HVAC Anomaly Detection and Fault Diagnosis Using ML and LLMs

Debrup Ghosh

June 26, 2025

1. What is HVAC?

HVAC stands for **Heating, Ventilation, and Air Conditioning**. It refers to the technology and systems used to control indoor environmental conditions such as temperature, humidity, and air quality in residential, commercial, and industrial buildings.

Core Features of HVAC Systems:

- **Heating**: Maintains optimal indoor warmth during cold weather.
- **Ventilation**: Ensures proper air circulation and filtration.
- **Air Conditioning**: Provides cooling and dehumidification during hot conditions.
- **Automation**: Uses sensors and controllers for energy-efficient operation.
- **Monitoring**: Supports IoT-based real-time monitoring of environmental and system parameters.

Modern HVAC systems often incorporate AI, fuzzy logic, and optimization algorithms to improve energy efficiency and comfort.

2. Dataset Description

The dataset used in this project is publicly available on Kaggle:

- **Title:** HVAC Sensor Data for Dynamic Fuzzy PID Control
- **URL:** https://www.kaggle.com/datasets/ziya07/hvac-sensor-data-for-dynamic-fuzzy-pid-control

Key Features of the Dataset

1. **Environmental Data**:
    - `Temperature (°C)`
    - `Humidity (%)`

- `CO Levels (ppm)`
- `External Temperature`
- `Occupancy Count`

2. **HVAC Control Variables**:
   - PID Controller Gains: `Kp`, `Ki`, `Kd`
   - Fuzzy Logic Modifiers
   - Optimization Scores from the Intelligent Seagull Algorithm (ISA)
3. **System Performance Metrics**:
   - Power Consumption
   - Cooling/Heating Output
   - Response Time
   - Energy Efficiency
   - User Comfort Index
4. **Target Column:** `HVAC Efficiency Class`
   - 0 = Low Efficiency
   - 1 = Medium Efficiency
   - 2 = High Efficiency

Use Cases

- Smart HVAC optimization using AI-based controllers.
- Energy consumption analysis for sustainable industrial environments.
- IoT-based predictive modeling for intelligent HVAC control.

3. Project Approach: Outlier Detection

For this project, we aimed to detect anomalous behavior in the HVAC system based on sensor readings.

Selected Features

The following input features were used to detect outliers:

- **Temperature (°C)**: Internal temperature recorded by sensors.
- **Humidity (%)**: Humidity level affecting thermal comfort.
- **CO Content (ppm)**: Indicates air quality and occupancy level.
- **Occupancy Count**: Number of people present in the space.

