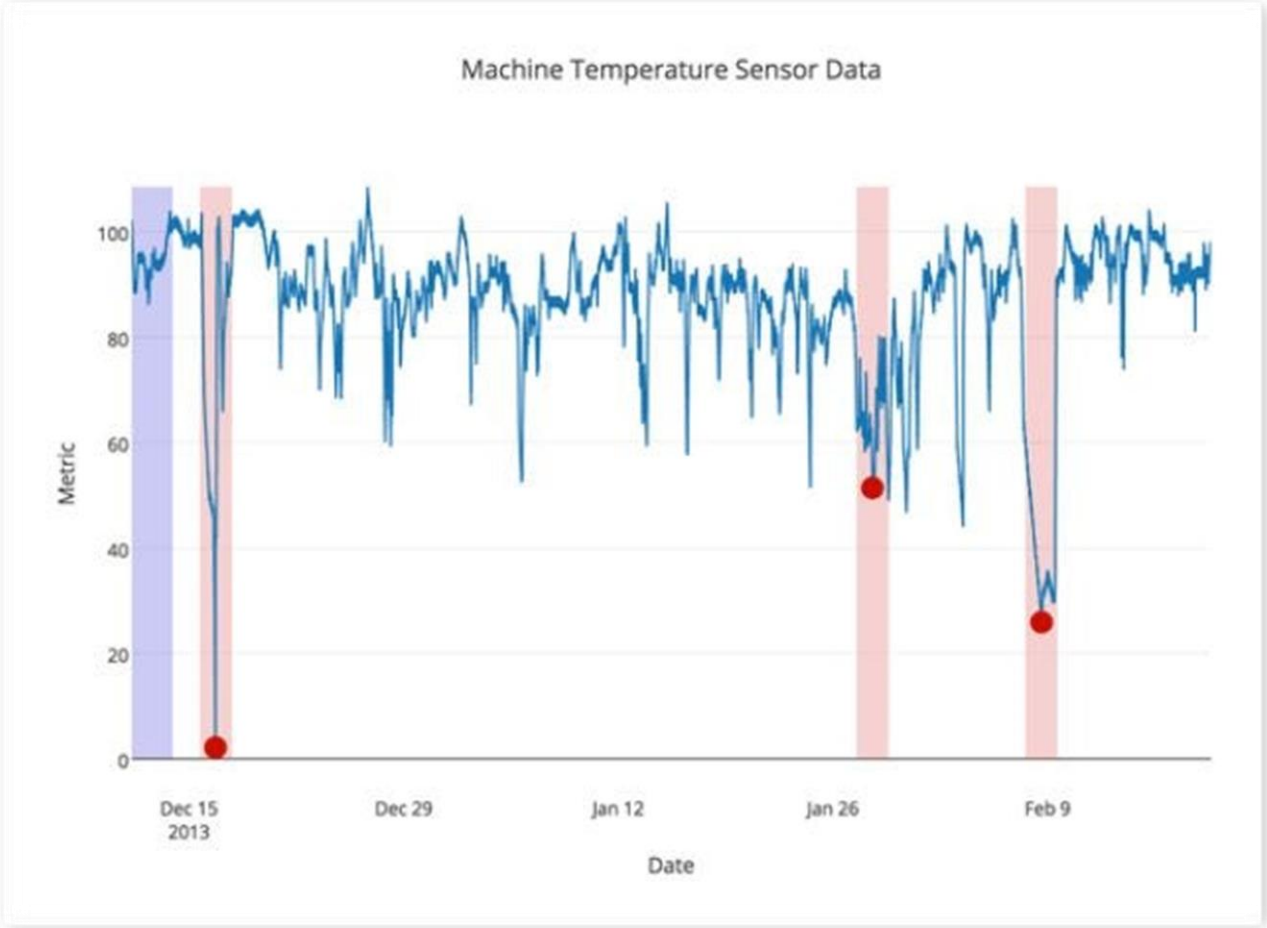
IoT Capstone Project Report

Temperature IoT Data Anomaly Detection

(Using ML methods)



Semester: Fall 2023

Armin Attarzadeh & Arshia Goshtasbi

Instructor: Dr. Reza Vahidnia

Iran University of Science &Technology

Applications of Anomaly Detection in IoT

Enhancing IoT Security and Efficiency

Industrial Automation:

Early identification of equipment malfunctions or overheating in industrial settings.

Energy Management:

Optimizing energy consumption by identifying anomalies in temperature patterns in smart buildings and facilities.

Healthcare:

Ensuring proper storage conditions for pharmaceuticals and medical supplies.

Agriculture & Food Safety:

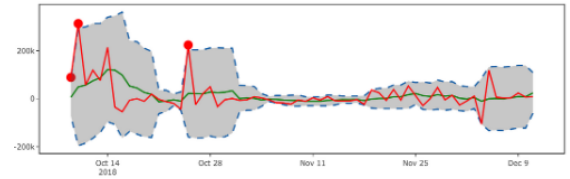
Early detection of climate-related anomalies affecting crop growth.

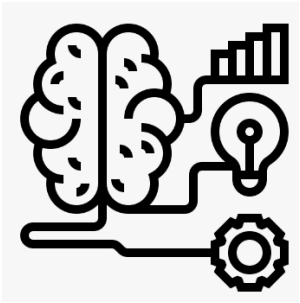
Ensuring the correct storage temperature for perishable goods during transportation and storage.

