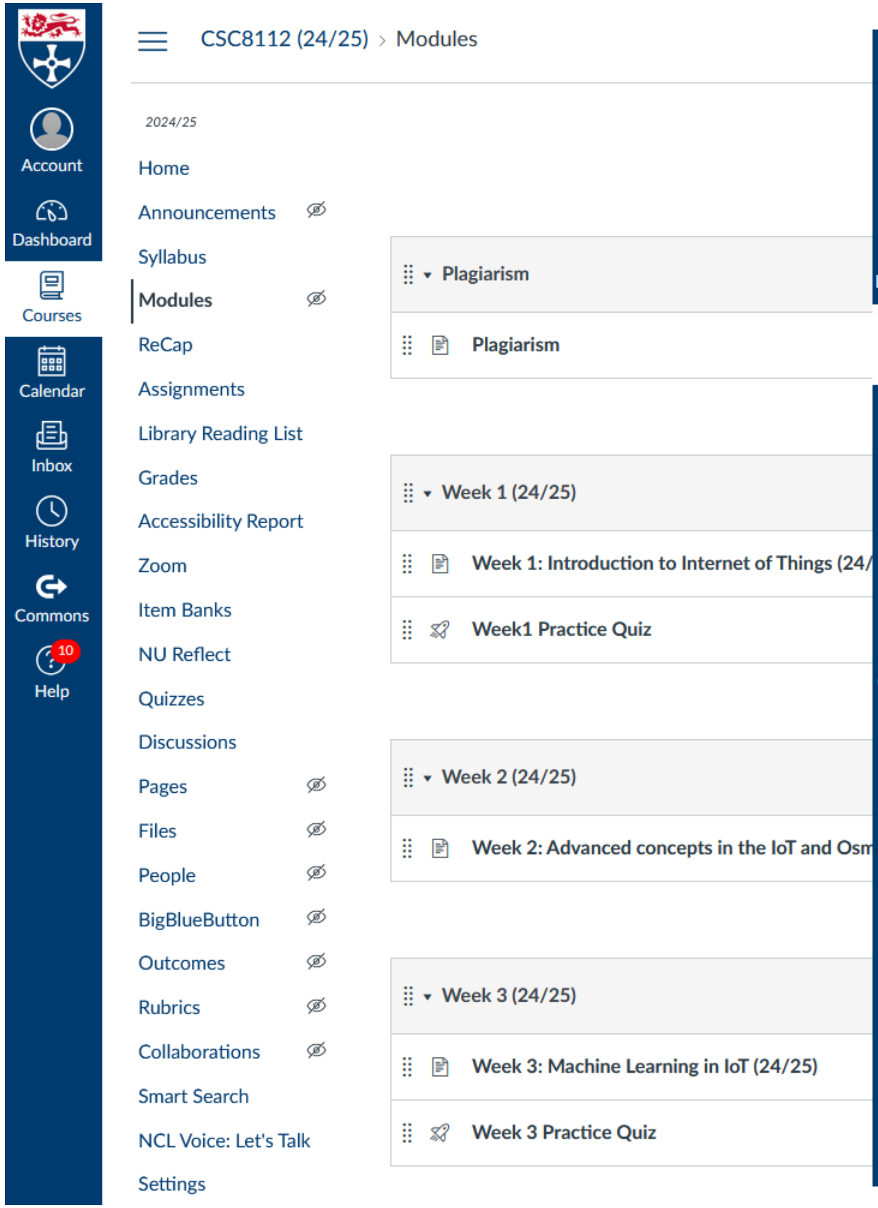
Coursework Assessment Reading List

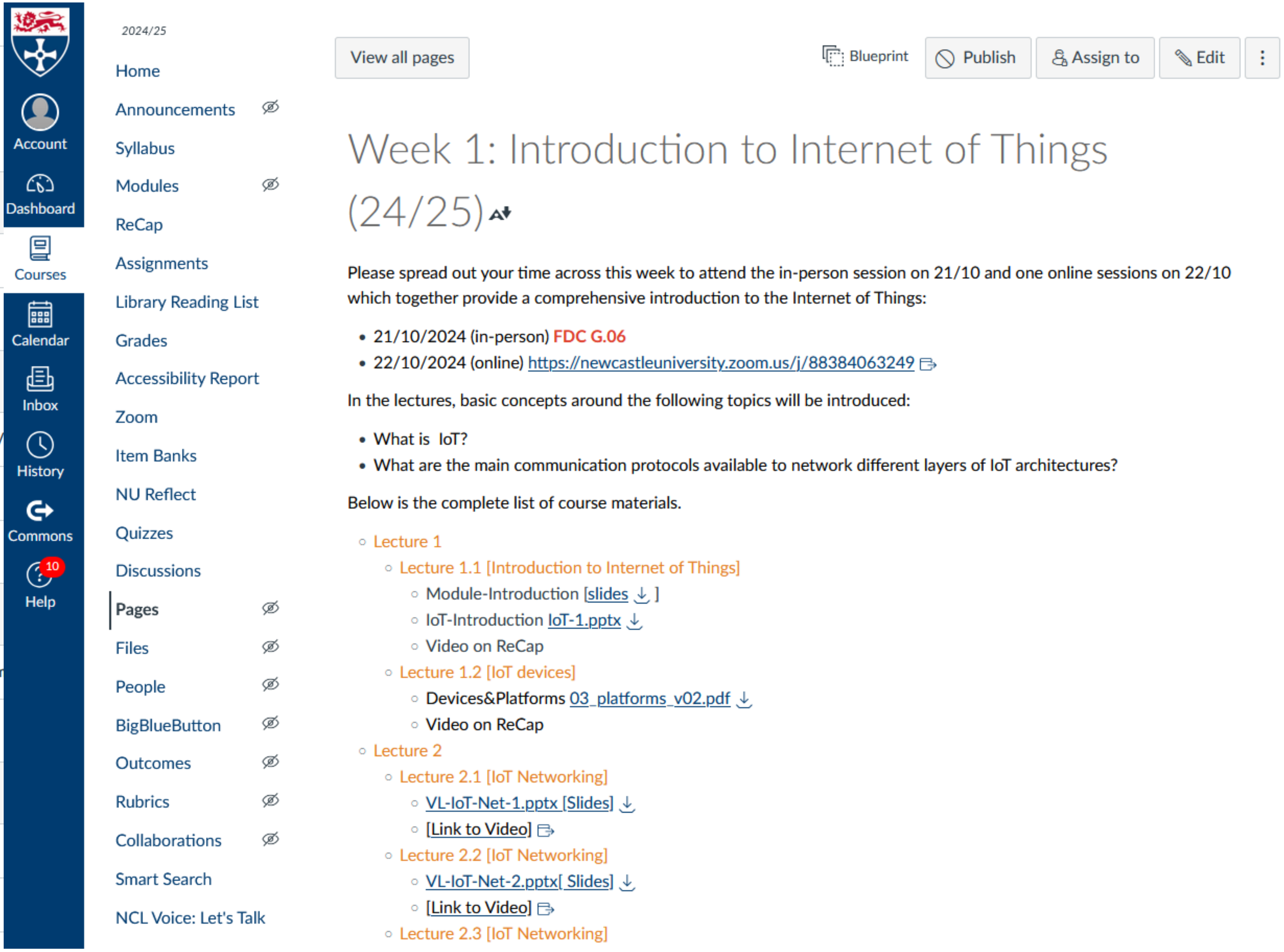
This coursework is aimed at helping students to understand the essential components and concepts in Edge computing, IoT network, and edge-cloud offloading concepts. In this coursework, you will get an opportunity to validate the core concepts that will be covered through online lectures. To do so, you will need to utilise a discrete event IoT Simulator called the IoTSim-OSMOSIS, which was developed by my research group. You are expected to read the paper in the reference list before commencing the coursework. By reading this paper, you will be able to learn about the **programming environment** around which this coursework is modelled on. The coursework is aimed to achieve the following learning outcomes:

- You will be able to understand and reason about core technical challenges involved with setting up complex IoT systems including networking and edge-cloud resource allocation.
- You will be to gain a technical understanding of an IoT simulator, which will help you validate IoT application and system engineering ideas in controlled environments.
- You will be able to test your knowledge gained via online lecture material.

[1] K. Alwasel, R. Ranjan, et al., IoTSim-Osmosis: A Framework for Modelling & Simulating IoT Applications Over an Edge-Cloud Continuum, Journal of Systems Architecture (OCT 2020).

