Capability Data Sheet

Anomaly Detection using Raspberry Pi and the HSU Sensor Kit

Overview:

This capability focuses on detecting anomalies in real-time using environmental sensor data collected from temperature, humidity, and motion sensors connected to a Raspberry Pi. By utilizing machine learning, specifically the Isolation Forest model, the system identifies when sensor readings deviate significantly from normal patterns, signaling potential anomalies.

The system operates in real-time, continuously monitoring and detecting anomalies in the sensor data. The application includes:

- Real-time logging and analysis of sensor data
- Anomaly detection using the Isolation Forest model
- A dashboard for live visualization of the sensor data and detected anomalies

Hardware Configuration:

□ Raspberry Pi 4/3 (or compatible model)

□ DHT11 Temperature and Humidity Sensor:

- Data Pin: Connected to GPIO pin 4 (Physical pin 7)
- VCC: 3.3V (Physical pin 1) or 5V (Physical pin 2 or 4)
- GND: Ground (Physical pin 6, 9, 14, 20, 25, 30, 34, or 39)

□ PIR Motion Sensor:

- Data Pin: Connected to GPIO pin 27 (Physical pin 13)
- VCC: 5V (Physical pin 2 or 4)
- GND: Ground (Physical pin 6, 9, 14, 20, 25, 30, 34, or 39)

□ Buzzer (Alert):

- Signal Pin: Connected to GPIO pin 18 (Physical pin 12)
- GND: Ground (Physical pin 6, 9, 14, 20, 25, 30, 34, or 39)

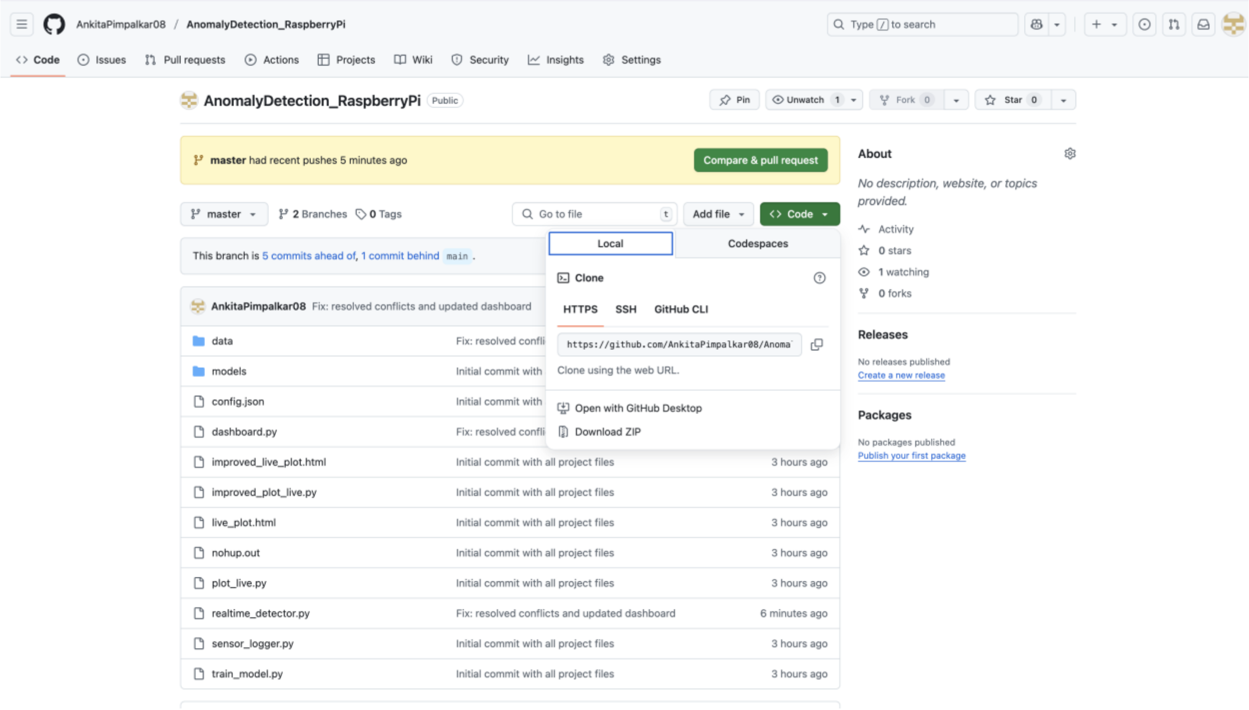
□ LED (Alert):