System Diagram

Overview of IoT System Components

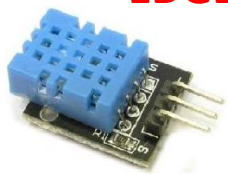
Real Time
ANAYLSIS

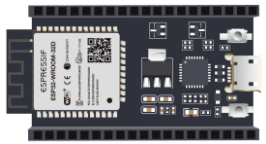




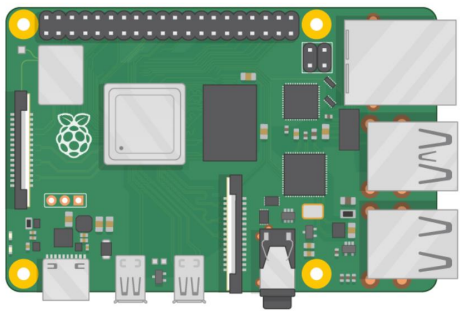
ML detection

EDGE





Cloud DB



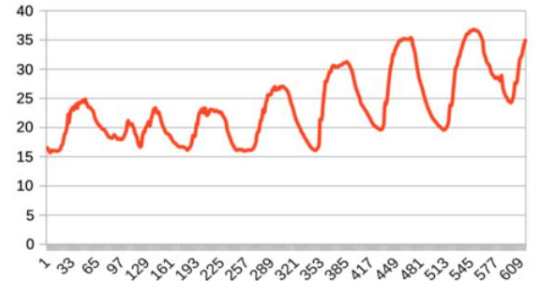
GETEAWAY RPI 4

MQTT + Encryption

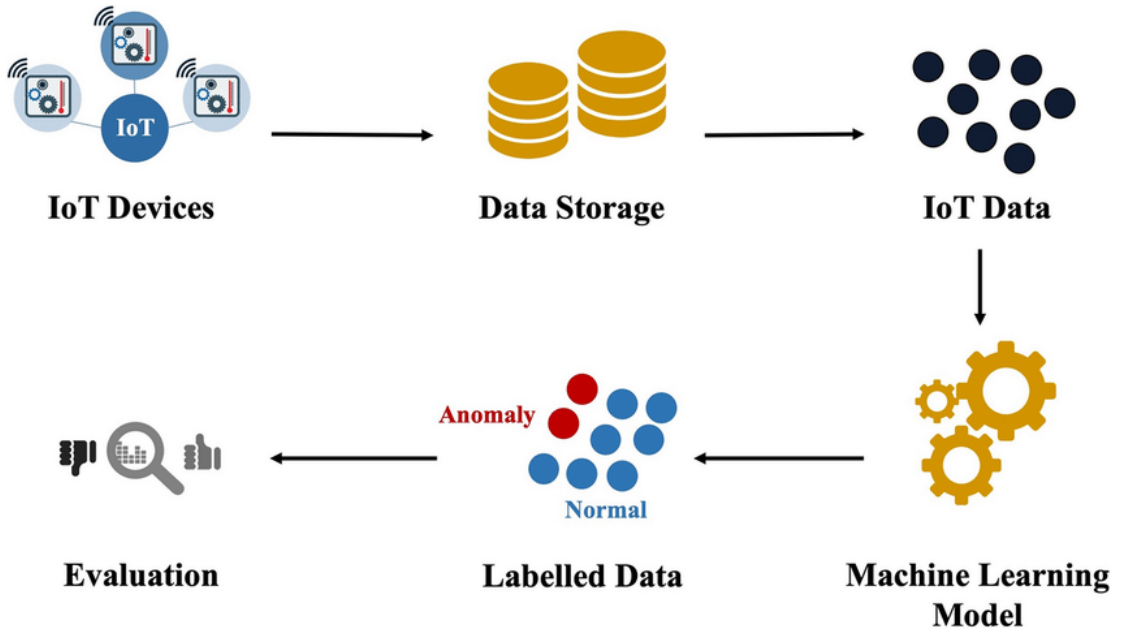
SECURE Transmission

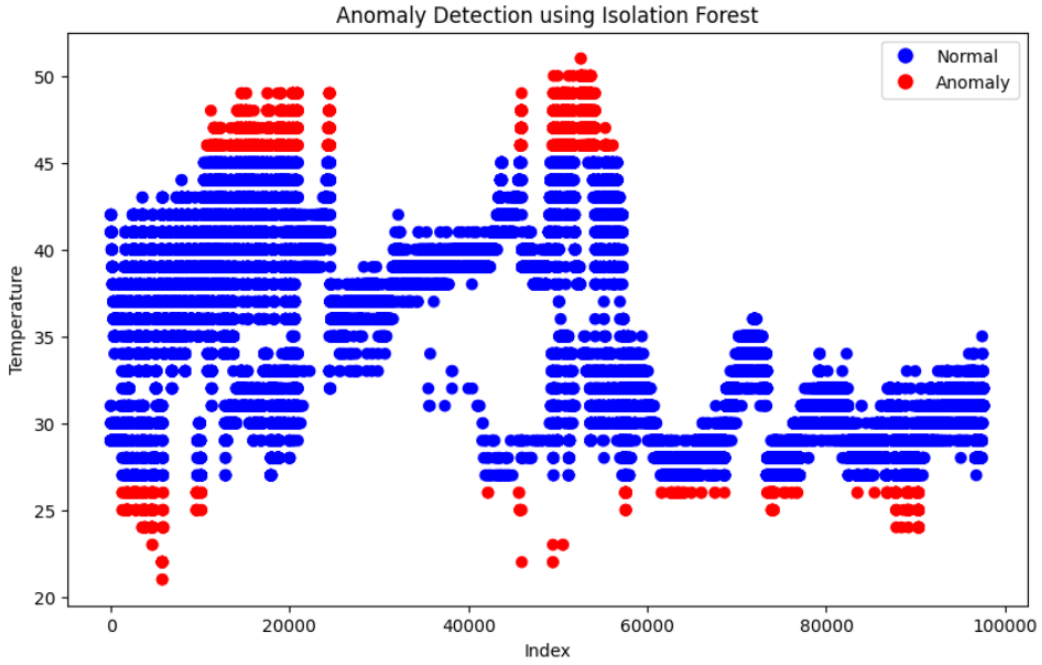
Node-RED

RT DASHBOARD



Review:
Steps for Implementing Anomaly Detection in IoT
From Data Collection to Actionable Insights