CSC8112 Canvas (PPT + Videos)





CSC8112 Canvas (PPT + Videos)

Account
Dashboard
Courses
Calendar
Inbox
History
Commons
10 Help

2024/25

- Home
- Announcements
- Syllabus
- Modules
- ReCap
- Assignments
- Library Reading List
- Grades
- Accessibility Report
- Zoom
- Item Banks
- NU Reflect
- Quizzes
- Discussions
- Pages
- Files
- People
- BigBlueButton
- Outcomes
- Rubrics
- Collaborations
- Smart Search
- NCL Voice: Let's Talk
- Settings

Week 2: Advanced concepts in the IoT and Osmotic Computing (24/25)

Please spread out your time across this week to attend the in-person session on 28/10 and online sessions on 29/10 and 31/10, which together provide discussion on advanced concepts in IoT:

- 28/10/2023 (in-person) **FDC G.06**
- 29/10/2023 (online) https://newcastleuniversity.zoom.us/j/81026032334
- 31/10/2023 (online) https://newcastleuniversity.zoom.us/j/85670865858

This week contains data flow control and management for IoT as well as the digital twin concept:

- Lecture 3
  - Lecture 3.1 [Osmotic (IoT+Edge+Cloud) Computing]
    - Osmotic-Computing.pptx [Slides]
    - [Link to Video1]
    - [Link to Video 2]
  - Lecture 3.2 [Osmotic Computing (IoT) Deployment Optimisation]
    - [Slides]
    - [Link to Video]
- Lecture 4
  - Lecture 4.1 [Digital Twin]
    - [digital_twin]
    - [Link to Video]
  - Lecture 4.2 [Multi-Objective IoT Deployment Optimisation]
    - [Slides]
    - [Link to Video]
- Lecture 5
  - Lecture 5.1 [Basics of Deep Neural Networks]
    - [Slides]
    - [video]
  - Lecture 5.2 [Federated Learning]

Additional Materials

- Lecture A [Osmotic Computing (IoT) Monitoring]
  - OsmoticMonitoring-1.pptx
  - [Link to Video]
- Lecture B [IoT Configuration Selection]
  - IoT_Configuration_Selection.pptx [Slides]

Account
Dashboard
Courses
Calendar
Inbox
History
Commons
10 Help

2024/25

- Home
- Announcements
- Syllabus
- Modules
- ReCap
- Assignments
- Library Reading List
- Grades
- Accessibility Report
- Zoom
- Item Banks
- NU Reflect
- Quizzes
- Discussions
- Pages
- Files
- People
- BigBlueButton
- Outcomes
- Rubrics
- Collaborations
- Smart Search
- NCL Voice: Let's Talk
- Settings

Week 3: Machine Learning in IoT (24/25)

Please spread out your time across this week to attend the in-person session on 11/11 and two online sessions on 12/11 and 13/11, which together provide a comprehensive introduction to the TinyML:

- 11/11/2024 (in-person) **FDCG.06**
- 12/11/2024 (online) https://newcastleuniversity.zoom.us/j/83990895017
- 13/11/2024 (online) https://newcastleuniversity.zoom.us/j/83968574351

In the lectures, basic concepts around the following topics will be introduced:

- What is TinyML?
- What are the problems of OTA ML model updates?
- What is the role of emerging Artificial Intelligence (AI) and Deep Learning (DL) techniques in the context of IoT data processing?
- What are the new technical challenges when implementing AI and DL techniques on resource-constrained IoT devices and edge computing hardware?

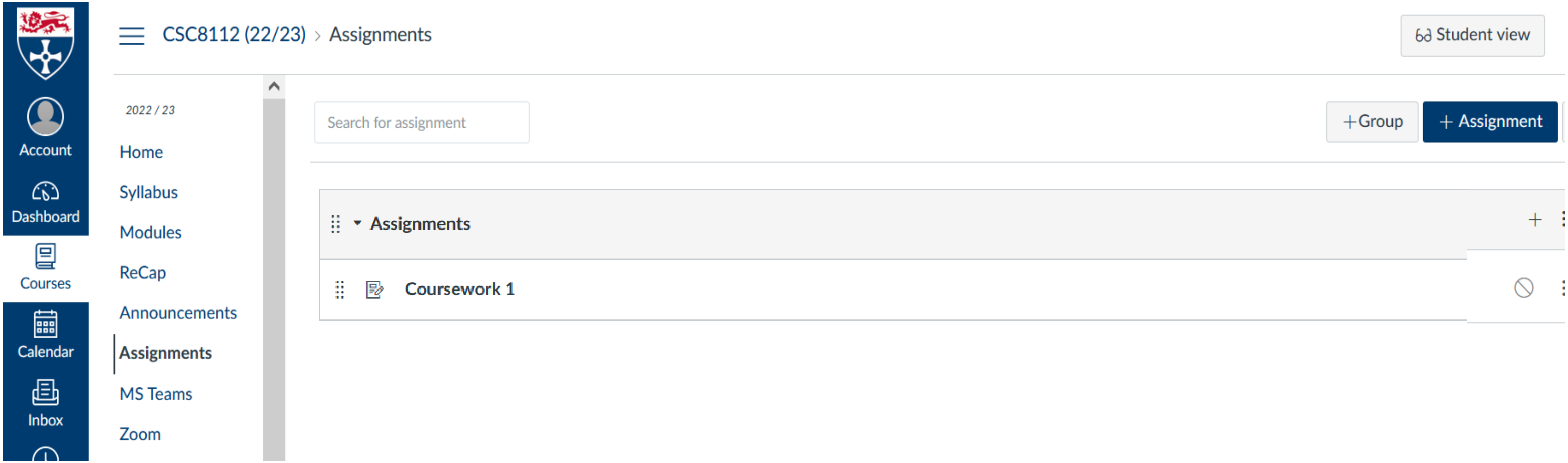
Below is the complete set of course materials for week 2.

