"Sensor Data Analysis of IoT"

Report of the Project

The IoT project work mainly consists of three components and they are- Data Collection, Data Plotting, Data Analysis. The details of each component have included in the following sections.

1. Data Collection

In the project guidelines, it is specified that we have to collect humidity and temperature data for continuously two hours on five different days. So, Adam and I have collected the data from April 5th – April 9th, 2021 at Adam's house at 7.00 pm to 9.00 pm. To keep consistency, we have collected the data 5 days in a row in a same time. We have collected the temperature and humidity data using DHT11 from ESP32 board and then transmitted data over the network via MQTT. DHT11 is a temperature-humidity sensor and ESP32 is a microcontroller chip with integrated Wi-Fi. On a Raspberry Pi 4 running MQTT server, a MQTT server was operating. The Raspberry Pi also served as an active MQTT client and it was listing for data from the ESP32 and sending it to the cloud with a timestamp. We have utilized an open-source cloud platform firebase to store our data. Firebase is a free cloud platform and it has a database named Firestore. The code of ESP32 and MQTT has been included in the dropbox. After storing the data into firebase, I have exported all the data from firebase to a csv file for the next sections on the project.

2. Data Plotting

The next step in the project guideline is data plotting. I have used the collected data to get the plots. Also, I have used some python libraries for plotting like I have used seaborn library for data visualization, pandas library for manipulating numerical tables, time series and matplotlib library for numerical data visualization. Fig 1 is the line plots for temperature and humidity data. Here the plots are showing 5 lines in each plot data corresponding with 5 days of data. Fig 2 is the box plots for temperature and humidity data. Here each box is representing one days of data. Fig 3 is the line grid plot for 3 days of data. The code of data plotting has been included in the dropbox.

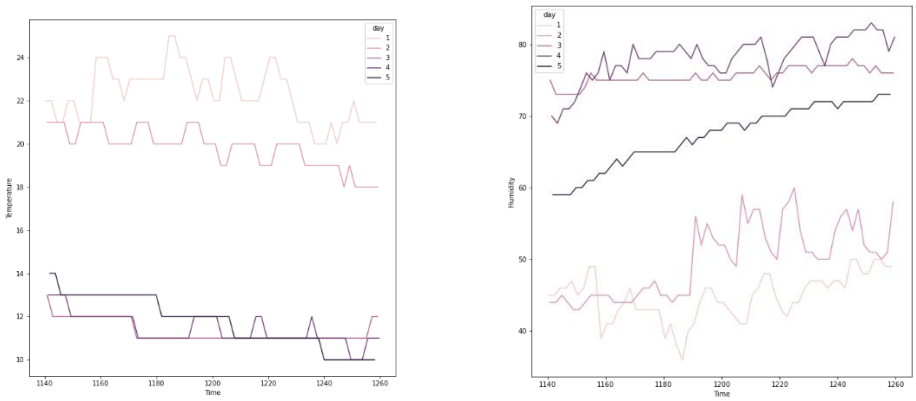

Fig 1: Plot 1 & 2 – Line plots for temperature and humidity data

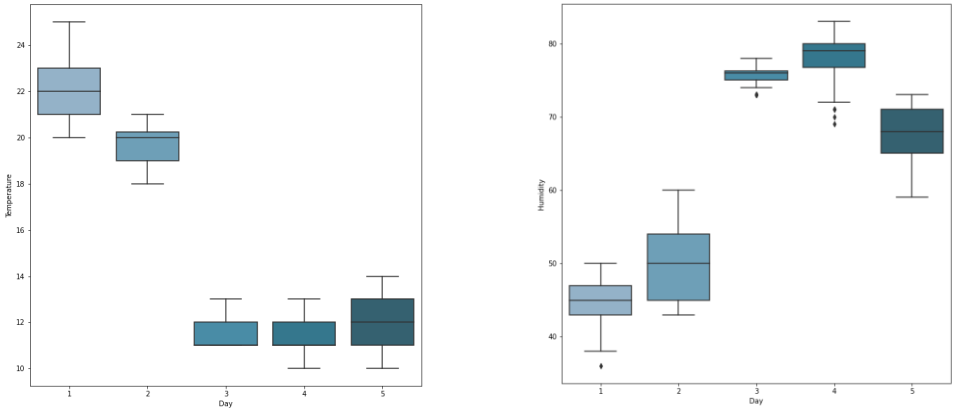

Fig 2: Plot 3 & 4 – Box plots for temperature and humidity data

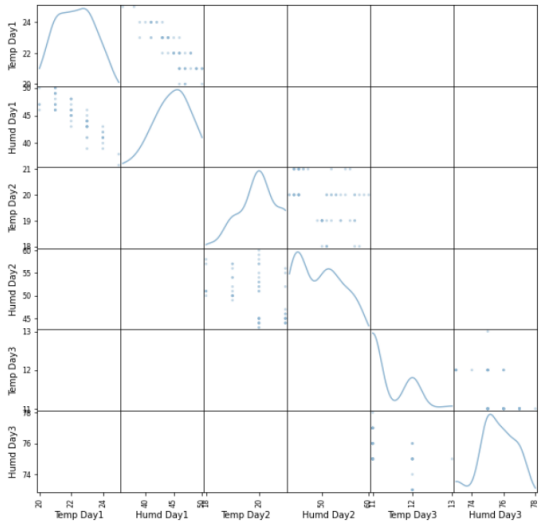

Fig 3: Plot 5 – Line grid plots for 3 days temperature and humidity data

3. Data Analysis

Data Pre-processing: Before starting data analysis, data pre-processing is required. To do the data pre-processing, first I have to look on the csv file. In the collected dataset, it has the following variables: timestamp, temperature, and humidity. Now I have to extract our time from the timestamp column and have to represent it in minutes form. To do that I have to use the attached command lines. I have created another column to extract the minutes and saved it in a different csv file which is also attached here.