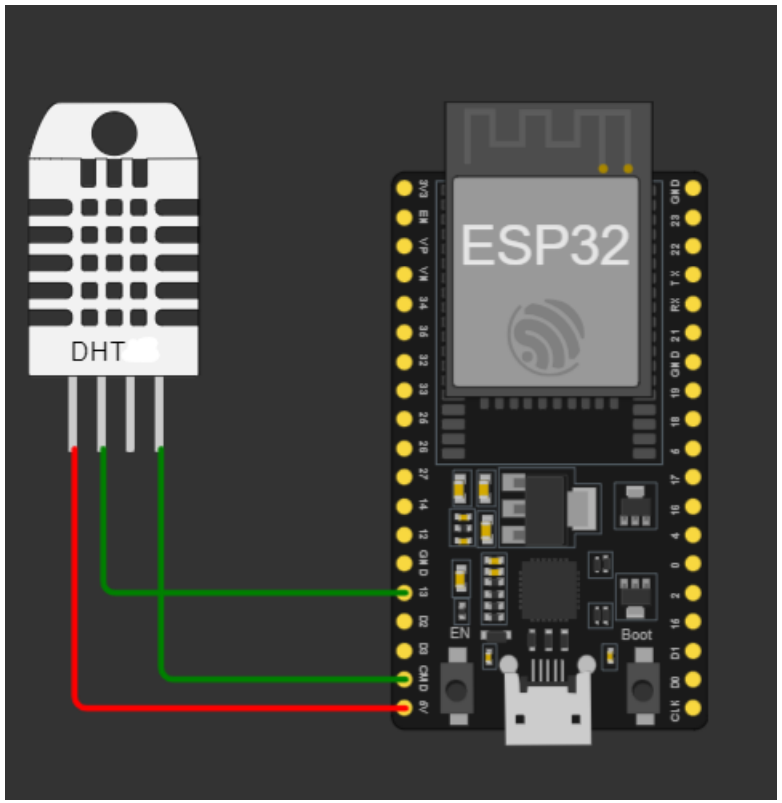


e.g. Isolation Forest

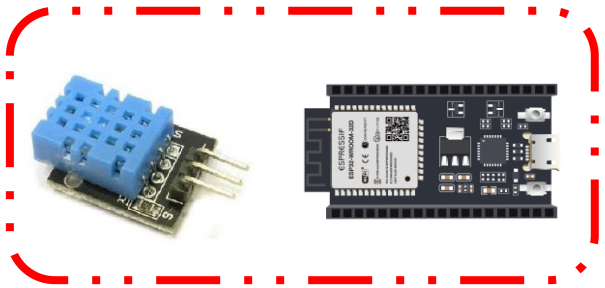
Part 1: (Design / Hardware / Sensors)

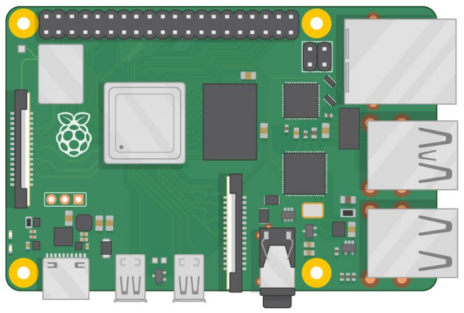
Connecting the DHT11 Sensor to ESP32 and Receiving the temperature:
After Installation of the sensor with the following format, the script to show the temp is used.

The Schematics:



Part 2: (Communications)