- Anode (+): Connected to GPIO pin 23 (Physical pin 16) through a 220-330Ω resistor
- Cathode (-): Ground (Physical pin 6, 9, 14, 20, 25, 30, 34, or 39)

 Additional accessories: Breadboard, jumper wires, resistors, and power supply for Raspberry Pi

GitHub Repository Link:

You can download the code and access the full project from the following GitHub repository:
https://github.com/AnkitaPimpalkar08/AnomalyDetection_RaspberryPi/tree/master
Click on the Download Zip



master had recent pushes 5 minutes ago

This branch is 5 commits ahead of, 1 commit behind main .

https://github.com/AnkitaPimpalkar08/Anoma

Clone using the web URL.

No description, website, or topics provided.

No releases published

No packages published

AnkitaPimpalkar08 Fix: resolved conflicts and updated dashboard

data	Fix: resolved confli	
models	Initial commit with	
config.json	Initial commit with	
dashboard.py	Fix: resolved confli	
improved_live_plot.html	Initial commit with all project files	3 hours ago
improved_plot_live.py	Initial commit with all project files	3 hours ago
live_plot.html	Initial commit with all project files	3 hours ago
nohup.out	Initial commit with all project files	3 hours ago
plot_live.py	Initial commit with all project files	3 hours ago
realtime_detector.py	Fix: resolved conflicts and updated dashboard	6 minutes ago
sensor_logger.py	Initial commit with all project files	3 hours ago
train_model.py	Initial commit with all project files	3 hours ago

Installation Steps:

1. Visit the GitHub repository link above
2. Click on the "Code" button and select "Download ZIP"
3. Extract the ZIP file to your local machine
4. Copy the extracted folder to your Raspberry Pi (see below for commands)

Required Software Libraries:

 Python 3.x (Recommended version: 3.8 or later)

 Streamlit: For building the real-time dashboard

- pip install streamlit

- Plotly: For interactive visualizations
  - pip install plotly
- pandas: For handling and processing sensor data
  - pip install pandas
- joblib: For loading the pre-trained machine learning model
  - pip install joblib
- scikit-learn: For machine learning model implementation (Isolation Forest)
  - pip install scikit-learn
- RPi.GPIO: For GPIO control on Raspberry Pi
  - pip install RPi.GPIO
- Pigpio: For enhanced GPIO access and interfacing with the DHT11 sensor
  - pip install pigpio
- Additional required packages:
  - matplotlib: For visualization
  - seaborn: For advanced visualization
  - XlsxWriter: For export functionality
  - pip install matplotlib seaborn XlsxWriter

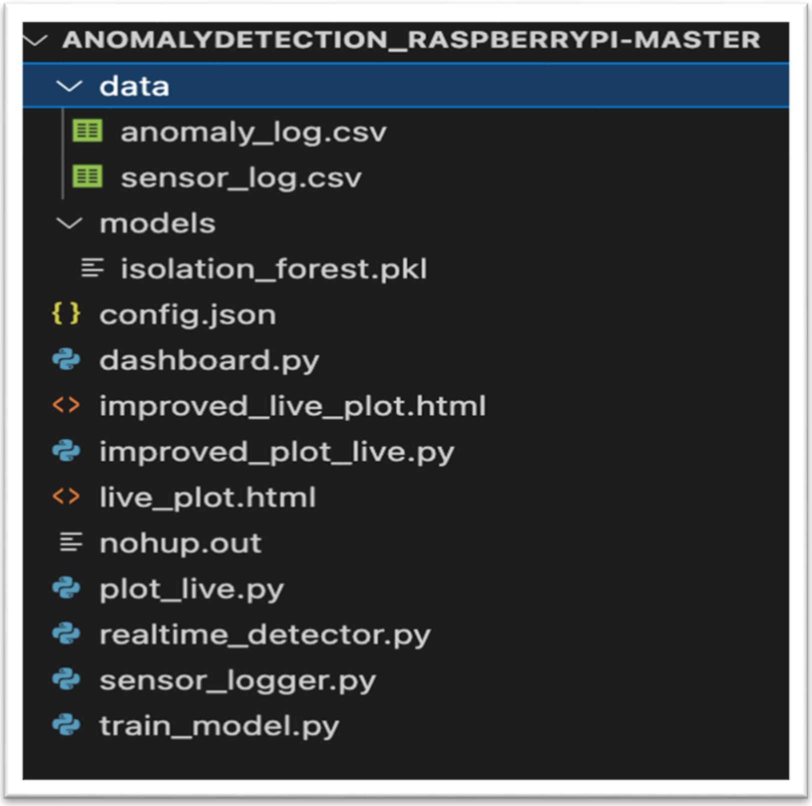
Required Data:

- **Sensor data CSV** (data/sensor_log.csv): A CSV file containing the timestamp, temperature, humidity, and motion data.
- **Anomaly log CSV** (data/anomaly_log.csv): A CSV file used to store detected anomalies along with their timestamps and sensor readings.

Key Parameters:

- **Contamination (contamination)**: This parameter in the Isolation Forest model defines the proportion of outliers in the dataset. A value of 0.1 (10%) is typically used for anomaly detection.
- **Rolling Window (rolling_window)**: Defines the size of the rolling window used for smoothing sensor data before passing it to the model for prediction(10).

Project File Structure:



```
ANOMALYDETECTION_RASPBERRYPI-MASTER
  data
    anomaly_log.csv
    sensor_log.csv
  models
    isolation_forest.pkl
  config.json
  dashboard.py
  improved_live_plot.html
  improved_plot_live.py
  live_plot.html
  nohup.out
  plot_live.py
  realtime_detector.py
  sensor_logger.py
  train_model.py
```

Python Code Files:

1. **sensor_logger.py**: Logs data from the temperature, humidity, and motion sensors.
   - Reads sensor data and writes it to sensor_log.csv in real time.
2. **train_model.py**: Trains the Isolation Forest model on historical data and saves it to models/isolation_forest.pkl.
   - The model is used for detecting anomalies.
3. **realtime_detector.py**: Continuously monitors the sensors in real time and detects anomalies.
   - Reads data, scales it, and applies the trained model to predict anomalies. Detected anomalies are logged in anomaly_log.csv.
4. **dashboard.py**: Displays a real-time dashboard with live data and anomaly detection.
   - Uses Streamlit and Plotly for visualization of temperature, humidity, motion, and anomalies.

Step-by-Step Guide to Run Project:

1. Download the Project

First, download the project from GitHub:

```
git clone https://github.com/AnkitaPimpalkar08/AnomalyDetection_RaspberryPi.git
```

Or download it as a ZIP file from:

https://github.com/AnkitaPimpalkar08/AnomalyDetection_RaspberryPi/tree/master

2. Access the Raspberry Pi Terminal

Either directly from the Raspberry Pi or SSH from another computer:

```
ssh pi@raspberrypi.local
password: raspberry123 or whatever your password maybe
```

3. Copy Project Files to Raspberry Pi

If you downloaded the project to your computer, transfer it to your Raspberry Pi:

```
scp -r /path/to/your/downloads/AnomalyDetection_RaspberryPi pi@raspberrypi.local:~/
```

Or if you downloaded it directly on the Raspberry Pi:

```
# On Raspberry Pi
cp -r ~/Downloads/AnomalyDetection_RaspberryPi ~/
```

4. Navigate to Your Project Directory

```
cd ~/AnomalyDetection_RaspberryPi
```

5. Set Up the Virtual Environment

bash

```
# First time setup only
python3 -m venv .venv
# Activate the virtual environment (do this every session)
source .venv/bin/activate
```

6. Install Required Dependencies

bash

```
# System dependencies
sudo apt-get update
sudo apt-get install pigpio
sudo systemctl enable pigpiod
sudo systemctl start pigpiod
# Python packages
pip install numpy pandas scikit-learn joblib
pip install RPi.GPIO adafruit-blinka adafruit-circuitpython-dht
pip install pigpio streamlit plotly matplotlib seaborn XlsxWriter
```

7. Collect Data

bash

```
python sensor_logger.py
```

Let this run for a long time to collect enough meaningful data. The more varied the environmental conditions you capture, the better your model will identify anomalies.

8. Train the Model

bash

```
python train_model.py
```

This processes the collected data in sensor_log.csv and creates a trained model saved to models/isolation_forest.pkl.

9. Run Real-Time Anomaly Detection

bash

```
python realtime_detector.py
```

This uses your trained model to analyze incoming sensor data in real-time and identify potential anomalies, logging them to data/anomaly_log.csv.

