

National Textile University, Faisalabad



Department of Computer Science

Name:	Muhammad Hassaan Raza
Class:	BSAI-6th
Registration No:	22-NTU-CS-1362
Assignment	Lab13-Lab/Home tasks
Course Code:	AIE-3079
Course Name:	Internet of Things Fundamentals
Submitted To:	Mr. Nasir Mehmood
Submission Date:	20-May-2025

LAB 13 REPORT

TASK 1: ARDUINO MQTT PUBLISHER FOR DHT SENSOR

What I Did:

I used Arduino code to send temperature and humidity data from a DHT11 sensor to an MQTT broker (Mosquitto). The ESP32 connects to Wi-Fi and publishes data every second.

Results:

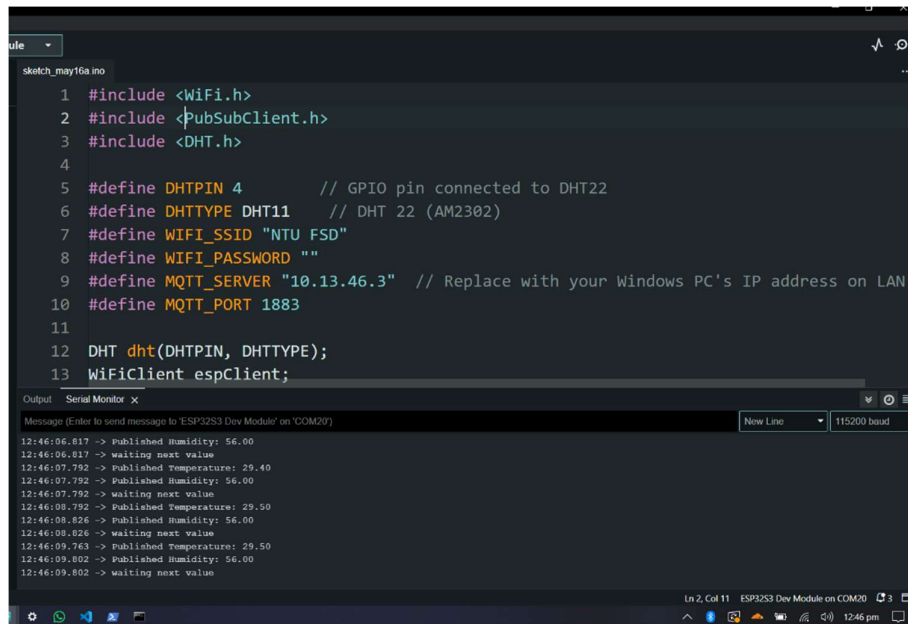
- The Serial Monitor showed live updates like:
Published Temperature: 29.40°C
Published Humidity: 56.00%
- The data was sent successfully to the topic esp32/dht/temp and esp32/dht/hum.

Problems Faced:

- Sometimes, the sensor failed to read values, but the code handled it by printing an error.

Screenshot Attached:

[illegible]

The image shows a screenshot of an Arduino IDE window. The top pane displays a sketch named 'sketch_may16a.ino' with the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <DHT.h>
4
5 #define DHTPIN 4 // GPIO pin connected to DHT22
6 #define DHTTYPE DHT11 // DHT 22 (AM2302)
7 #define WIFI_SSID "NTU FSD"
8 #define WIFI_PASSWORD ""
9 #define MQTT_SERVER "10.13.46.3" // Replace with your Windows PC's IP address on LAN
10 #define MQTT_PORT 1883
11
12 DHT dht(DHTPIN, DHTTYPE);
13 WiFiClient espClient;
```

The bottom pane shows the 'Serial Monitor' window, which is set to 'COM20' and '115200 baud'. It displays the following output:

```
12:46:06.617 -> Published Humidity: 56.00
12:46:06.617 -> waiting next value
12:46:07.792 -> Published Temperature: 29.40
12:46:07.792 -> Published Humidity: 56.00
12:46:07.792 -> waiting next value
12:46:08.792 -> Published Temperature: 29.50
12:46:08.826 -> Published Humidity: 56.00
12:46:08.826 -> waiting next value
12:46:09.763 -> Published Temperature: 29.50
12:46:09.802 -> Published Humidity: 56.00
12:46:09.802 -> waiting next value
```

TASK 2: STORE MQTT DATA IN INFLUXDB

What I Did:

I ran a Python script (1-dht_data_only.py) to subscribe to MQTT topics and save the sensor data to InfluxDB.

Results:

- The script printed confirmations like:
Received Temperature: 29.40°C
Data written to InfluxDB: Temperature: 29.40°C, Humidity: 56.00%
- Data was stored in InfluxDB under the measurement dht_data with tags and fields.

Problems Faced:

- First, the script didn't connect because of a wrong IP address. I fixed it by checking the broker address.

screenshot Attached:

The screenshot displays two parts of a development environment. The top part is the 'Data Explorer' interface, which shows a table of sensor data. The table has columns for time, value, field, measurement, and device. The data is filtered for 'lab13' and 'sensor_data'. The bottom part is a Jupyter Notebook showing a Python script that connects to an MQTT broker and processes incoming data. The script includes functions for connecting to the broker, subscribing to topics, and handling incoming messages. The terminal output shows the script running successfully and receiving data from the MQTT broker.

Data Explorer Table:

_start	_stop	_time	_value	_field	_measurement	device
2025-05-16 12:54...	2025-05-16 12:59...	2025-05-16 12:54...	51	humidity	dht_data	esp32
2025-05-16 12:54...	2025-05-16 12:59...	2025-05-16 12:54...	51	humidity	dht_data	esp32
2025-05-16 12:54...	2025-05-16 12:59...	2025-05-16 12:54...	51.40	humidity	dht_data	esp32
2025-05-16 12:54...	2025-05-16 12:59...	2025-05-16 12:55...	51.70	humidity	dht_data	esp32
2025-05-16 12:54...	2025-05-16 12:59...	2025-05-16 12:55...	52	humidity	dht_data	esp32

Jupyter Notebook Code:

```
def on_message(client, userdata, msg):
    # Reset the values to avoid duplicate writes
    Temperature = None
    Humidity = None
    except Exception as e:
        print(f"Error processing message: {e}")

# Function to connect to MQTT broker and subscribe to topics
def on_connect(client, userdata, flags, rc):
    print(f"Connected to MQTT broker with result code {rc}")
    client.subscribe(MQTT_TOPIC_TEMP)
    client.subscribe(MQTT_TOPIC_HUM)

# Set up MQTT client
mqtt_client.on_connect = on_connect
mqtt_client.on_message = on_message

# Connect to MQTT broker
```

Terminal Output:

```
Received Temperature: 28.9°C
Received Humidity: 53.0%
Data written to InfluxDB: Temperature: 28.9°C, Humidity: 53.0%
Received Temperature: 28.9°C
Received Humidity: 53.0%
Data written to InfluxDB: Temperature: 28.9°C, Humidity: 53.0%
Received Temperature: 28.9°C
Received Humidity: 53.0%
Data written to InfluxDB: Temperature: 28.9°C, Humidity: 53.0%
Received Temperature: 28.9°C
Received Humidity: 53.0%
Data written to InfluxDB: Temperature: 28.9°C, Humidity: 53.0%
Received Temperature: 28.9°C
Received Humidity: 53.0%
Data written to InfluxDB: Temperature: 28.9°C, Humidity: 53.0%
```

TASK 3: TRAIN MODEL WITH NOISE

What I Did:

I ran 2-train_model_with_noise.py to create a synthetic dataset and train a neural network to classify temperature/humidity into 5 classes (e.g., "Hot and Dry").

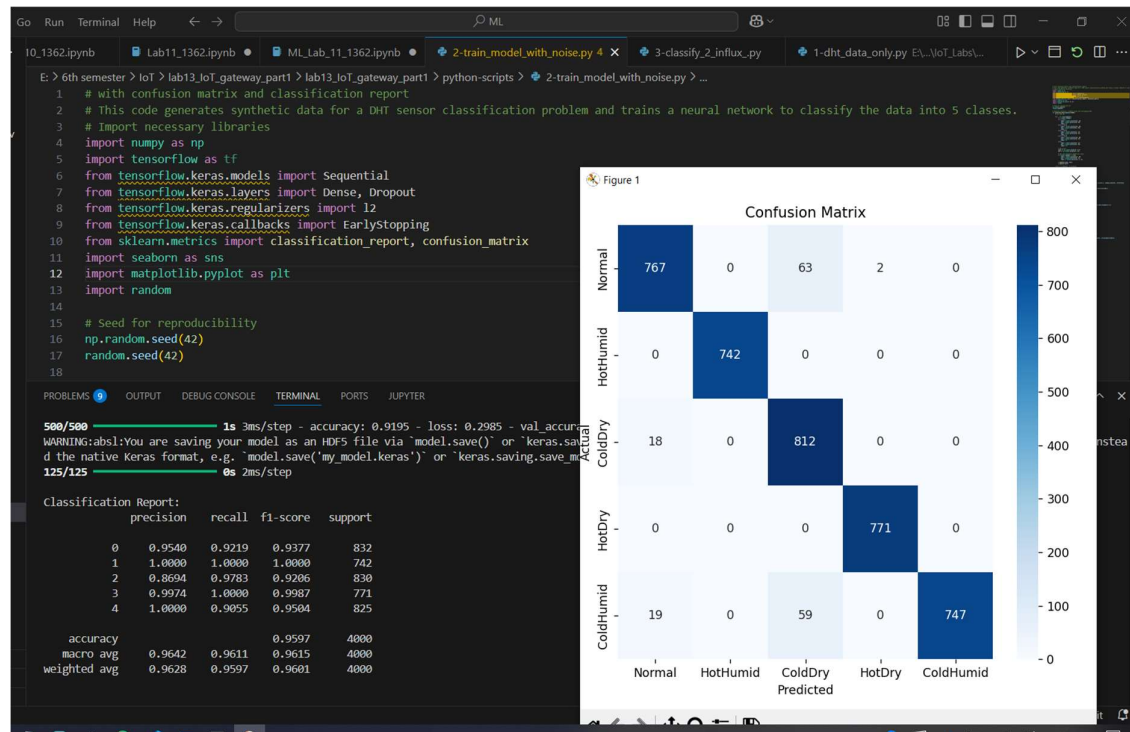
Results:

- The model achieved **~92% accuracy**.
- The confusion matrix showed good performance but some confusion between similar classes (e.g., "Normal" and "Cold and Humid").

Problems Faced:

- The training was slow on my laptop, but eventually it trained.

Screenshots Attached:



TASK 4: CLASSIFY AND STORE IN INFLUXDB

What I Did:

I executed 3-classify_2_influx.py to read live sensor data, classify it using the trained model, and save everything (temperature, humidity, and class label) to InfluxDB.

Results:

- The console showed real-time updates:
 - 🌡 Received Temperature: 29.40°C
 - 🔍 Predicted Class: Normal
 - ✅ Data saved to InfluxDB
- The InfluxDB dashboard displayed all fields: temperature, humidity, and class_label.

Problems Faced:

- At first, the class labels didn't appear in InfluxDB because of a typo in the field name. I fixed it by checking the script.

Screenshots Attached:

influxdb

admin
iotlab

Load Data

Data Explorer

Notebooks

Dashboards

Tasks

Alerts

Settings

Help & Support

Data Explorer

TableCUSTOMIZELocalSAVE AS

_time	class_label	humidity	temperature
2025-05-16 01:08:30 GMT+5	Normal	52	28
2025-05-16 01:08:31 GMT+5	Normal	52	28
2025-05-16 01:08:32 GMT+5	Normal	52	28
2025-05-16 01:08:33 GMT+5	Normal	52	28
2025-05-16 01:08:34 GMT+5	Normal	52	28

Query 1 (0.03s)View Raw DataCSVPast 5mQUERY BUILDERSUBMIT