- Lecture 6 [Machine Learning]
  - Lecture 6.1 [TinyML]
    - [Slides]
    - Video on ReCap
- Lecture 7
  - Lecture 7.1 [IoT Securiy]
    - IoT Security.pdf
    - Video on ReCap
  - Lecture 7.2 [Osmotic Computing (IoT) and Data Flow Control]
    - Osmotic_Dataflow.pptx
    - [Link to Video]
- Lecture 8
  - Lecture 8.1 [Edge AI]
    - [Slides]
    - [video1]
    - [video2]
  - Lecture 8.2 [TinyML model deployment]
    - [OTA-TinyML]
    - [Link to Video]

Additional Materials

- Lecture A [ML model development]

CSC8112 Canvas (Coursework)



Coursework (Assessment 1)-40%

CSC8112 - Internet of Things

Assessment 1

Assessment Overview

This assessment contributes 40% towards the total mark for this module. Out of this, 70% marks are assigned for implementation tasks and 30% marks are for the final report. It is an individual exercise: no group work is permitted for the assessment. You are advised to read and view all the instructional tutorial resources before you start implementing the solutions for the coursework Tasks 1 to 3. Each Task has been assigned a specific mark, which you will be awarded once you successfully demonstrate the completion of the same.

Once you complete Tasks 1-3, you will need to prepare the final Report (Task 4). **This coursework Report must be submitted on NESS by 3 pm on November 15, 2024.** In this Report, you will need to provide an in-depth discussion of how you implemented the solutions (e.g., code and commands) to solve Tasks 1-3. Additionally, you will be required to demonstrate successful executions of Tasks 1-3. Before the report submission deadline, you will be provided with a 30 mins slot to conduct **live demonstration**. In case of unforeseen disruptions (e.g., further lockdowns), we may also allow recorded demonstrations.

Coursework (Assessment 1)

A high-level picture showing the overall system design scope of the coursework is shown in the Figure 1, a short explanation of the components is given below:

IoT tier:

- *Newcastle Urban Observatory (NCL UO)* [`https://urbanobservatory.ac.uk/`]: The largest set of publicly available real-time urban data in the UK. NCL UO sensors are gathering data across Newcastle city. With over 50 data types and counting, there are lots of live data for you to access.

Edge tier:

- *Data Injector*: This will be a software component that you will design and implement in Task 1, focusing on (i) reading data from Urban Observatory API and (ii) transmitting data to the machine learning pipeline.

- *EMQX*: A broker of MQTT protocol, a message queuing system given to you as a Docker image, which forms the basis for enabling asynchronous service-to-service communication in a complex Machine Learning (ML)-based IoT data processing pipeline.

- *Data Preprocessing Operator*: A software component that you will develop in Task 2, Responsible for preparing training data of Machine Learning model.

Cloud tier:

- *RabbitMQ*: A cloud-based message queuing system.

- *Machine Learning Model/Classifier/Engine*: A software component that can be trained to predict particular types of future events.

- *Visualization*: A component that will visualize the trend of raw time-series data and the prediction results (input from the Machine Learning Model/Classifier/Engine).

