

Lab 13 – IoT

Gateway & Machine Learning Report

Name: Hasham UL Haq

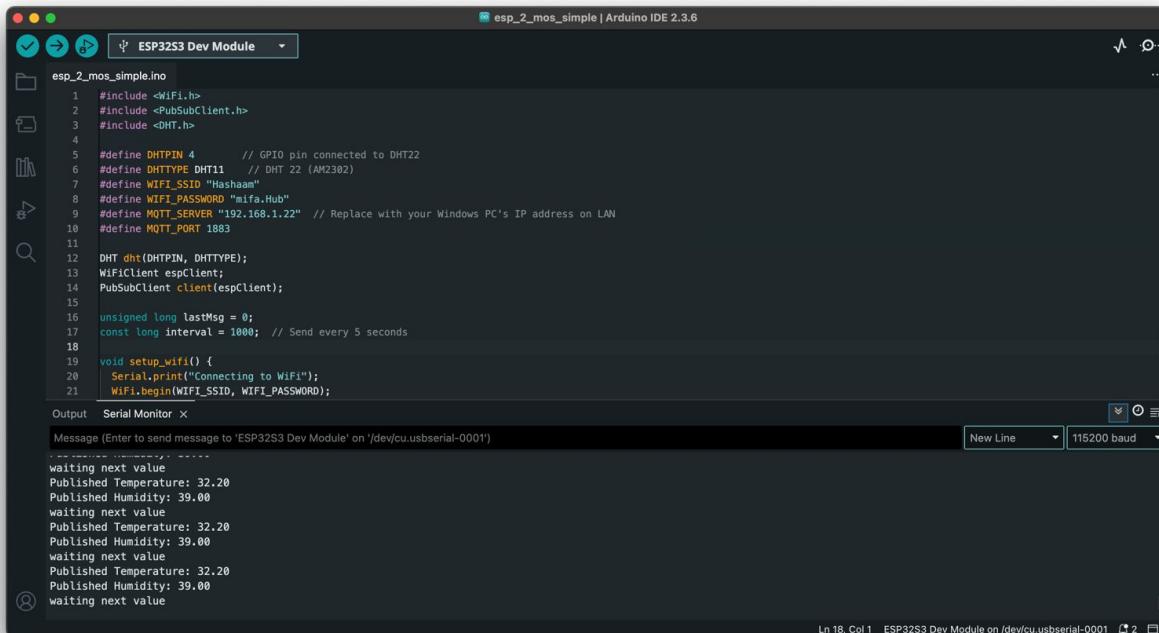
Reg-No:22-NTU-CS-1361

Date: 18-May-2025

Course: Internet Of Things

Task 1: Publish DHT Data via MQTT

- Used ESP32 + DHT11 sensor to publish temp & humidity to Mosquitto broker.
- Data sent every second to topics:
 - esp32/dht/temp
 - esp32/dht/hum



The screenshot shows the Arduino IDE interface with the following details:

- Sketch:** esp_2_mos_simple.ino
- Board:** ESP32S3 Dev Module
- Serial Monitor Output:**

```
waiting next value
Published Temperature: 32.20
Published Humidity: 39.00
waiting next value
Published Temperature: 32.20
Published Humidity: 39.00
waiting next value
Published Temperature: 32.20
Published Humidity: 39.00
waiting next value
```
- Bottom Status Bar:** Ln 18, Col 1 ESP32S3 Dev Module on /dev/cu.usbserial-0001

Task 2: Store MQTT Data in InfluxDB

- Ran 1-dht_data_only.py to subscribe & store MQTT values.
- Connected to InfluxDB bucket (iot_data).
- Verified data using InfluxDB queries.