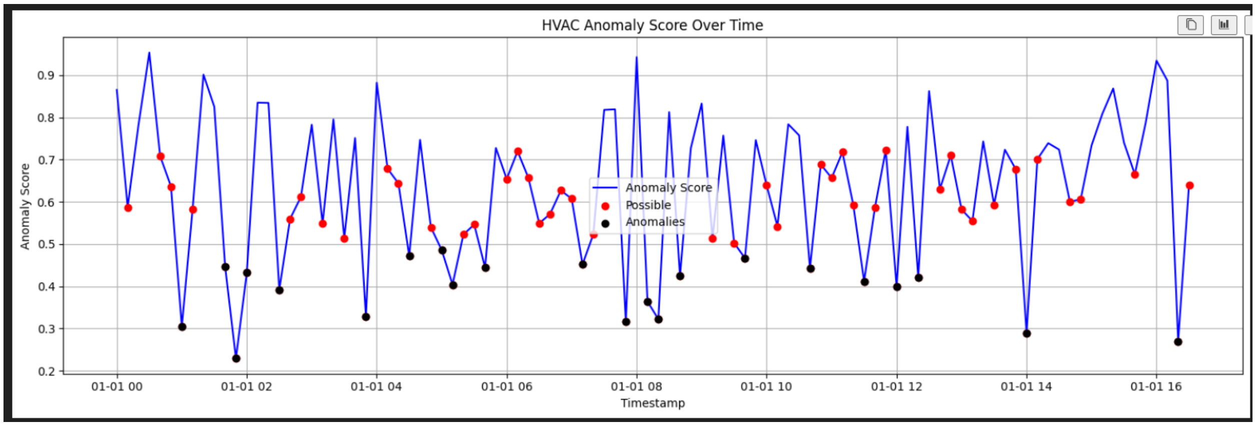

Figure 1: Outlier Detection

Methodology

- An **Isolation Forest** model was trained on the dataset to learn normal patterns of operation.
- The anomaly score produced by the model was scaled using `MinMaxScaler`.
- Threshold-based classification was applied:
  - Score ¡ 0.5: Critical Anomaly
  - 0.5 $\leq$ Score ¡ 0.7: Warning
  - Score $\geq$ 0.7: Normal
- For anomalous cases, a prompt was sent to a Large Language Model (LLaMA-3.3) using a predefined `PromptTemplate`, requesting fault diagnosis based on sensor values.

Evaluating Anomaly Detection with Efficiency Labels

To evaluate how well the unsupervised anomaly detection model correlates with HVAC system inefficiency, we created a binary classification comparison. The assumption was that most anomalies should correspond to low efficiency, and normal cases should correspond to high efficiency.

Transformation and Accuracy Calculation

We performed the following steps:

1. Marked points with an anomaly score below 0.5 as extreme anomalies.
2. Converted HVAC Efficiency Class such that:
   - 0 $\rightarrow$ High Efficiency (mapped to 1)

- 1 → Medium or Low Efficiency (mapped to 0)

3. Compared predicted anomaly labels with the transformed actual labels to compute binary accuracy.

The Python code used is shown below:

```
df["Extreme"] = (df["Anomaly_score"] < 0.5)
df["EXtreme"] = df["Extreme"].astype(int)

def transformation(x):
    if x == 0:
        return 1
    else:
        return 0

df["Y_actual"] = df["HVAC_Efficiency_Class"].apply(transformation)
accuracy = (abs(df["Y_actual"] == df["EXtreme"])).mean()
```

Interpretation

This code compares predicted anomaly labels (unsupervised) with expected low-efficiency labels. The transformation ensures consistency between "bad cases" being labeled as '1' in both predictions and ground truth. The final accuracy represents the percentage of correct matches between anomaly detection and actual system inefficiency.

This shows that anomaly detection aligns strongly with real-world HVAC inefficiency, despite being trained without supervision. In our dataset, we found an accuracy of approximately **70%**, validating the effectiveness of our anomaly detection model.

Confusion Matrix Interpretation

The confusion matrix shows that:

- When the model predicts **normal operation**, the system is highly likely to be operating with **high efficiency** (True Negative = 6742).
- When the model predicts an **anomaly**, only a fraction (306 out of 2150) actually correspond to low efficiency (True Positive rate ≈ 14%).

This suggests that:

> The model is effective in ruling out inefficient systems. If no anomaly is detected, there is a high confidence that the HVAC system is functioning efficiently. However, an anomaly does not always imply inefficiency — it may also reflect sensor noise or other operational deviations.