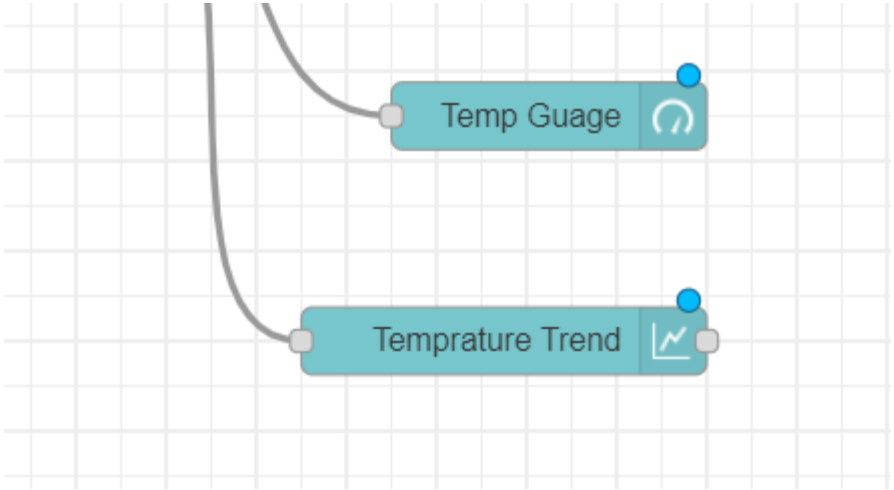


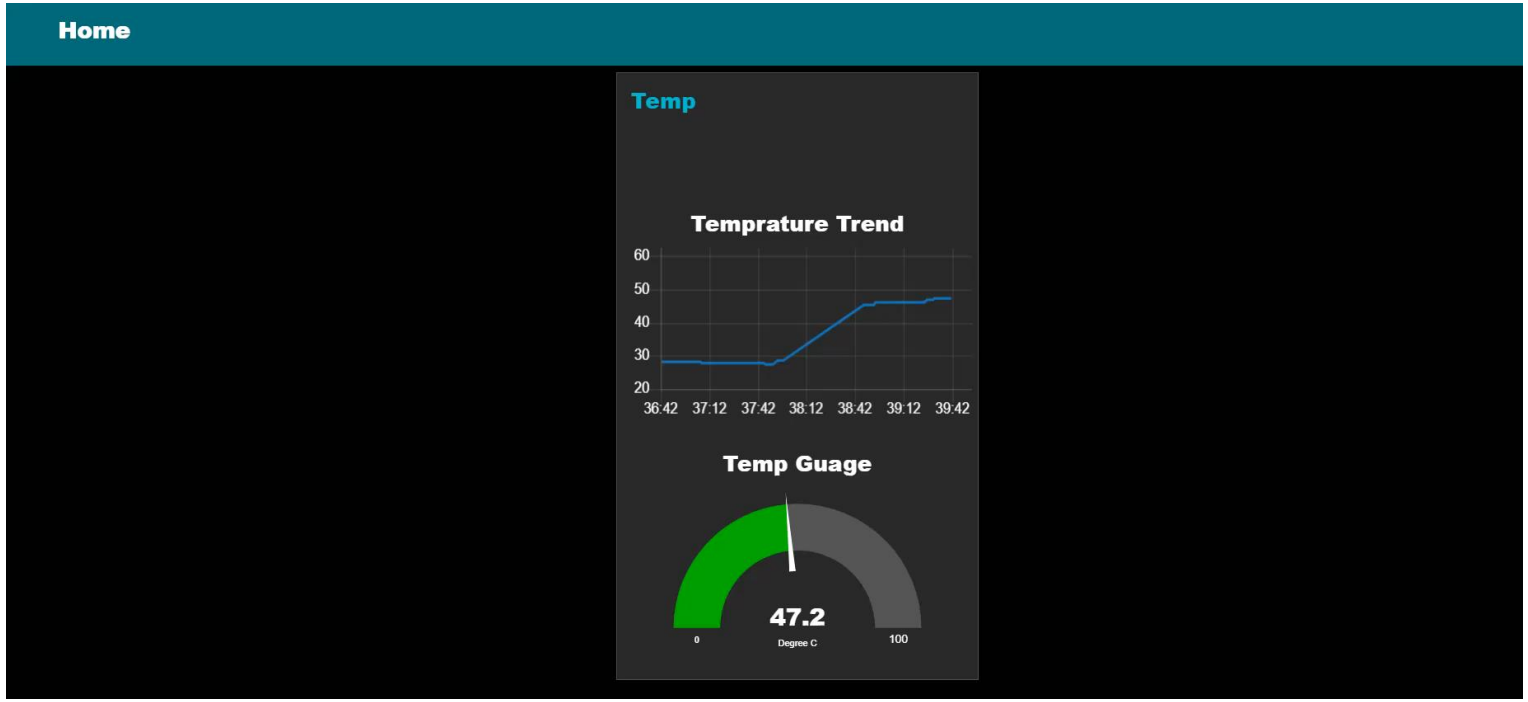
Sending the received data via Bluetooth to Raspberry pi. (C Code)
Receiving the data via Bluetooth in Raspberry pi. (Python Code)
The bluepy library is used
In the part the root access is required.
Receiving the data via Bluetooth in Raspberry pi,
Encrypting the data using RC4 and sending it to cloud using MQTT. (Python Code)

SECURE Transmission

Part 3: Dashboard & Monitoring

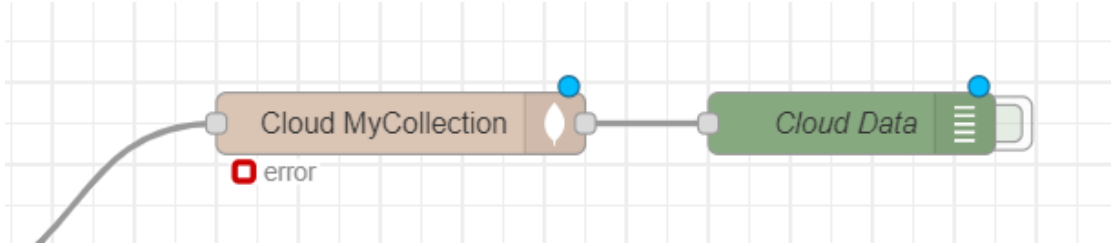
Node-RED





node-red-dashboard 3.6.2

Part 4: Cloud Database



Cloud Hosted mongoDB® for FREE

node-red-contrib-mongodb-aleph 0.3.0

Filter | Type a query: { field: 'value' } | Reset | Apply | Options

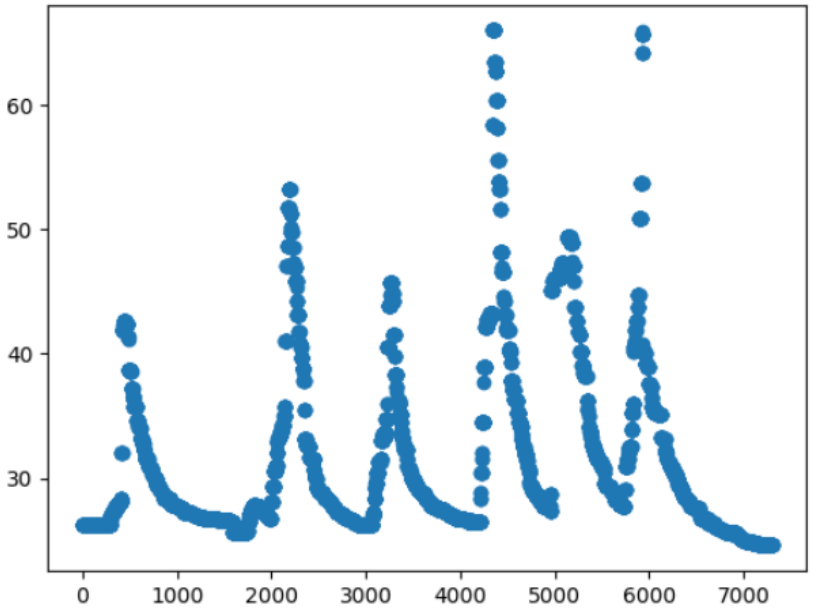
QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('65b3d3ad01da125fe019f048')
payload: 26.3
_msgid: "c8ea51d0d5658175"
```