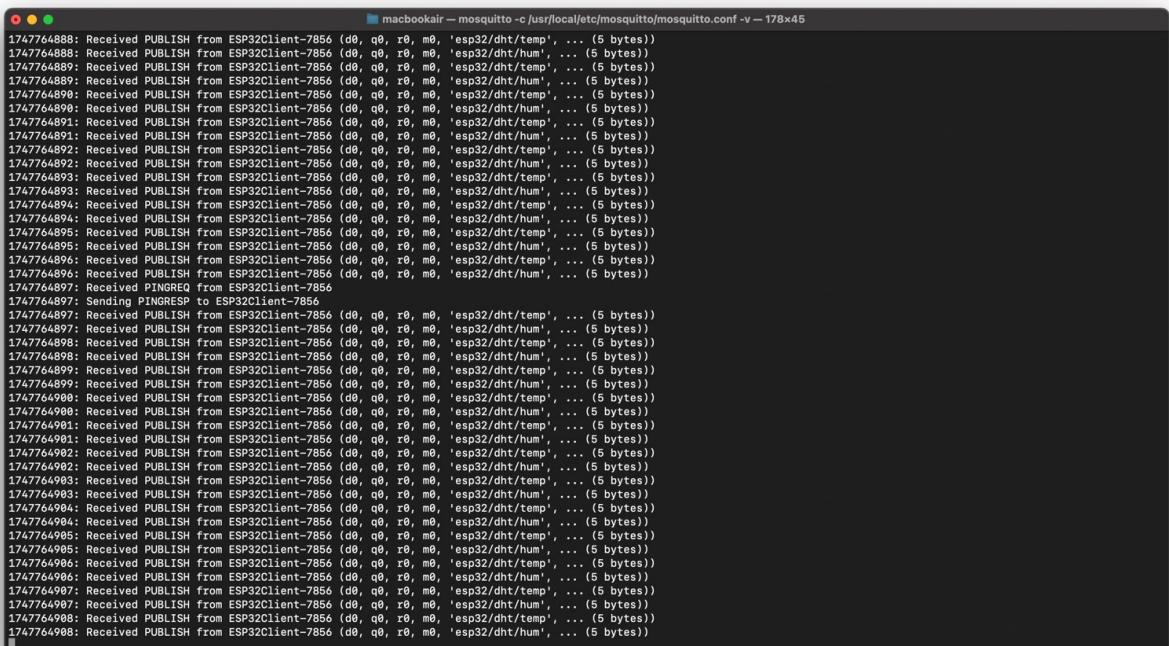
MQTT Received:

- Temp: e.g., 28.5°C
- Humidity: e.g., 36.6%

InfluxDB: Service ON Query

Sample:

```
sql
CopyEdit
from(bucket: "iot_data")
|> range(start: -1h)
|> filter(fn: (r) => r._measurement == "dht_data")
```



```
macbookair — mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf -v - 178x45
1747764888: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764888: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764889: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764889: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764890: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764890: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764891: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764891: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764892: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764892: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764893: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764893: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764894: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764894: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764895: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764895: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764896: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764896: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764897: Received PINGREQ from ESP32Client-7856
1747764897: Sending PINGRESP to ESP32Client-7856
1747764897: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764897: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764898: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764898: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764899: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764899: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764900: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764900: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764901: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764901: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764902: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764902: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764903: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764903: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764904: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764904: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764905: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764905: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764906: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764906: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764907: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764907: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
1747764908: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/temp', ... (5 bytes))
1747764908: Received PUBLISH from ESP32Client-7856 (d0, q0, r0, m0, 'esp32/dht/hum', ... (5 bytes))
```

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a folder named "PYTHON-SCRIPTS" containing four files: "0-train_model.py", "1-dht_data_only.py", "2-train_model_with_noise.py", and "3-classify_2_influx_.py".
- Code Editor:** The active file is "1-dht_data_only.py". The code is as follows:

```
# only dht data store to influxdb from esp32 via mosquitto mqtt broker
import paho.mqtt.client as mqtt
from influxdb_client import InfluxDBClient, Point
import time

# InfluxDB setup
INFLUXDB_URL = "http://localhost:8086" # InfluxDB server URL
INFLUXDB_TOKEN =
    "weeo9RKHYueNS0hzXS7Del96JBvvi9WqpMYjVifLTzeJc_HU5nZ-eN4_ToGvAz_TUJBvqIXPZ3W2onSMju0vA==" #
Replace with your InfluxDB token
INFLUXDB_ORG = "NTU"      # Replace with your InfluxDB organization name
INFLUXDB_BUCKET = "Lab13" # InfluxDB bucket name

# MQTT setup
MQTT_BROKER = "192.168.186.177" # ESP32's MQTT broker address
MQTT_PORT = 1883                 # MQTT port
MQTT_TOPIC_TEMP = "esp32/dht/temp"
```