After successfully completing the coursework, you will be able to gain hands-on experience in the following interrelated technology stacks, including:

- configuring a Docker-based IoT data processing pipeline;
- pulling images from the Docker Hub (a global repository of software components' images maintained by the developers);
- creating and deploying a self-contained IoT data processing service, which is often referred to as microservices;
- training a machine-learning-based predictor based on real-world data streams available from Newcastle Urban Observatory;

- implementing a machine learning-based air-quality prediction micro-service;
- visualizing time-series data using graphs;

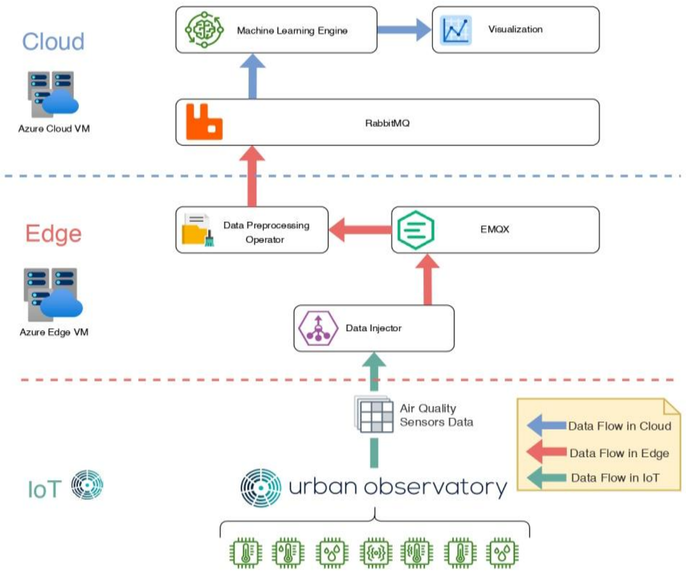


Figure 1: Overview

Pre-Requisites

Before starting with the course work, you are advised to carefully go through the training content covered in the Lecture q and extra supplements provided in the Yuque Document (am online document platform) at [`https://github.com/ncl-iot-team/CSC8112`]. Together, these provide in-depth details on:

- how to access and start Azure VMs, as shown in the Figure 2;
- how to download and run a docker image on Azure Labs;
- how to run your experiments on Azure Labs;

Coursework (Assessment 1)

- some hints for system structures of every task;

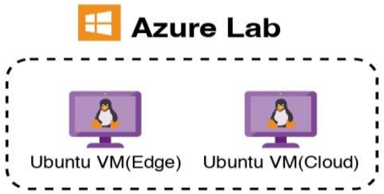


Figure 2: Relationship structure of Azure Lab and Ubuntu VMs.

Specification of Tasks

The coursework consists of 4 tasks. **Please note that Tasks 1-3 need to be done by both the command line and implementing the logic using Python language.**

Task 1: Design a data injector component by leveraging Newcastle Urban Observatory IoT data streams (20 Marks)

Task Objective : Understand and learn how to pull and run a Docker image from Docker Hub using the command line interface, how to collect real-world IoT data streams by invoking the Application Programming Interface (API) of Newcastle Urban Observatory, how to save data into the EMQX (a scalable MQTT broker for IoT applications), and how to re-compile and build Docker image using the command line and programmatic interfaces.

Hints : You are advised to carefully read and view the tutorial content relevant to task 1 that we have provided in the Yuque Doc [https://github.com/ncl-iot-team/CSC8112]. To download the EMQX docker image, please go to the following link [https://hub.docker.com/r/emqx/emqx]. To install python dependency package "requests" for sending HTTP request, please use the following Python package. [https://pypi.org/project/requests/]. Finally, the Phyton MQTT SDK ["paho.mqtt"] is available from [https://pypi.org/project/paho-mqtt/].

1. Pull and run the Docker image "emqx/emqx" from Docker Hub in the virtual machine running on Azure lab (Edge). Perform this task first using the command line interface (CLI).
2. Develop a data injector component with the following functions (Code) in Azure Lab (Edge) or the Azure Lab localhost:
   (a) Collect data from Urban Observatory platform by sending HTTP request to the following url ([http://uoweb3.ncl.ac.uk/api/v1.1/sensors/PER_AIRMON_MONITOR1135100/data/json/?starttime=20220601&endtime=20220831]). Following that please print out the raw data streams that you collected on the console.

(b) Although the raw air quality data you collected from the Urban Observatory API contains many metrics including NO_2, NO, CO_2, PM2.5, and PM10, among others, for the purpose of this coursework you only need to store and analyze PM2.5 data. While many meta-data are available for PM2.5 data, such as sensor name, timestamp, value, and location, you only need to store the metrics related to the Timestamp and Value meta-data fields.

(c) Send all PM2.5 data to be used by Task 2.2 (a) to EMQX service of Azure lab (Edge).