10. View the Dashboard

bash

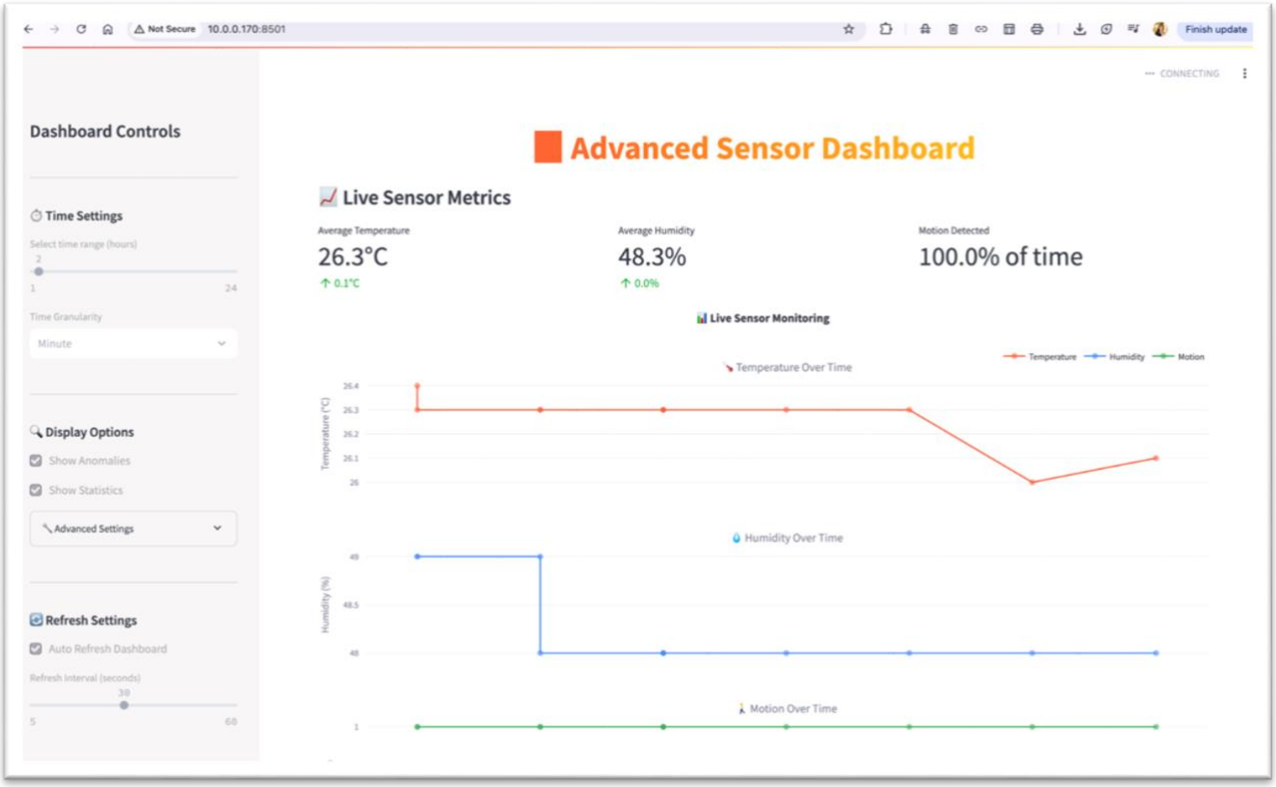
```
streamlit run dashboard.py
```

The dashboard visualizes live data and displays detected anomalies using interactive graphs.

Dashboard Features

The dashboard provides:

- Real-time temperature, humidity, and motion sensor readings
- Historical data visualization with customizable time ranges
- Anomaly detection visualization with highlighted anomalous periods
- Data export functionality (CSV, Excel)
- Interactive plots for detailed analysis



Dashboard Controls

⏱ Time Settings

Select time range (hours): 2 (1–24)

Time Granularity: Minute

🔍 Display Options

☑ Show Anomalies

☑ Show Statistics

Advanced Settings

Refresh Settings

☑ Auto Refresh Dashboard

Refresh Interval (seconds): 30 (5–60)