```
_id: ObjectId('65b3d3ae01da125fe019f049')
payload: 26.3
```

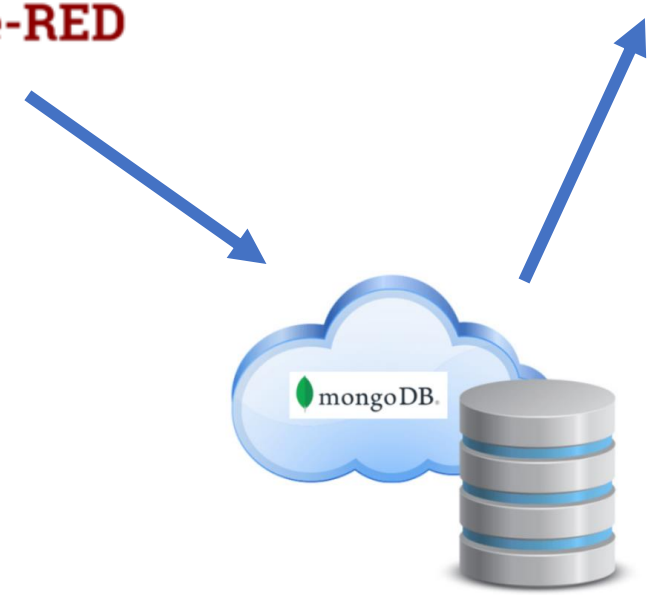
Part 4: Cloud Database

Cloud Hosted mongoDB® for FREE



Node-RED

python™

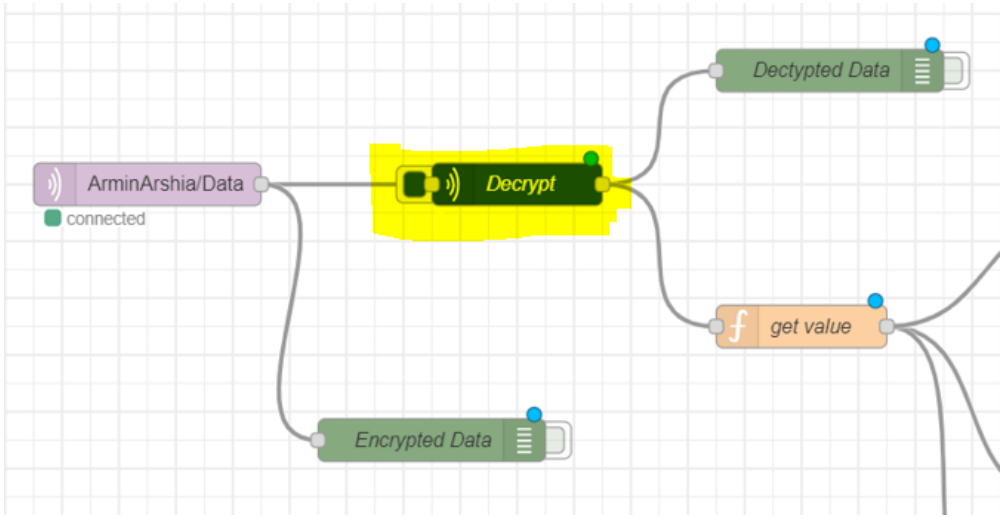


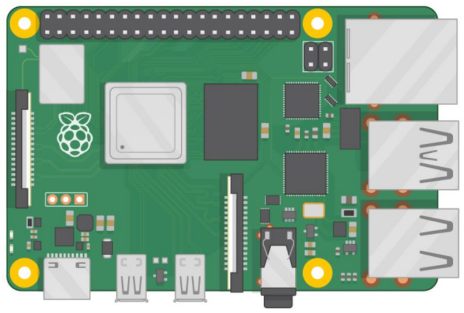

