IntelliHack 5.0

Machine Learning hackathon

Q1 - Report 02

The Code Rushers

01. System Diagram

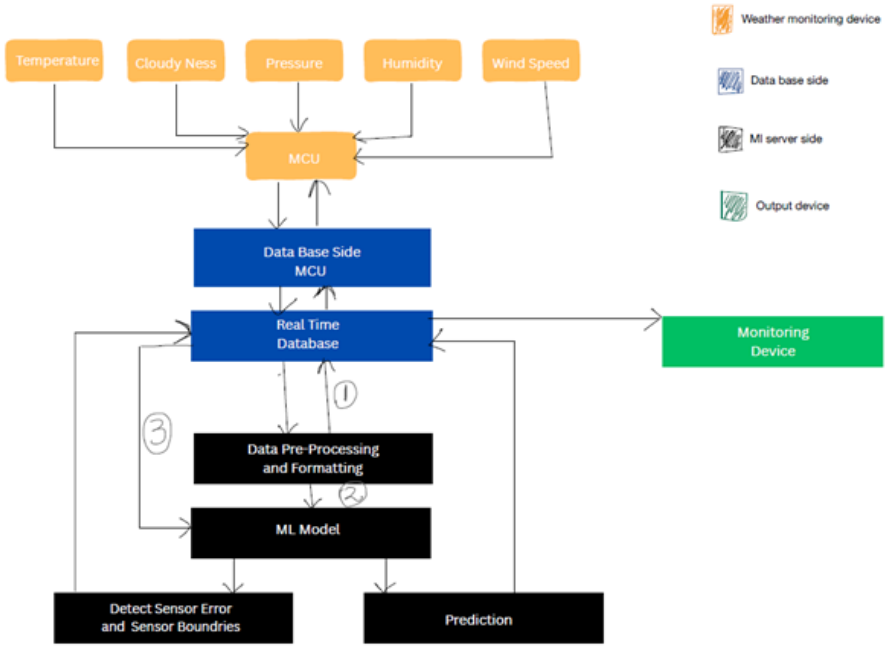

Figure 1: System Diagram

1.1 Weather Monitoring Side (Orange)

Depending on the facility and the required level of accuracy, more IoT weather monitoring stations can be implemented. Keeping this in mind, the design follows a modular pattern. In this system, data from all sensors is first sent to the IoT device's MCU. The device then performs an initial validation check to ensure data accuracy by comparing it with predefined range boundaries provided by an ML model. These boundaries are retrieved from a real-time database. If a long-term error is detected, the system updates the database with the sensor's error status, allowing the monitoring device to identify the faulty sensor and alert the user. This improves maintenance efficiency and enhances system reliability. Once validated, the data is transmitted to the database using remote connections.

1.2 Real-Time Database Side (Blue)

The Real time DB is divided into two main parts: the MCU and the database. the Database Side MCU collects data from all-weather monitoring stations and stores it in a well-organized manner. This approach ensures modularity and maintains a synchronized data flow, improving efficiency and scalability.

1.3 ML Side (Black)

1. **Data Preprocessing**: All incoming data undergoes pre-processing and formatting to fit the required structure for the model to function effectively. Once processed, the data is updated back into the database.
2. **Data Retrieval**: The ML model retrieves the current weather data from the database and accesses previous data for analysis.
3. **Prediction and Analysis**: Using the retrieved data, the model predicts new results while analyzing sensor errors and updating sensor reading boundaries. These updated boundaries are stored in the database, allowing the database MCU to transmit them to the respective weather station MCUs. This dynamic updating process ensures that sensor thresholds remain adaptive.
4. **Error Detection and Alerts**: If a sensor error is detected, the database triggers an alert to the user's device, enhancing system reliability and enabling automatic maintenance.
5. **Continuous Learning**: The model's predictions are stored in the database, facilitating continuous learning from new data. This ongoing learning process improves accuracy over time.
6. **Scalability**: Since predictions are stored in a common database, multiple user devices can access them simultaneously, ensuring the system's scalability.

02. Features of the System

The system is designed with the following key features:

- **Scalability**: The system supports multiple user devices and weather stations, ensuring seamless expansion without compromising performance.
- **Modularity**: The modular design enables easy debugging and maintenance, allowing components to be updated or replaced without affecting the entire system.
- **Reliability**: Multiple validation mechanisms for input data, along with the ability to dynamically update these mechanisms, enhance the model's reliability in real-world applications.
- **Accuracy**: The combination of reinforcement learning, reliable data input mechanisms, and synchronization across multiple weather stations improves the overall accuracy of the system.