ENDG 510

Cyber-Physical Systems Engineering

Lab 1

November 7, 2024

Christian Daniel Magsombol, 30090792
Jeffrey Ramsay, 30061357

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 application 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 edge server computer to which the data was transmitted to. IPv4 is beneficial as it is the most widely used protocol for internet communication, providing broad compatibility and integration with most network systems. This ensures that IPv4 remains a reliable standard, supported by many devices, routers, and platforms. Additionally, the hierarchical addressing of IPv4 enables efficient IP organization into networks and sub-networks, optimizing data routing and address management. Furthermore, IPv4's simple 32-bit addressing scheme facilitates easy implementation, supporting compatibility across diverse network hardware and software.

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 with a range of 0-50 degrees Celsius and 20-80% humidity respectively. It has a peak sampling rate of 1 Hz though it is mostly sustained at 0.5 Hz.

GPIO stands for general purpose input/output. These pins are what feed measured data from the sensor and into the RaspberryPi. It is general purpose as any compatible digital signal can be inputted into the RaspberryPi or outputted. Ground is needed to complete the circuit and ensure the DHT11 sensor, or other sensors, are provided with power. The other two pins are for the 5V power and for the digital signal.

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, and thus, it 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?**

For similar applications in the experiment, Bluetooth can be used to transmit data wirelessly between two nearby devices. Bluetooth experiences less interference than Wi-Fi networks and is more efficient. Wireless networks such as LTE and 5G can also be used for larger range distances.

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

It is important to label the data to let the machine-learning algorithm properly learn the differences in the data and validate its predictions during testing. Data labelled with 1 are considered “valid”, and 0 is considered “invalid”. This indicates the real and fake data, respectively. During training, the machine-learning algorithm will use these labels to verify if their predictions are accurate.

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

The edge server processes and stores received data locally and is used for nearby communication between IoT devices. It is beneficial to use an edge server rather than a cloud server as latency is reduced, data post-processing can be done immediately as it is stored locally, and data spends less time in transit due to nearby physical distances which means data is less likely to be intercepted.

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?**

Anaconda is an open-source package and environment system. It provides many tools and environments for data analysis and visualization within one very convenient package.

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

Null, missing values show up as rows in the CSV that have no data. Duplicates are rows that have identical data to other rows. We need to clean the null data to ensure that the ML model is not learning incorrect data, and we need to clean duplicates to reduce the chances of overfitting or increased training time due to the large amount of data.

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

We split the data into training and testing sets so that we can accurately judge the accuracy of a model before it is put into production on data that we do not know whether it is real or fake. By splitting the data into training and testing sets, we can train the model with a subset of the collected (known) data, and then test it on the remainder of the known data.

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

Scaling is important as it prevents large variables from disproportionately contributing to results. Existing scaling techniques include standardization, normalization, and min-max scaling. The main drawback of min-max scaling is that it's relatively sensitive to outliers, which can impact the minimum and maximum values used for scaling.

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

Three machine learning models were chosen: KNeighborsClassifier, LogisticRegression, and GaussianNB. They were chosen as they are all relatively simple and effective algorithms. LogisticRegression should be good as it is specifically intended for binary classification which is what we have. GaussianNB seemed like a good option as it is a fast algorithm that works well with small amounts of training data. KNeighborsClassifier was chosen as it was there by default and seemed like a good option due to its ability to work with binary classifications and ability to work with a wide range of dataset sizes.

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

The machine-learning models were evaluated by comparing their accuracy when performing predictions on the test data sets. All models were found to have perfect accuracies, which is likely due to our fake data being significantly different to the legitimate data, making it easy for any algorithm to predict. The accuracy is summarized in the following table.

Accuracy Comparison for Classification Models

Algorithm	Accuracy
KNeighborsClassifier	1.0
LogisticRegression	1.0
GaussianNB	1.0

2. Experiments

2.1 Task 1

Edge Server and Data Acquisition Procedure:

1. Set up the RaspberryPi to communicate with the Adafruit DHT11 sensor.
2. Setup the socket and define the IP address and port 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 randomly 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 RaspberryPi 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.

2.2 Task 2

Machine-Learning Classification Model on IoT Data:

1. Preprocess the data to ensure that there are no null values, redundant data, or missing entries.
2. Split the data into a training set and validation set.
3. Standardize the data to ensure uniformity.
4. Using the training data, develop three machine-learning models that use classification algorithms.
    a. KNeighborsClassifier
    b. Logistic Regression
    c. GaussianNB
5. Use the validation data to assess the performance of the model. For this experiment, accuracy was chosen as the metric to compare between each model.
6. Save the training models.