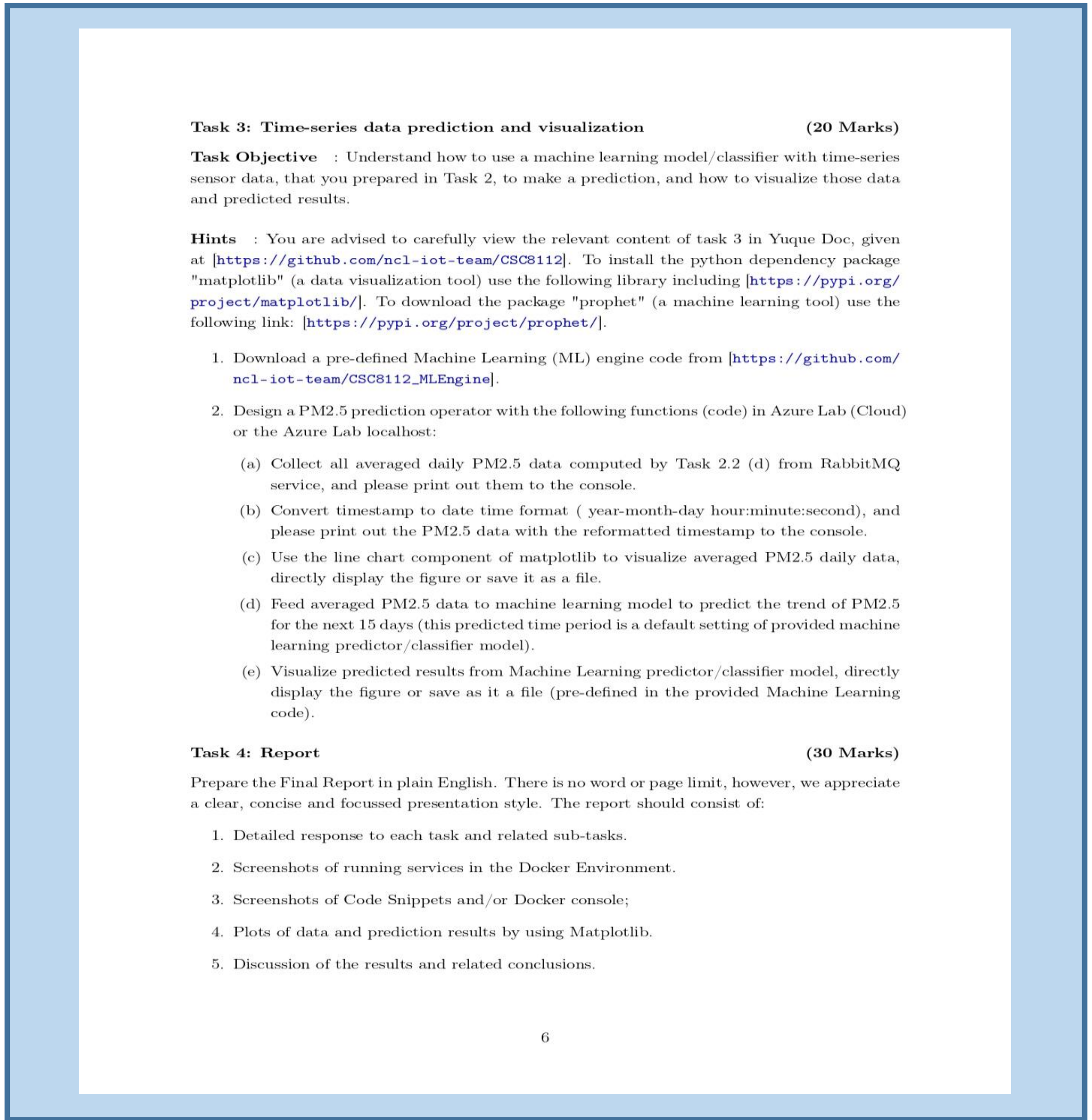
Task 2: Data preprocessing operator design (30 Marks)

Task Objective : Understand how to clean and prepare data for machine learning training by applying data processing operations, such as outliers cleaning and data reformatting. Moreover, you will also learn how to collect/send data from/to message queuing systems (e.g., EMQX and RabbitMQ), which are central to IoT data stream management. This task will also help you understand how native *Docker Compose techniques* can be leveraged to manage and deploy a complex IoT application stack/pipeline.

Hints : You are advised to carefully view the content relevant to Task 2 in Yuque Doc, which is given at [https://github.com/ncl-iot-team/CSC8112]. To install a python dependency package for sending messages to RabbitMQ(a message queue broker), please download "pika" from [https://pypi.org/project/pika/].

