EE250 Final Project: End-to-end IoT System Richard Oh, Ellen Ko

---

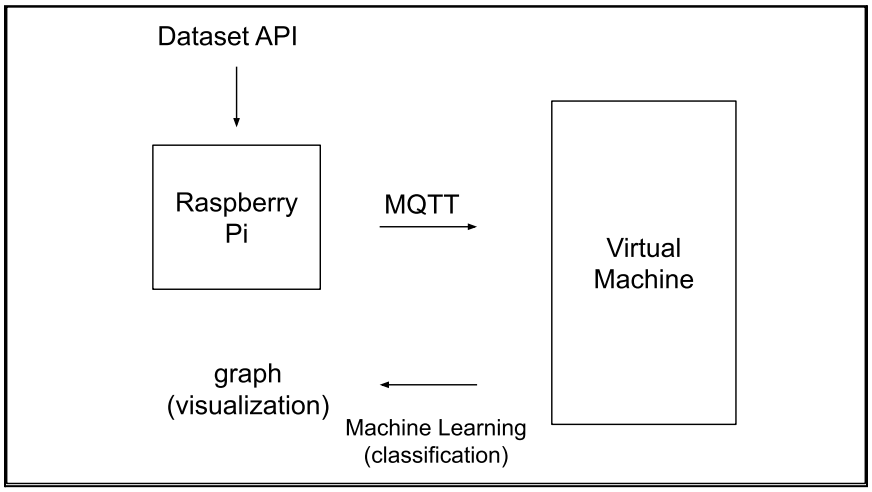
Github Repository: https://github.com/ohrjh10/Richard_Oh_Ellen_Ko_EE_250_Final_Project

Objective:
Building our own end-to-end IoT system using multiple concepts from the labs, where an end-to-end IoT system is defined as one or more physical nodes connected to a central node for data collection.processing, control, or visualization.

Brief Outline:
The proposed plan for this project is to design a Machine Learning algorithm associated with the use of API and MQTT protocols. Raspberry Pi will be used as one physical node for this project, and its functionality is to load the API data set (server side) and to send the obtained data set to the second physical node, which is the Virtual Machine (client side). During this process, the data set will be converted into a .csv file which is another functionality of the python file on the server side. The Machine Learning code from the Virtual Machine then will be able to classify the data set and give us a visualization of the result with the Jupyter Notebook application.

Components:
Physical nodes used: Raspberry Pi, Laptop(Virtual Machine)
Protocols, techniques used: MQTT, Machine Learning
Data Collection: data will be collected from a virtual sensor, which will use public data API from https://openweathermap.org/api (Weather API)
Machine Learning aspect: classification of weather based on the data collected by the weather API. The Machine Learning code will classify whether the weather is moderate or not
Node-to-node communication: The Raspberry Pi sending the data from the API to the VM
Visualization: the scatter plot of the classification will be shown towards the end of the demo

Block Diagram:



Reflection:

Limitations and lesson learned: An assumption that was made with the Machine Learning process was that humidity and temperature are factors that affect the weather to great extent compared to other factors such as wind speed or visibility. However, the data set collected from cities in real life suggests that the weather patterns cannot just be classified using two parameters, as weather patterns are considered extremely inconsistent. Therefore, the original scatter plot shown in the start of the Jupyter Notebook did not show a general trend and thus made it harder to classify them using the data set we have obtained from the weather API. In the end, this resulted in a low accuracy of the Machine Learning model. This made us recognize that the Machine Learning model could be improved by adding more parameters in order to see a general trend of the parameters and the end result weather condition. In addition, having more data points would possibly lead to an increase in the accuracy of the model as more data points can be used for training and testing.

External Sources referenced:

Weather API - openweathermap. (n.d.). Retrieved May 6, 2023, from https://openweathermap.org/api

IJRASET Journal for Research in Applied Science and Engineering Technology. (n.d.). Retrieved May 6, 2023, from https://www.ijraset.com/research-paper/prediction-and-classification-of-weather-using-ml