Advanced Sensor Dashboard

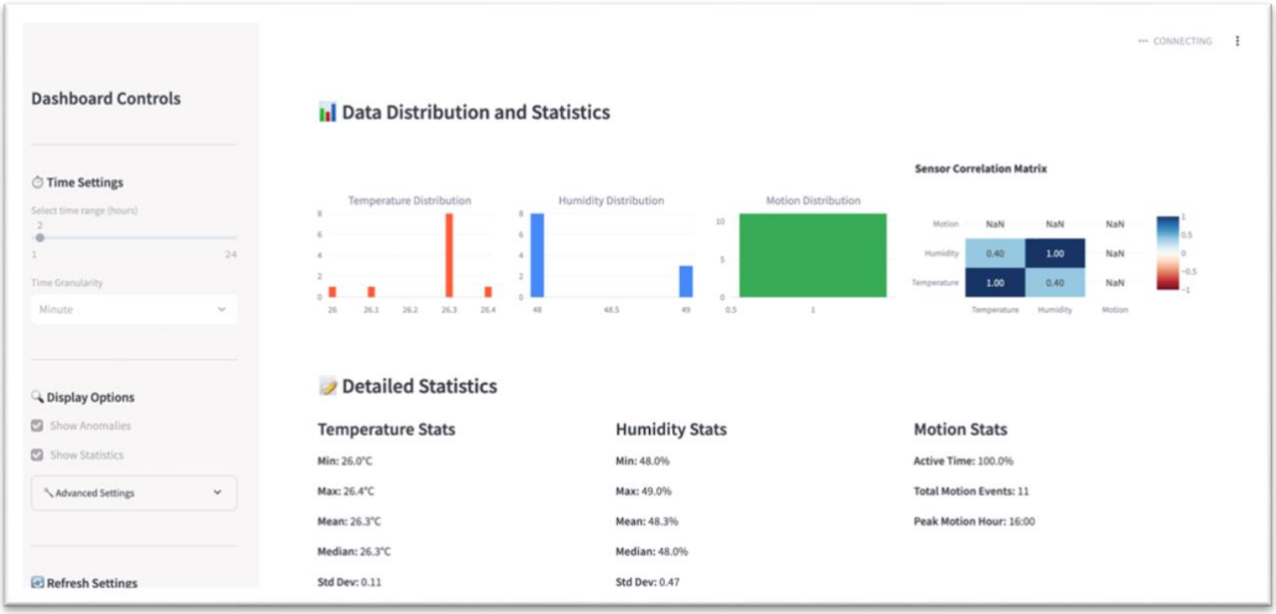
📈 Live Sensor Metrics

Average Temperature: 26.3°C ↑ 0.1°C

Average Humidity: 48.3% ↑ 0.0%

Motion Detected: 100.0% of time

📊 Live Sensor Monitoring



Dashboard Controls

⏱ Time Settings

Select time range (hours): 2 (1–24)

Time Granularity: Minute

🔍 Display Options

☑ Show Anomalies

☑ Show Statistics

🔧 Advanced Settings

🔄 Refresh Settings

📊 Data Distribution and Statistics

Sensor Correlation Matrix

	Temperature	Humidity	Motion
Motion	NaN	NaN	NaN
Humidity	0.40	1.00	NaN
Temperature	1.00	0.40	NaN