1. Define a *Docker compose file* which contains the following necessary configurations and instructions for deploying and instantiating the following set of Docker images (as shown in Figure 1) on Azure lab (Cloud):
   (a) Download and run RabbitMQ image (rabbitmq:management);
2. Design a data preprocessing operator with the following functions (code) in *Azure Lab (Edge)*:
   (a) Collect all PM2.5 data published by Task 1.2 (c) from EMQX service, and please print out the PM2.5 data to the console (this operator will run as a Docker container, so the logs can be seen in the docker logs console automatically).
   (b) Filter out outliers (the value greater than 50), and please print out outliers to the console (docker logs console).
   (c) Since the original PM2.5 data reading are collected every 15 mins, so please implement a python code to calculate the averaging value of PM2.5 data on daily basis (every 24 hours) and please print out the result to the console (docker logs console).
   (d) Transfer all results (averaged PM2.5 data) to be used by Task 3.2 (a) into RabbitMQ service on *Azure lab (Cloud)*.
3. Define a Dockerfile to migrate your "data preprocessing operator" source code into a Docker image and then modify the *docker-compose file* to run it as a container locally on the Azure lab (Edge). If you need the example code please refer to Yuque Doc, which is given at [https://github.com/ncl-iot-team/CSC8112].

Coursework (Assessment 1)



Task 3: Time-series data prediction and visualization **(20 Marks)**

Task Objective : Understand how to use a machine learning model/classifier with time-series sensor data, that you prepared in Task 2, to make a prediction, and how to visualize those data and predicted results.

Hints : You are advised to carefully view the relevant content of task 3 in Yuque Doc, given at [https://github.com/ncl-iot-team/CSC8112]. To install the python dependency package "matplotlib" (a data visualization tool) use the following library including [https://pypi.org/project/matplotlib/]. To download the package "prophet" (a machine learning tool) use the following link: [https://pypi.org/project/prophet/].

1. Download a pre-defined Machine Learning (ML) engine code from [https://github.com/ncl-iot-team/CSC8112_MLEngine].
2. Design a PM2.5 prediction operator with the following functions (code) in Azure Lab (Cloud) or the Azure Lab localhost:
   (a) Collect all averaged daily PM2.5 data computed by Task 2.2 (d) from RabbitMQ service, and please print out them to the console.
   (b) Convert timestamp to date time format (year-month-day hour:minute:second), and please print out the PM2.5 data with the reformatted timestamp to the console.
   (c) Use the line chart component of matplotlib to visualize averaged PM2.5 daily data, directly display the figure or save it as a file.
   (d) Feed averaged PM2.5 data to machine learning model to predict the trend of PM2.5 for the next 15 days (this predicted time period is a default setting of provided machine learning predictor/classifier model).
   (e) Visualize predicted results from Machine Learning predictor/classifier model, directly display the figure or save as it a file (pre-defined in the provided Machine Learning code).

Task 4: Report **(30 Marks)**

Prepare the Final Report in plain English. There is no word or page limit, however, we appreciate a clear, concise and focussed presentation style. The report should consist of:

1. Detailed response to each task and related sub-tasks.
2. Screenshots of running services in the Docker Environment.
3. Screenshots of Code Snippets and/or Docker console;
4. Plots of data and prediction results by using Matplotlib.
5. Discussion of the results and related conclusions.

CSC8112 (Assessment 2) Exam 60%

Online Video Lecture + PPT (Q2)

Question 2: The following sub-questions are related to the online lecture material.

(Q2.1) How do IoT communication protocols differ in their energy consumption and communication range capabilities? [5 marks]

(Q2.2) Which IoT communication protocol or combination of protocols would you recommend if you were given a task of deploying a landslide IoT sensor (tier 1) and an edge gateway (tier 2) in the Scottish Highlands (e.g. Inverness)? The landslide sensor and edge gateway will need to stream real-time data to a Cloud Datacentre (tier 3) hosted by Newcastle Urban Observatory. [5 marks]

(Q2.3) In the context of Q2.2, which data analysis strategy would you consider if you would like to minimise the number of data packets that edge gateway (tier 2) needs to send to Cloud Datacentre (tier 3). [5 marks]

CSC8112 (Assessment 2)

Online Video Lecture + PPT (Q5)

Question 5: The following sub-questions are related to the online lecture material.

(Q5.1) How is Osmotic Computing related to Edge and Cloud Computing? [5 marks]

(Q5.2) Assume you want to develop a deep learning model which should be capable of detecting the number of cars from images received from a network of CCTV cameras. Next assume that image volume and velocity is too large for undertaking training on one IoT or Edge device.

- What strategies would you use for accelerating the training and detection phase? [5 marks]

(Q5.3) In an IoT network, we have a set of devices that are continuously generating data, and these data are first injected to a set of edge datacentres (EDC) and then forwarded to a cloud data centre by the corresponding edge devices in the EDC.