- Terminal:** Shows the command "python-scripts %". Below it is a stack trace for a TimeoutError:

```
File "/opt/anaconda3/envs/myenv/lib/python3.12/site-packages/paho/mqtt/client.py", line 1598, in reconnect
    self._sock = self._create_socket()
File "/opt/anaconda3/envs/myenv/lib/python3.12/site-packages/paho/mqtt/client.py", line 4689, in _create_socket
    sock = self._create_socket_connection()
File "/opt/anaconda3/envs/myenv/lib/python3.12/site-packages/paho/mqtt/client.py", line 4640, in _create_socket_connection
    return socket.create_connection(addr, timeout=self._connect_timeout, source_address=source)
File "/opt/anaconda3/envs/myenv/lib/python3.12/socket.py", line 865, in create_connection
    raise exceptions[0]
File "/opt/anaconda3/envs/myenv/lib/python3.12/socket.py", line 850, in create_connection
    sock.connect(sa)
TimeoutError: timed out
```

- Status Bar:** Shows the current file is "1-dht_data_only.py", line 8, column 32. It also indicates the environment is "Python 3.12.8 ('myenv': conda)".

Data Explorer | NTU | InfluxDB | MQTT localhost url? - Google | localhost

localhost:8086/orgs/dd168287844a1a0d/data-explorer

influxdb hashamulhaq NTU

Data Explorer

Table CUSTOMIZE Local SAVE AS

Filter tables... _start _stop _time _value _field _measurement device

_field=humidity _measurement=dht_data device=esp32
_field=temperature _measurement=dht_data device=esp32
_field=temperature _measurement=dht_data device=esp32

Query 1 (0.02s) + View Raw Data CSV Past 3h SCRIPT EDITOR SUBMIT

FROM Filter Filter Filter WINDOW PERIOD

Search buckets _measurement _field device CUSTOM AUTO

IOT Lab13 _monitoring _field humidity Search _field tag values Search device tag values auto (1m)
_tasks + Create Bucket dht_data temperature esp32

Fill missing values AGGREGATE FUNCTION

CUSTOM AUTO

Help & Support

Data Explorer | NTU | InfluxDB | MQTT localhost url? - Google | localhost

localhost:8086/orgs/dd168287844a1a0d/data-explorer

influxdb hashamulhaq NTU

Data Explorer

Table CUSTOMIZE Local SAVE AS

Filter tables... _start _stop _time _value _field _measurement device

_field=humidity _measurement=dht_data device=esp32
_field=temperature _measurement=dht_data device=esp32
_field=temperature _measurement=dht_data device=esp32

Query 1 (0.02s) + View Raw Data CSV Past 3h SCRIPT EDITOR SUBMIT

FROM Filter Filter Filter WINDOW PERIOD

Search buckets _measurement _field device CUSTOM AUTO

IOT Lab13 _monitoring _field humidity Search _field tag values Search device tag values auto (1m)
_tasks + Create Bucket dht_data temperature esp32