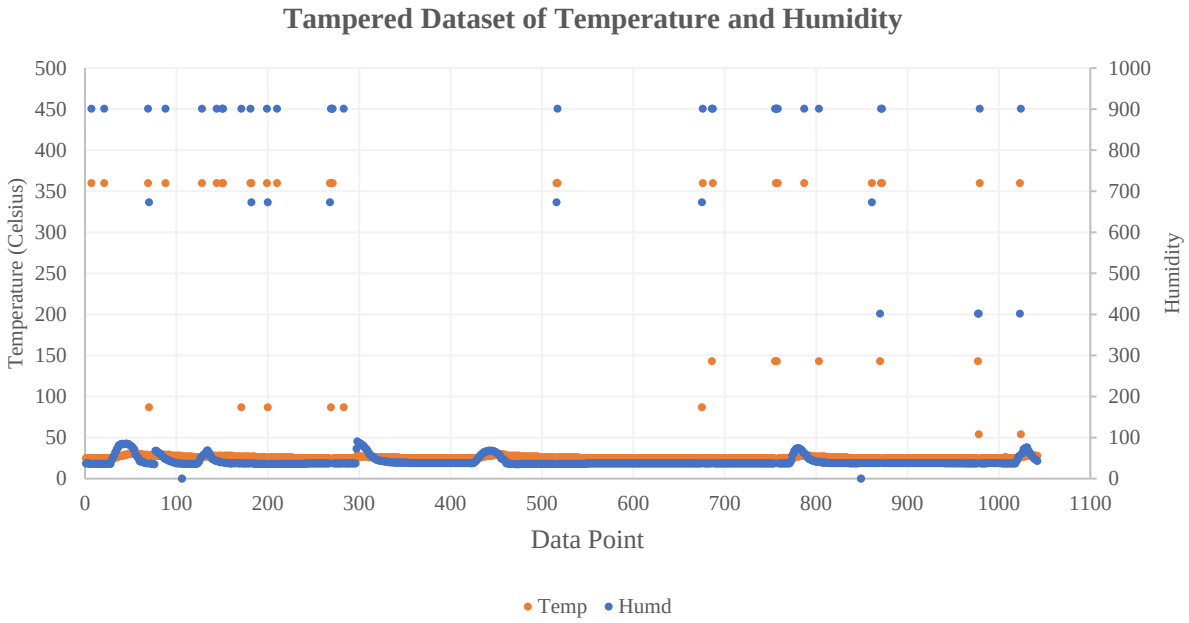
Challenges

This part of the experiment involved the challenge of selecting an appropriate machine-learning model and the principles behind how they work. With all the options for machine-learning models, choosing three relevant models and understanding their underlying methods for this specific application was challenging. Furthermore, with such a simple dataset, it was difficult to develop a higher-level justification as to how certain models are more advantageous over others. It may be more applicable for more complex datasets and task requirements, but the implications were negligible for the scale of the experiment.

3. Performance Evaluation

3.1 Task 1

The tampered data is plotted below to visualize the data points that include invalid data as well as data points with missing entries.



It is evident that the invalid data is not representative of the real data, and thus, it can easily be identified by the classification models used for Task 2. Both the temperatures and humidity are too high to be hidden within the real data.

3.2 Task 2

Both the data preprocessing and classification models can be evaluated in the second experiment. Within the code, the sum of null values were taken and outputted prior to training the model using the dataset. The accuracy of each algorithm was also taken as an output to assess performance.

Data Preprocessing

The following code was used to identify missing values.

```
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)
```

This was the following output which verified that there were missing values.

```
Missing Values:
  Temp     0
  Humd     0
  Label    0

dtype: int64
```

To remove duplicated values and effectively scale the data, the following line was used.

```
df = df.drop_duplicates()
```

Therefore, the data was successfully preprocessed.

Performance Metrics

To assess the performance of the classification models, the accuracy each model was recorded. Each of the algorithms achieved perfect accuracy as summarized in the following table.

Accuracy Comparison for Classification Models

Algorithm	Accuracy
KNeighborsClassifier	1.0
LogisticRegression	1.0
GaussianNB	1.0

This does lead to some suspicion as it is not often that models are always 100% accurate, let alone all three algorithms resulting in the exact same performance. However, it is important to note that even after postprocessing the data, there is still a vast distinction between the valid and invalid data. Evident in the plot evaluated in Task 1, most temperatures recorded for the valid data remained within a window of 25°C to 30°C and humidity levels never exceeded 87%. The invalid data had far larger values. Additionally, since there were few unique invalid data points added, scaling the data made it even easier for the algorithms to identify which points were invalid as all the duplicate invalid points would have been removed even before the training began. Thus, it can be said that despite a 100% accurate model regardless of the algorithm, the classification models trained were valid and that the preprocessing was beneficial in filtering data that does not fit valid trends. A more complex dataset would yield different results especially if the invalid data was more carefully selected.

4. Conclusion

This lab experiment demonstrated the practical application of IoT devices in data acquisition and the role of machine learning in identifying security faults in cyber-physical systems. The first task highlighted the use of edge servers in collecting data from nearby local network devices, emphasizing the importance of correctly configuring ports, encoding, decoding, and implementing measures to mitigate network issues and data loss. The second task explored machine learning's effectiveness in detecting cybersecurity vulnerabilities, particularly when an attack yields distinct data patterns. While the differences between machine learning algorithms were minimal due to the simplicity of the dataset, proper preprocessing ensured that each algorithm produced valid and effective results. Overall, it is important to understand these concepts to ensure that IoT devices utilize physical systems that mitigate data interception within a cyber-physical system such as using edge networks, and that datasets are robust and provide machine-learning algorithms with enough information to be able to evaluate more realistic scenarios of cyber-security issues.

5. Code and Dataset

GitHub Repository: https://github.com/cdmags/ENDG-510-cdmgs/tree/main/Lab%201

Group Members	Contribution	Details
Christian Daniel Magsombol	50%	Task 1 - Experiment
		Task 1 - Questions
		Report
Jeffrey Ramsay	50%	Task 2 - Experiment
		Task 2 - Questions
		Report