# 1. Introduction

## 1.1 Task 1

The objective of this experiment is to establish an edge server for data collection and storage, using a RaspberryPi to measure and transmit temperature and humidity data via the IPv4 protocol. Key concepts applied in this experiment include socket programming and network protocols, encoding and decoding data for network use, and evaluating the benefits and limitations of edge servers in data acquisition. Python was used to implement a cyber-physical system that enabled the integration of hardware and software for effective data collection. This experiment provided a practical and hands-on experience to real-time data handling in a networked environment as well as the practical applications of edge computing in IoT and cyber-physical systems.

**1. What is a Raspberry Pi? Why did you use it? Which version did you use?**

A RaspberryPi is a small computer often used for small-scale and real-time control system projects. For this experiment, the RaspberryPi 4 was used. It takes digital inputs and process the data using programmable scripts that can be written and modified using Python. It is important to note that the RaspberryPi can also connect to the internet to be able to connect to a network to transmit data.

**2. What is socket programming, a server, and a client?**

Socket programming enables the establishment of communication links within a network, allowing multiple processes to execute across interconnected devices. Through this setup, data is transmitted and received seamlessly over the network, supporting both remote and local operations. In this configuration, a server hosts various processes and fulfills client requests within the network, supplying the necessary data for execution and analysis.

**3. Which protocol did it use in socket programming? What are the benefits of using that protocol, and what are other protocols that can also be used?**

* The protocol used was IPv4 and used the IPv4 address of the client computer which the data was transmitted to.
* IPv4 Benefits:
  + Widely used
  + Heirarchal addressing
  + Simple addressing
* IPv6 is the next iteration of the Internet Protocol. It offers more address space while keeping the simplicity and efficiency of IPv4.

**4. Which sensor did you use for the experiments, and what does it do? What are GPIO pins? What is "ground" in GPIO pins?**

* The sensor used was the Adafruit DHT11. It is a capacitive temperature and humidity sensor.
* GPIO stands for general purpose input/output.
* Ground is needed to complete the circuit and ensure the sensor gets power. The other two pins are for the 5V power and signal that is sent to the RaspberryPi.

**5. What do "encode" and "decode" mean in the provided code? Why are they used here?**

* Encoding is used before the data is sent out over the network. When the data is received by the client, the data is then decoded before the data is added to the dataframe.
* The data collected is not in raw bytes. To transfer data over a network, the information needs to in raw bytes. Here, the data is converted to UTF-8 which is then sent through the IPv4 protocol.

**6. Apart from socket programming, what are other ways to transmit data?**

**7. Why is 'Label' added in the data? What do 'valid' and 'invalid' mean? Which value represents 'valid' and 'invalid'?**

**8. What is an edge server and what are advantages of using an edge server?**

## 1.2 Task 2

The objective of this experiment is to conduct data analysis on collected data by applying machine learning techniques to evaluate cyber-physical system security. Key concepts applied include machine learning for data-driven analysis using Python, along with required preprocessing steps to ensure that data is standardized and uniform for both training and validation. The experiment also investigates the performance variances across different machine learning algorithms, assessing each method's strengths in detecting the tampered data hidden within the dataframe. By analyzing performance metrics, this study evaluates the effectiveness of machine learning in identifying potential flaws within cyber-physical systems, and its security.

1. **What is Anaconda? What benefits does Anaconda provide?**

**2. What is a missing value, null, or duplicate? Why do we need to clean them?**

**3. Why do we need to split our data into training and testing sets?**

**4. Why do we need scaling? What are the existing scaling techniques? What are the drawbacks of the min-max scaling technique?**

**5. Which machine-learning model did you choose? Why?**

**6. How did you evaluate your machine-learning model? Present your results in figure or table.**

# 2. Experiments

## 2.1 Task 1

**Edge Server and Data Acquisition Procedure:**

1. Set up the RaspberryPi to read sensor data
2. Setup the socket and define the IP address and port to send to in order to connect to the edge server.
3. Encode the real-time data and send it through the server.
4. On the other computer, setup the socket to listen to the server using the same port as defined earlier. Using the same port ensures that the data is sent and received to the proper devices.
5. The humidity and temperature sensor measurements will then be recorded by the RaspberryPi, and the data will be encoded and sent through the network to the IP address and port.
6. The computer will receive the data via the matching port and decode it for recording within a dataframe.

**Data Tampering Simulation:**

1. The false data will be inserted by adding values to the dataframe CSV.

**Challenges**

The experiment was faced with two main challenges: unstable internet connectivity and slow data speeds. Intermittent connection issues led to interruptions in data transmission, causing loss of data and requiring multiple restarts of the experiment. The edge server code lacked functionality to retain collected data, while the Raspberry Pi did not store a local backup, preventing continuous data collection during connectivity loss. Once the connection recovered, data could not be sent afterwards. Additionally, the slow data speed presented further difficulties. The edge server only received data at intervals of 3-6 seconds, limited by the DHT11 sensor's 0.5 Hz sampling rate. These delays hindered the ability to track rapid changes in temperature and humidity in real time. The implications emphasize the need for more reliable network solutions and improved data retention methods to ensure accurate and continuous data collection, addressing the limitations that arise from network instability and slow data sampling.

# 3. Performance Evaluation

Task 1

Task 2

Algorithm 1

Algorithm 2

Algorithm 3

# 4. Conclusion

# 5. Code and Dataset

* GitHub repository