Fill missing values AGGREGATE FUNCTION

CUSTOM AUTO

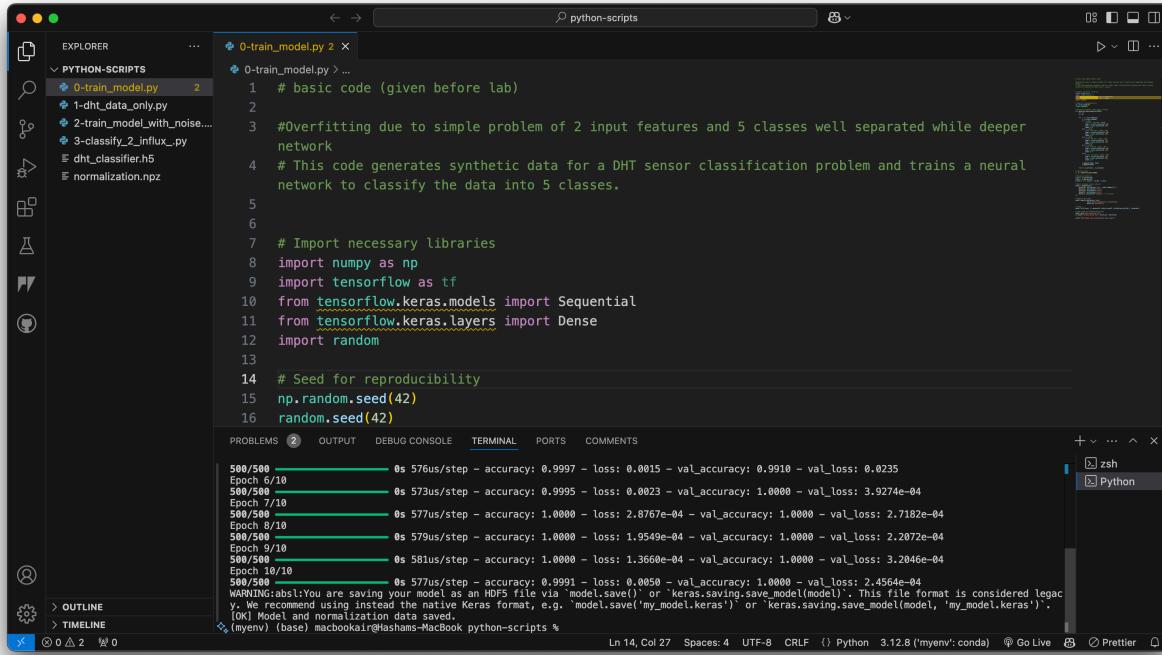
Help & Support

Task 3: Train ML Model

- Ran 2-train_model_with_noise.py to train classifier.
- Output included a confusion matrix & classification report.

Accuracy: ~93%

Labels: Low, Normal, High

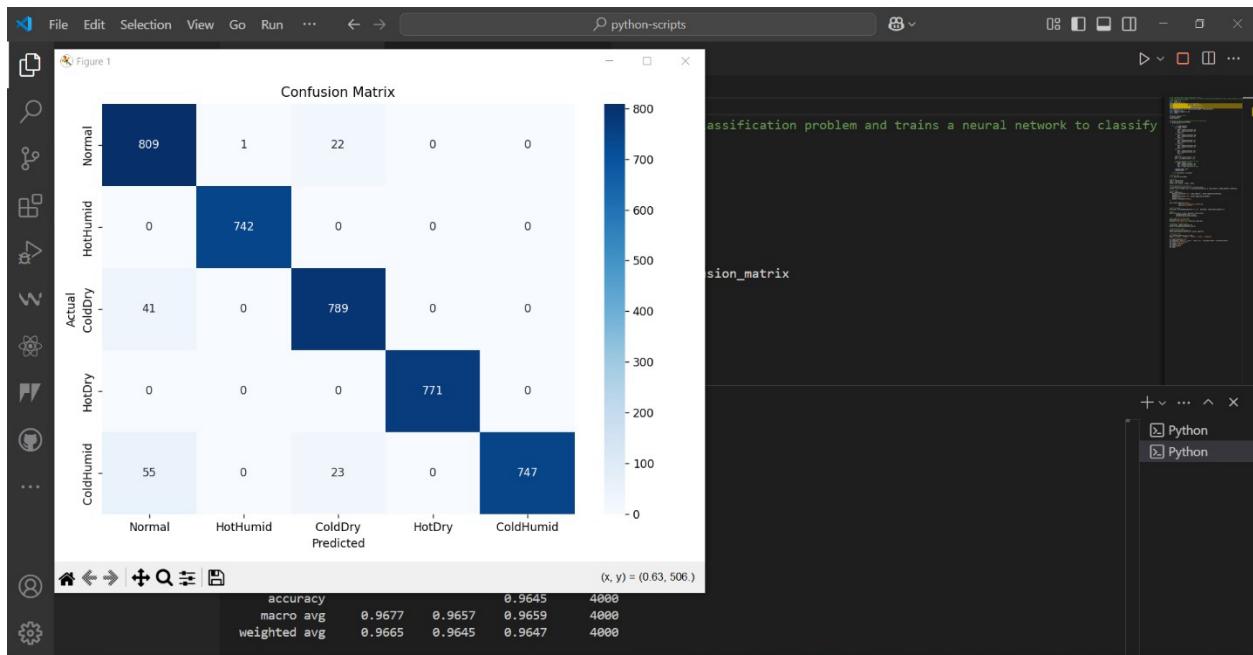


```
# basic code (given before lab)
#
# Overfitting due to simple problem of 2 input features and 5 classes well separated while deeper
# network
# This code generates synthetic data for a DHT sensor classification problem and trains a neural
# network to classify the data into 5 classes.

# Import necessary libraries
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import random

# Seed for reproducibility
np.random.seed(42)
random.seed(42)

500/500    0s 576us/step - accuracy: 0.9997 - loss: 0.0015 - val_accuracy: 0.9910 - val_loss: 0.0235
500/500    0s 573us/step - accuracy: 0.9995 - loss: 0.0023 - val_accuracy: 1.0000 - val_loss: 3.9274e-04
500/500    0s 577us/step - accuracy: 1.0000 - loss: 2.8767e-04 - val_accuracy: 1.0000 - val_loss: 2.7182e-04
500/500    0s 579us/step - accuracy: 1.0000 - loss: 1.9549e-04 - val_accuracy: 1.0000 - val_loss: 2.2072e-04
500/500    0s 581us/step - accuracy: 1.0000 - loss: 1.3660e-04 - val_accuracy: 1.0000 - val_loss: 3.2046e-04
500/500    0s 577us/step - accuracy: 0.9991 - loss: 0.0059 - val_accuracy: 1.0000 - val_loss: 2.4564e-04
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
[OK] Model and normalization data saved.
```