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('firebase-1618885447171.csv')
df['minutes'] = df['time'].str.split(':').apply(lambda x: int(x[0]) * 60 + int(x[1]) + int(x[2])/60)
df.head()
```

```
df.head()
```

	day	temp	humidity	time	minutes
0	1	22	45	4/5/2021 19:00	1140.316667
1	1	22	45	4/5/2021 19:02	1142.316667
2	1	21	46	4/5/2021 19:04	1144.316667
3	1	21	46	4/5/2021 19:06	1146.333333
4	1	22	47	4/5/2021 19:08	1148.333333

Data analysis: With the guidelines of project, I have created linear regression model M1. I have used scikit learn linear regression model to build M1 model. M1 model is created between temperature and minutes. Then I have created another linear regression model between humidity and minutes and named this model M2. In the same way, another linear regression model between temperature, humidity and minutes where temperature is the response variable has been created with the name of M3. Then, I have computed mean squared error and AIC for M1, M2, and M3. Mean squared error (MSE) is an estimator which measures the

average of error squares like the average squared difference between the estimated values and the true value. It is a risk function and corresponding to the expected value of the squared error loss. It is always a non-negative and values close to zero are better. AIC is also an estimator of prediction error and there by relative quality of statistical models for a given set of data. AIC estimates the quality of each model from a given collection of models for the data. Fig 4, 5 and 6 are the plots of the predicted and actual values of the response variable for all the three models. From the plot we can see that for Model M1 and M2 the predictions are not that much good but for model M3 prediction is good. If the blue line is closer to the red values then we can believe that our model is predicting respectably.

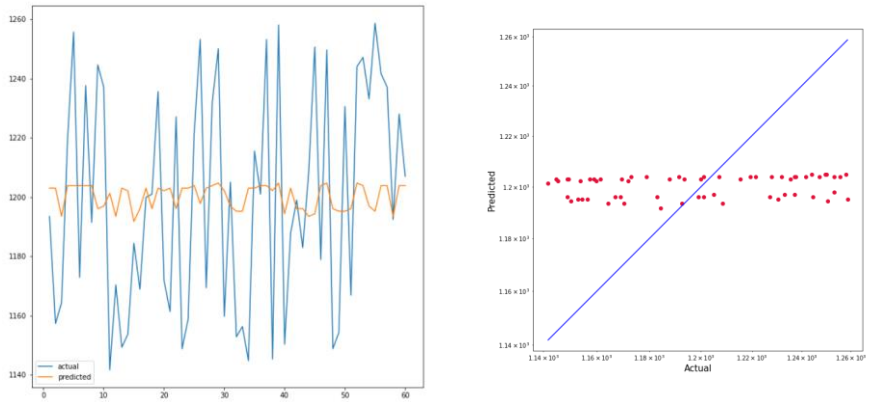

Fig 4: Predicted and actual value of M1 model

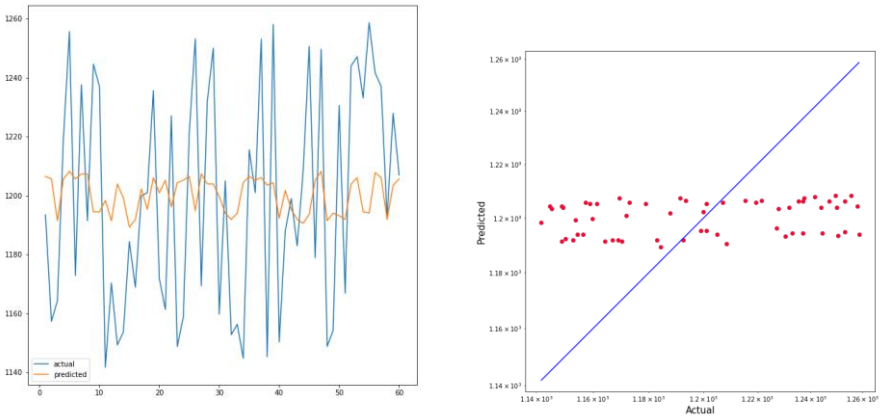

Fig 5: Predicted and actual value of M2 model

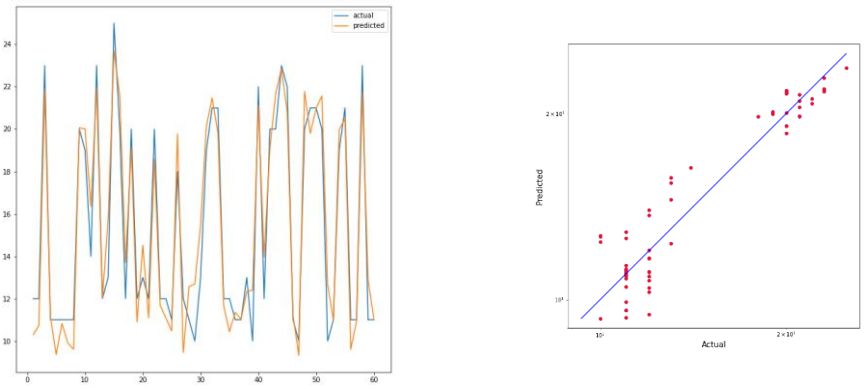

Fig 6: Predicted and actual value of M3 model