📝 Detailed Statistics

Temperature Stats

Min: 26.0°C

Max: 26.4°C

Mean: 26.3°C

Median: 26.3°C

Std Dev: 0.11

Humidity Stats

Min: 48.0%

Max: 49.0%

Mean: 48.3%

Median: 48.0%

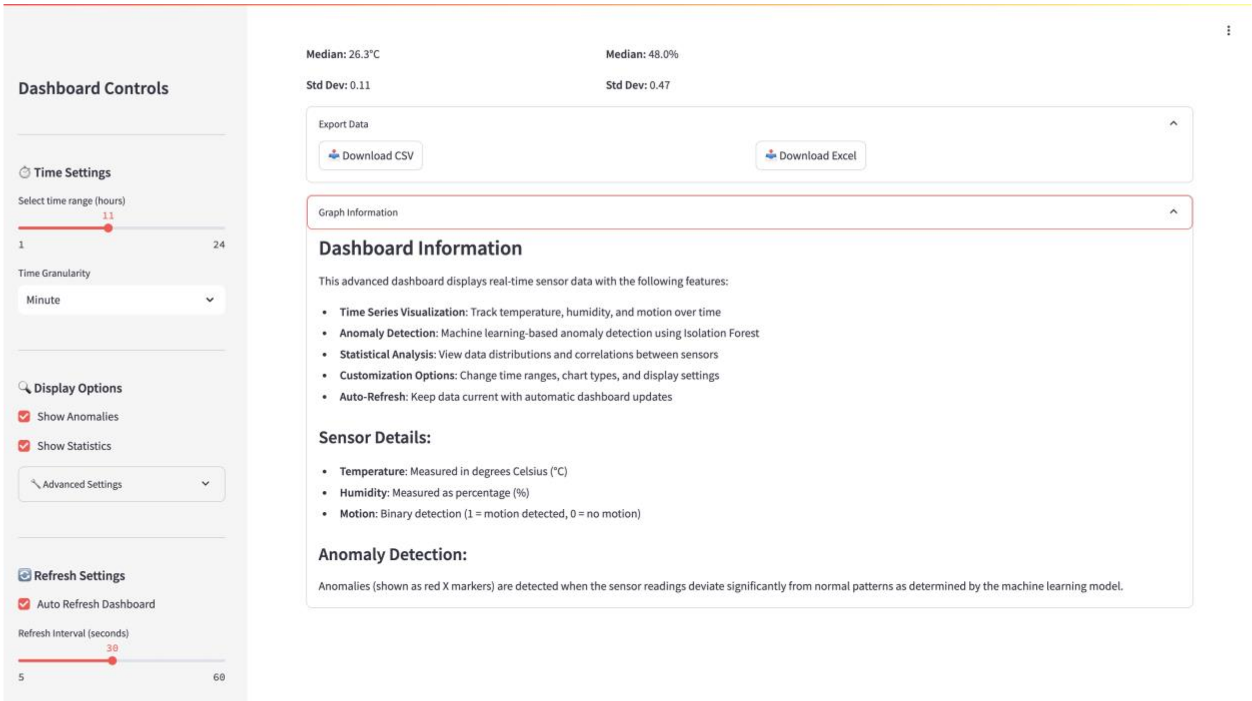
Std Dev: 0.47

Motion Stats

Active Time: 100.0%

Total Motion Events: 11

Peak Motion Hour: 16:00



Dashboard Controls

⏱ Time Settings

Select time range (hours): 11 (1–24)

Time Granularity: Minute

🔍 Display Options

☑ Show Anomalies

☑ Show Statistics

🔧 Advanced Settings

🔄 Refresh Settings

☑ Auto Refresh Dashboard

Refresh Interval (seconds): 30 (5–60)

Median: 26.3°C

Std Dev: 0.11

Median: 48.0%

Std Dev: 0.47

Export Data

Graph Information

Dashboard Information

This advanced dashboard displays real-time sensor data with the following features:

- **Time Series Visualization**: Track temperature, humidity, and motion over time
- **Anomaly Detection**: Machine learning-based anomaly detection using Isolation Forest
- **Statistical Analysis**: View data distributions and correlations between sensors
- **Customization Options**: Change time ranges, chart types, and display settings
- **Auto-Refresh**: Keep data current with automatic dashboard updates

Sensor Details:

- **Temperature**: Measured in degrees Celsius (°C)
- **Humidity**: Measured as percentage (%)
- **Motion**: Binary detection (1 = motion detected, 0 = no motion)

Anomaly Detection:

Anomalies (shown as red X markers) are detected when the sensor readings deviate significantly from normal patterns as determined by the machine learning model.

Common Issues and Solutions

1. Missing scaler.pkl file
   - Error: FileNotFoundError: [Errno 2] No such file or directory: 'models/scaler.pkl'
   - Solution: The realtime_detector.py script will automatically create a scaler if missing, but you can also manually run train_model.py again to generate both the model and scaler.
2. Sensor Reading Errors
   - If you see [WARN] DHT Read Error messages, check physical connections and wait for sensor stabilization.
   - For persistent errors, try powering off and on the Raspberry Pi.
3. Column Name Mismatches
   - If you encounter ValueError: Feature names unseen at fit time errors, ensure the column names in realtime_detector.py match those used during training.
4. Dashboard Not Loading
   - Ensure that all required packages are installed, especially matplotlib, seaborn, and XlsxWriter.
   - Check if Streamlit is properly installed and the dashboard.py file has no syntax errors.

Conclusion:

This capability leverages the power of Python and Raspberry Pi to create a real-time anomaly detection system using environmental sensors. It provides insights into sensor data through live graphs and detects any abnormal behavior, making it ideal for monitoring applications that require real-time data analysis.