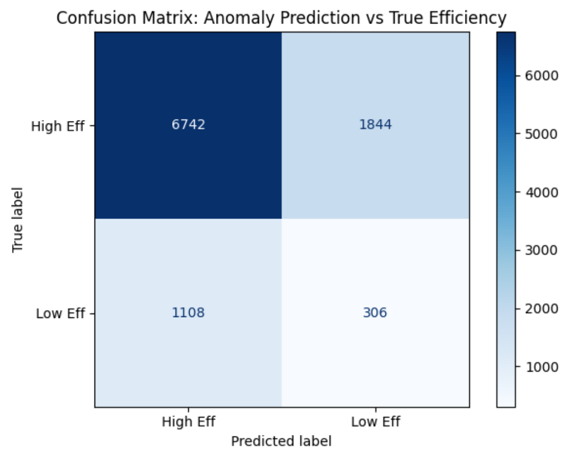

Figure 2: Confusion Matrix

4. Workflow

This section outlines the step-by-step workflow used for real-time HVAC anomaly detection and diagnosis. The system is designed to process live sensor data, detect anomalies, and retrieve contextual knowledge using a Retrieval-Augmented Generation (RAG) pipeline powered by a large language model (LLM).

Step-by-Step Pipeline

1. **Sensor Input:** The system receives continuous input from environmental sensors, including temperature, humidity, CO levels, and occupancy.

2. **Real-Time Anomaly Detection:** Incoming data is analyzed by an unsupervised anomaly detection model (e.g., Isolation Forest). If the model detects an outlier (anomaly score below threshold), an alert is triggered.

3. **Query Generation:** Based on the sensor input that caused the anomaly, a natural language query is programmatically generated to describe the current condition.

4. **Semantic Retrieval:** A retriever performs semantic search over a vector database (built from standard operating procedures, fault manuals, etc.) to retrieve relevant documents or context related to the anomaly.

5. **LLM-Based Diagnosis:** The retrieved documents and current sensor context are sent to a large language model (e.g., LLaMA) which performs reasoning and provides likely root causes, fault classification, and suggested actions.

6. **Feedback or Action:** The system can optionally log this diagnosis, notify the user, or recommend further steps for automated resolution or human intervention.

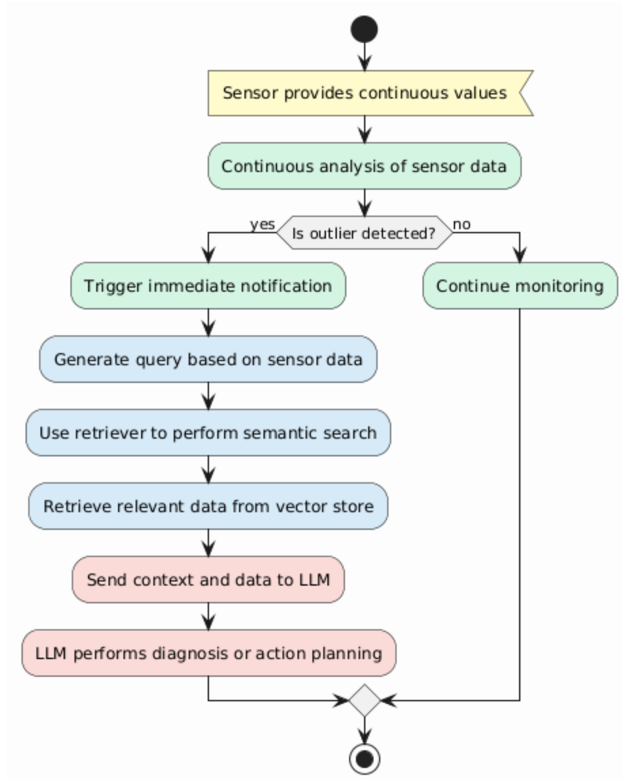

Figure 3: WorkFlow

Activity Diagram

The following activity diagram represents the overall control flow of the system:
As shown above, the workflow starts with continuous sensor inputs and progresses through anomaly detection, semantic retrieval, and intelligent diagnosis by an LLM. Each component plays a crucial role in enabling intelligent, real-time fault detection and diagnosis for HVAC systems.

5. Conclusion

This project demonstrates how machine learning models can be integrated with large language models (LLMs) and Streamlit interfaces to build intelligent systems for real-time HVAC anomaly detection and fault diagnosis. Such AI-enhanced solutions enable predictive maintenance, reduce energy waste, and improve system reliability and user comfort in industrial environments.