```
# Retrieve temperature data from MongoDB
```

Part 5: Encryption & Decryption

We use RC4 algorithm for Encryption.
Its implementation is easy but useful for security

(Send) Encyption.py --> running on RPI
Data has sent to mqtt borker in secure way
(Receive) Dectyption.py --> running Node-red

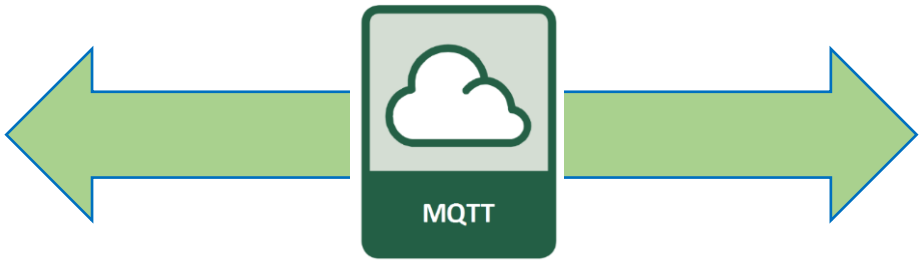




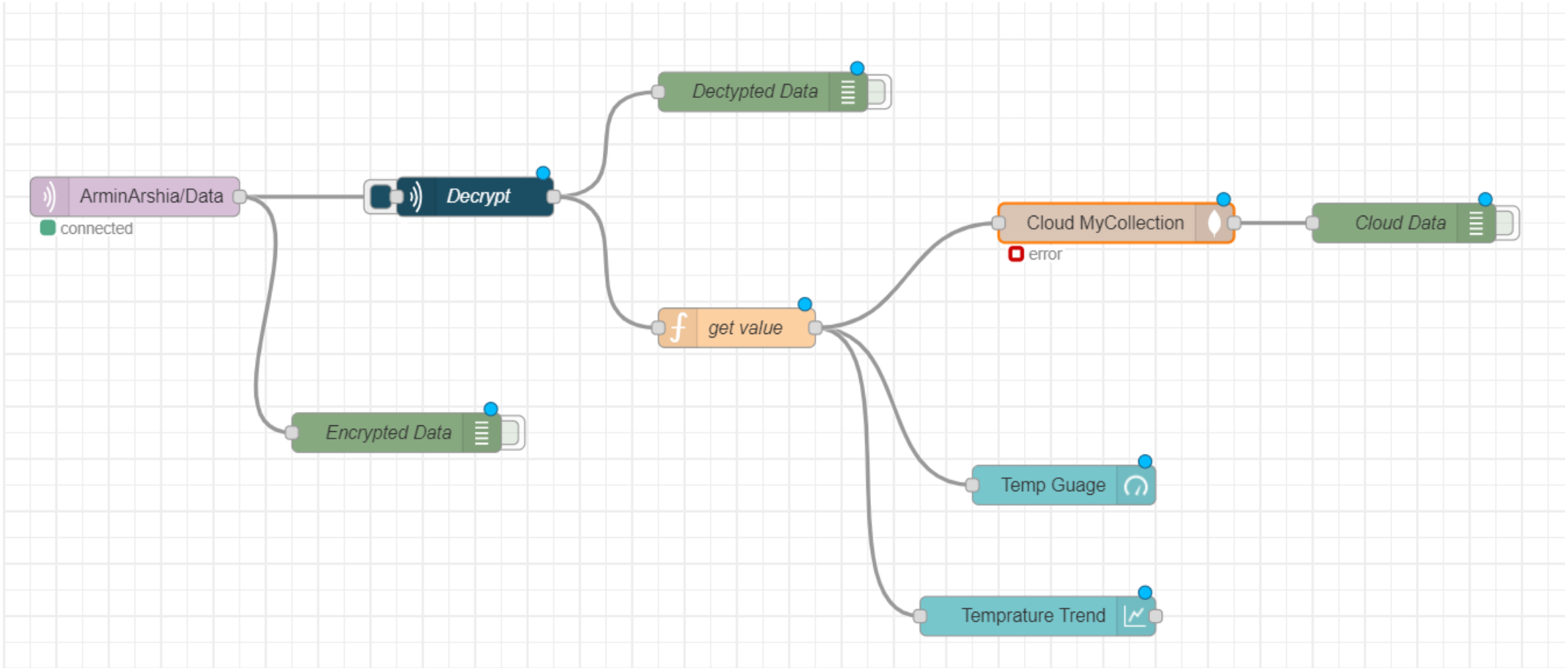
/in: {'temp':35}
/key: Armin

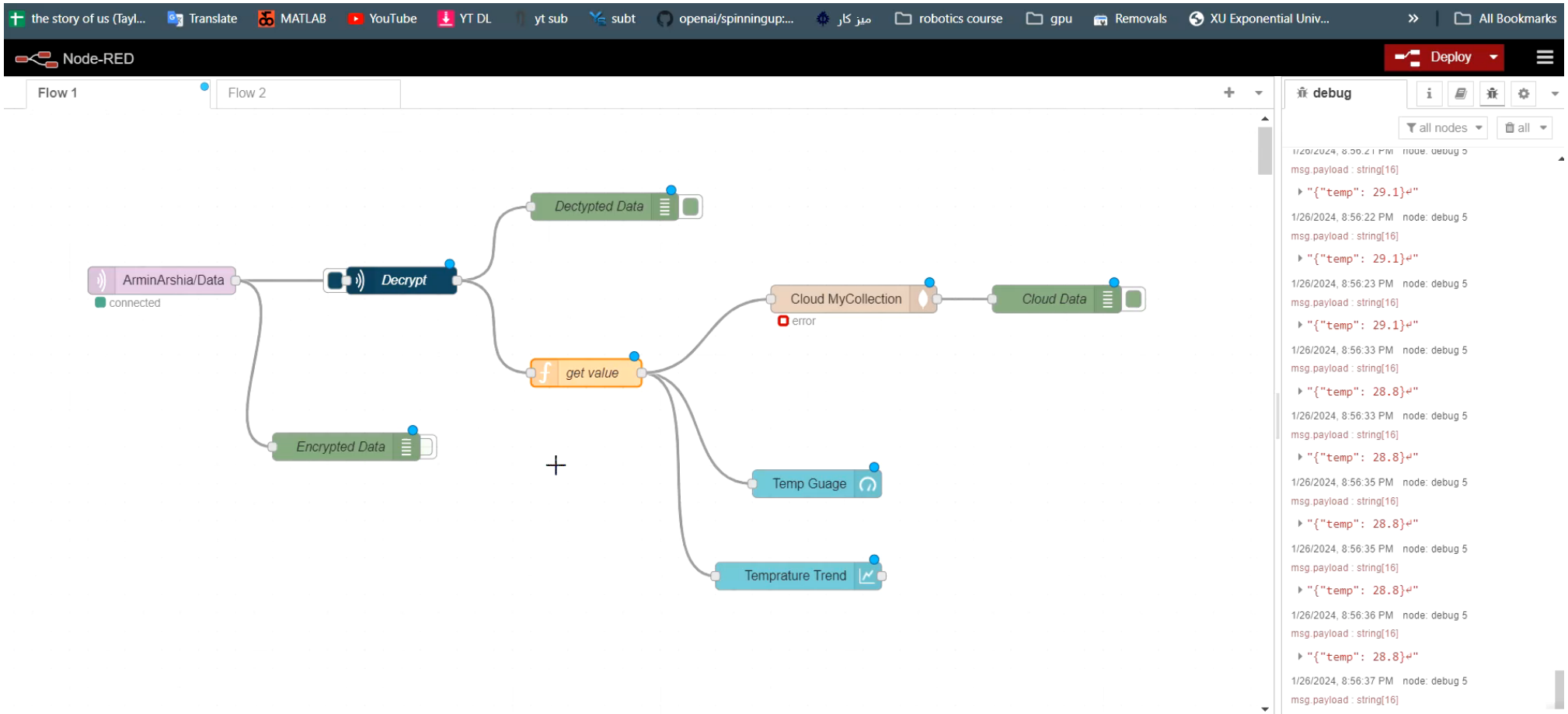
RC4 Encryption

/sent: 4daf9b6d4ba702afcc2497



/out {'temp':35}
/key: Armin

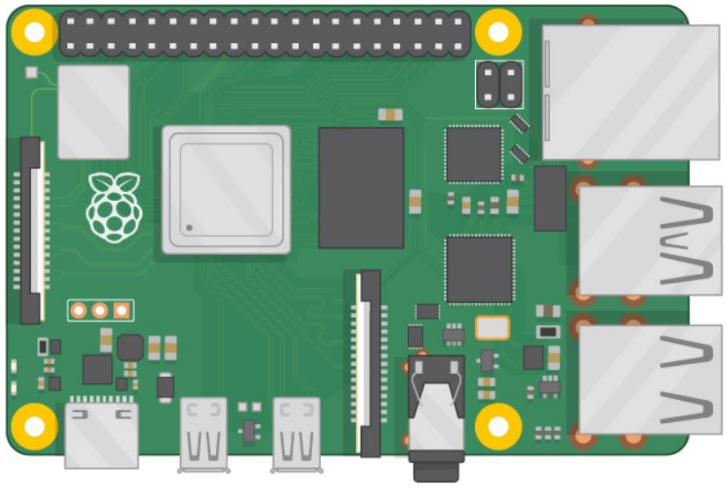

ArminArshia/Data
connected
Decrypt
Dectypted Data
Encrypted Data
get value
Cloud MyCollection
error