Task 4: Classify & Store in InfluxDB

- Ran 3-classify_2_influx.py.
- Subscribed to data, used trained model to classify conditions.
- Saved predictions to InfluxDB.

Query Sample:

```
sql
CopyEdit
from(bucket: "iot_data")
|> range(start: -30m)
|> filter(fn: (r) => r._field == "classification")
```

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER**: Shows files in the workspace:
 - OPEN EDITORS: 1-dht_data_only.py, 3-classify_2_influx.py
 - PYTHON-SCRIPTS: 0-train_model.py, 1-dht_data_only.py, 2-train_model_wit..., 3-classify_2_influx..., dht_classifier.h5, normalization.npz
- TERMINAL**: Displays the output of the Python script execution, showing logs for writing data to InfluxDB and saving it to disk.
- PROBLEMS**: No problems listed.
- OUTPUT**: No output listed.
- DEBUG CONSOLE**: No debug console output listed.
- PORTS**: No ports listed.

```
3-classify_2_influx.py
...
9     # InfluxDB setup
10    INFLUXDB_URL = "http://localhost:8086" # InfluxDB server URL
11    INFLUXDB_TOKEN = "O-1iMz5nWitp26k2hpTjMCvtfj4uyt-RYC2tok_2B05LF8PAQwQ0ayf4e85eSUWD2Ubtjold6KOfPuRgSyRKA==" # Replace
12    INFLUXDB_ORG = "IOT-LAB13"           # Replace with your InfluxDB organization name
13    INFLUXDB_BUCKET = "LAB13"          # InfluxDB bucket name
14
15    # MQTT setup
16    MQTT_BROKER = "192.168.18.31" # ESP32's MQTT broker address
17    MQTT_PORT = 1883                # MQTT port
18    MQTT_TOPIC_TEMP = "esp32/dht/temp"
19    MQTT_TOPIC_HUM = "esp32/dht/hum"
20
21    # Class names
now = datetime.utcnow().replace(tzinfo=timezone.utc)
Writing to InfluxDB: dht_data,device=esp32 class_label="Hot and Dry",humidity=38.6,temperature=28.7 1747558938034538000
Data saved: Temp=28.70, Hum=38.60, Class=Hot and Dry
Received Temperature: 28.70°C
Received Humidity: 38.70%
Predicted Class: Hot and Dry
Writing to InfluxDB: dht_data,device=esp32 class_label="Hot and Dry",humidity=38.7,temperature=28.7 1747558942785248000
Data saved: Temp=28.70, Hum=38.70, Class=Hot and Dry
Received Temperature: 28.70°C
Received Humidity: 38.70%
Predicted Class: Hot and Dry
Writing to InfluxDB: dht_data,device=esp32 class_label="Hot and Dry",humidity=38.7,temperature=28.7 1747558947915002000
Data saved: Temp=28.70, Hum=38.70, Class=Hot and Dry
```

Challenges Encountered

Issue	Resolution
WiFi connection failed	Double-checked SSID and restarted ESP32
MQTT data not arriving	Opened port 1883 in the system firewall
No data in InfluxDB	Corrected bucket and measurement names
Model performance poor	Minimized noise and retrained the model

Conclusion

Lab 13 was completed successfully. I managed to stream live sensor data from the ESP32 through MQTT into InfluxDB and implemented a machine learning model to classify it. All components were tested and confirmed to work correctly. This lab provided valuable hands-on experience with integrating IoT and AI systems in a practical setting.