```
1 from(bucket: "lab13")
2   > range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   > filter(fn: (r) => r["_measurement"] == "dht_data")
4   > filter(fn: (r) => r["_field"] == "class_label" or r["_field"] == "humidity" or r["_field"] == "temperature")
5   > filter(fn: (r) => r["device"] == "esp32")
6   > pivot(rowKey: ["_time"], columnKey: ["_field"], valueColumn: "_value")
7   > keep(columns: ["_time", "temperature", "humidity", "class_label"])
```

Filter Functions...Transformationsaggregate.ratechandelMomentumOscillatorcolumnscov

GoRunTerminalHelp

Lab11_1362.ipynbLab11_1362.ipynbML_Lab_11_1362.ipynb2-train_model_with_noise.py43-classify_2_influx_pyE:\Task_43-classify_2_influx_pyE:\p

E: > 6th semester > IoT > IoT_Labs > lab13 > Task_4 > 3-classify_2_influx_py > ...

1 import paho.mqtt.client as mqtt
2 from influxdb_client import InfluxDBClient, Point
3 from influxdb_client.client.write_api import SYNCHRONOUS
4 import tensorflow as tf
5 import numpy as np
6 import time
7 from datetime import datetime, timezone
8
9 # InfluxDB setup
10 INFLUXDB_URL = "http://localhost:8086" # InfluxDB server URL
11 INFLUXDB_TOKEN = "HyqbISWV2BP-DAFIwOCmm7Vdfz2Z1Dbv6DN0Micc9BSR4bjs-Wwz1a6zrusfQ8Y1KojU-VeguxvOrvImLnCbqQ==" # Replace with your InfluxDB token
12 INFLUXDB_ORG = "iotlab" # Replace with your InfluxDB organization name
13 INFLUXDB_BUCKET = "lab13" # InfluxDB bucket name
14
15 # MQTT setup
16 MQTT_BROKER = "10.13.46.3" # ESP32's MQTT broker address
17 MQTT_PORT = 1883 # MQTT port
18 MQTT_TOPIC_TEMP = "esp32/dht/temp"
19
20 # TensorFlow model setup
21 model = tf.keras.models.load_model('model.h5')
22
23 def callback(mqtt_client, obj, msg):
24 print(f"Received Temperature: {msg.payload}")
25 temp = float(msg.payload)
26 humidity = 52.00
27 class_label = "Normal"
28 point = Point("dht_data", device="esp32", class_label=class_label, humidity=humidity, temperature=temp)
29 write_api.write(bucket=INFLUXDB_BUCKET, organization=INFLUXDB_ORG, point=point)
30 print(f"Data saved: Temp={temp}, Hum={humidity}, Class={class_label}")
31 predicted_class = model.predict([temp, humidity])
32 print(f"Predicted Class: {predicted_class}")
33
34 client = mqtt.Client()
35 client.on_connect = callback
36 client.connect(MQTT_BROKER, MQTT_PORT, 60)
37 client.subscribe(MQTT_TOPIC_TEMP)
38 client.loop_start()
39
40 while True:
41 time.sleep(1)
42 client.loop_check_timeout()
43
44 client.loop_stop()

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTSJUPYTER

Python

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.5 64-bit