Cloud Data
Temp Guage
Temprature Trend


the story of us (Tayl...
Translate
MATLAB
YouTube
YT DL
yt sub
subt
openai/spinningup:...
robotics course
gpu
Removals
XU Exponential Univ...
All Bookmarks
Node-RED
Deploy
Flow 1
Flow 2
debug
all nodes
all
ArminArshia/Data
connected
Decrypt
Dectypted Data
Encrypted Data
get value
Cloud MyCollection
error
Cloud Data
Temp Guage
Temprature Trend
msg.payload : string[16]
"{"temp": 29.1}↵"
1/26/2024, 8:56:22 PM node: debug 5
1/26/2024, 8:56:23 PM node: debug 5
1/26/2024, 8:56:33 PM node: debug 5
"{"temp": 28.8}↵"
1/26/2024, 8:56:35 PM node: debug 5
1/26/2024, 8:56:36 PM node: debug 5
1/26/2024, 8:56:37 PM node: debug 5

Part 6: (Gateway)

Raspberry pi 4 is used as a gateway.
Tasks:

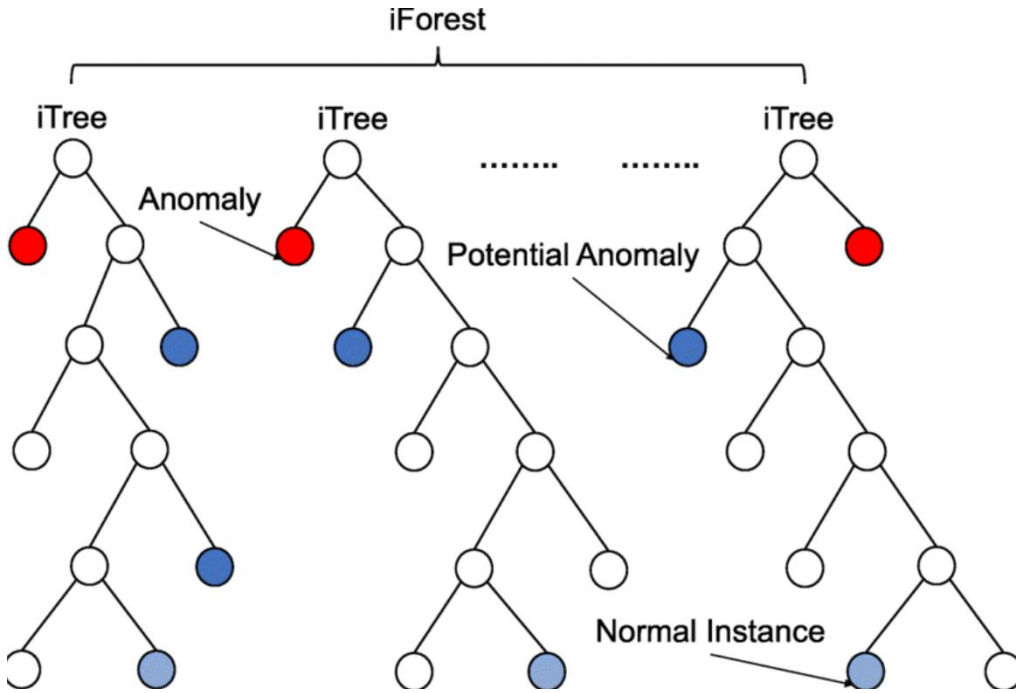
- Collecting the data from ESP32 via Bluetooth
- Encrypting the data using RC4
- Sending the encrypted data to cloud Via MQTT

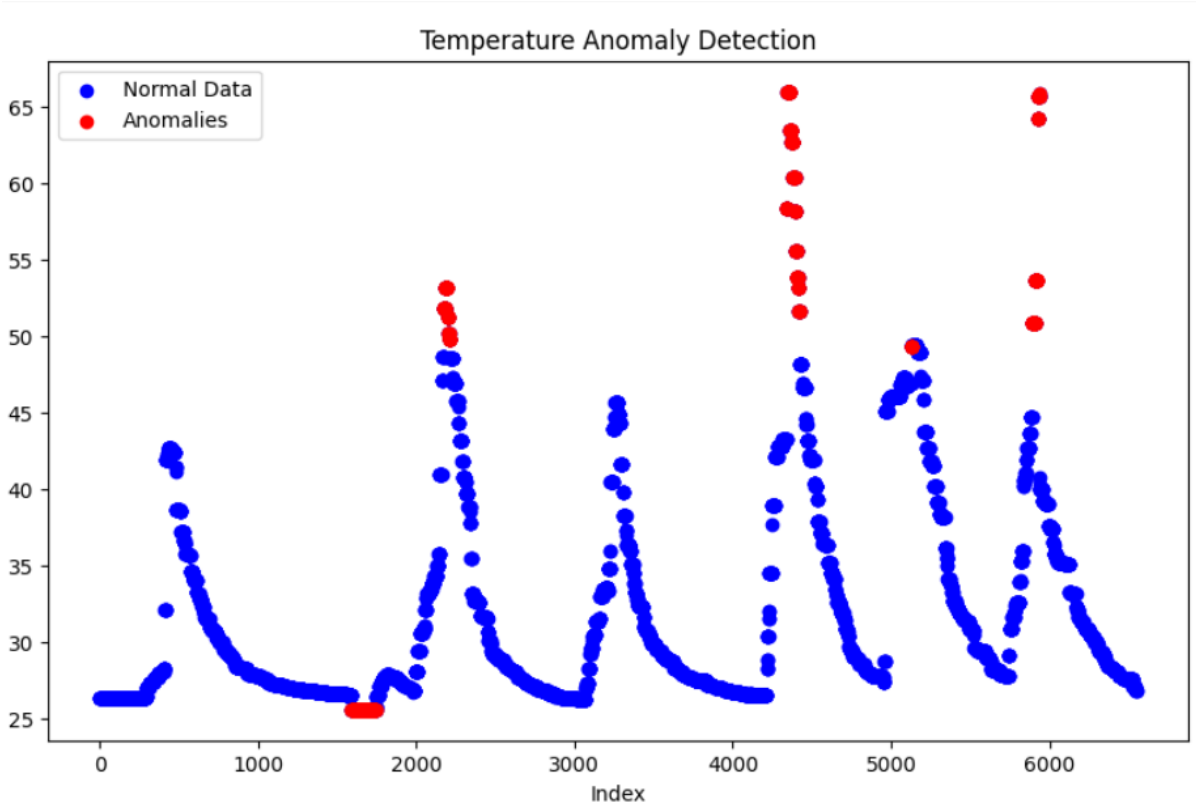


MQTT

Part 7: ML Anomaly Detection

Unsupervised ML Algorithm





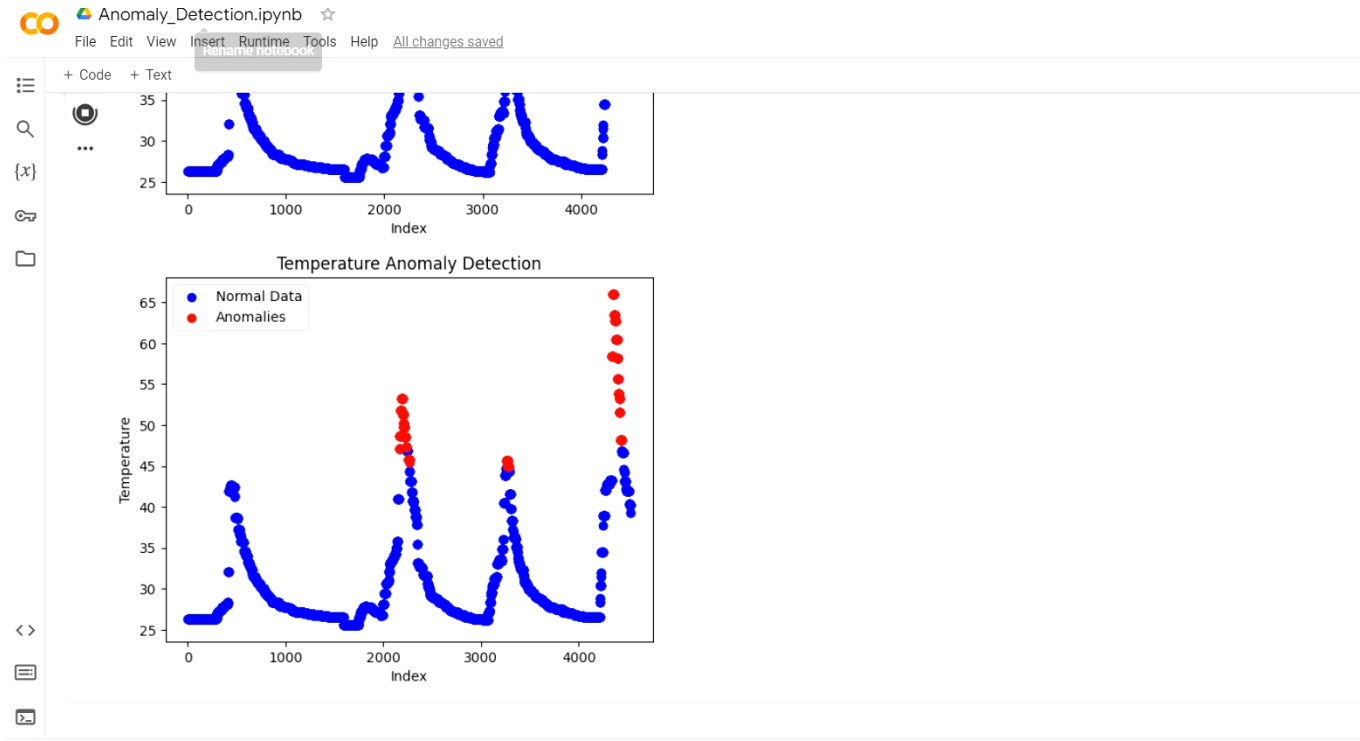
Part 7: ML Anomaly Detection

Unsupervised ML Algorithm

```
while True:
    # Read temperature data
    temperature_data = read_temperature_data()

    # Perform anomaly detection and visualization
    perform_anomaly_detection(temperature_data)

    # Wait for 10 seconds
    time.sleep(10)
```



Thank you !