- How can we improve the throughput of forwarding these data from devices to a cloud data centre? [5 marks]

(Q5.4) If Newcastle University wants to establish a blockchain network for sharing staff payroll information across various departments (e.g. finance, human resource, and estates), then what kind of blockchain will it be? Why? [5 marks]

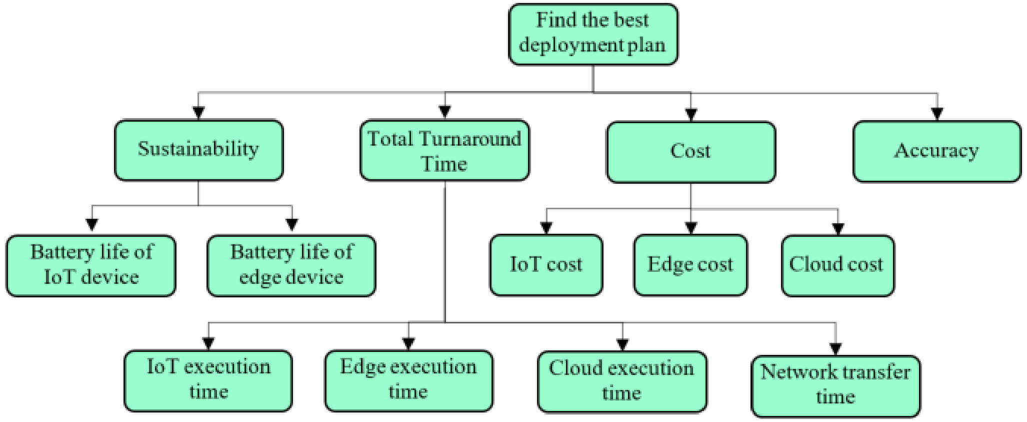
Online Video Lecture + PPT (Q5)

(Q5.5) Let us assume an IoT network of 15 devices, and they are making one-to-one communication.

- How many keys are required for this network when they use symmetric cryptography? [2.5 marks]
- How many key pairs are required for this network when they use asymmetric cryptography? [2.5 marks]

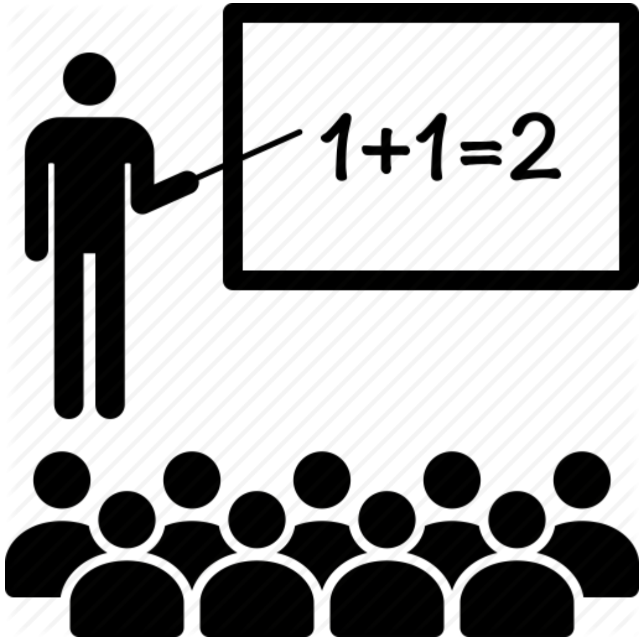
(Q5.6) Consider a case when we have 10 low-level QoS/non-functional attributes for making a deployment decision. Using the AHP-based Multi Objective Optimization method, we can decompose decision attributes into two level attributes (with 4 at level 1 and those remaining at level 2) as shown in Figure 1.

- Compute the time complexity for **calculating weights** using the AHP-Based Multi Objective optimization (ABMO) method. [5 marks]
- Also compare the complexity with traditional methods and show how much gain/loss is achieved, percentage wise. [5 marks]



LEARNING OUTCOMES

- Skill 1st: Know fundamentals of IoT.
  - A basic skill for anyone who stays in IT industry in this digital-era.
- Skill 2nd: Know how to design and deploy IoT application architecture if you are given the technology choices & resources.
- Skill 3rd: Know the complexity of IoT Networking and Data Management
- Skill 4th: Be able to provide consultancy on IoT to businesses, city councils, and hospitals
  - Critical for anyone who want to become "IoT consultant"



Only if you put enough effort in it