Wind Turbine Data Processing Pipeline

1. Overview

This pipeline processes wind turbine data to clean, analyze, and detect anomalies.

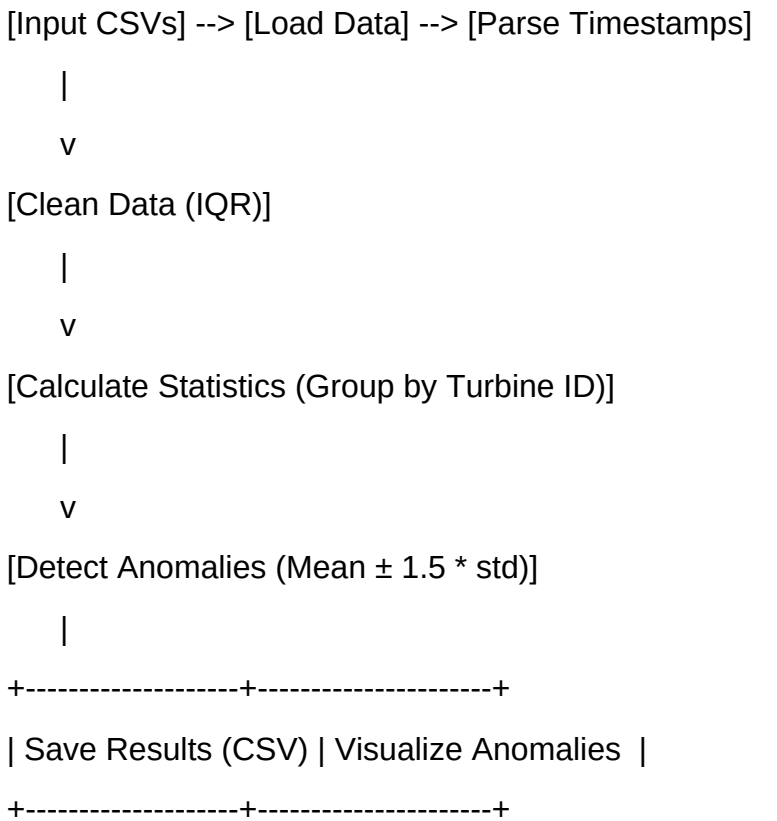
It includes steps for data loading, cleaning, statistical analysis, anomaly detection, and visualization.

2. Workflow

1. Load Data: Combine multiple CSV files into a single dataset.
2. Parse Timestamps: Convert timestamps to datetime and remove invalid rows.
3. Clean Data: Handle missing values and remove outliers using the Interquartile Range (IQR).
4. Calculate Statistics: Compute min, max, mean, and standard deviation for each turbine.
5. Detect Anomalies: Identify anomalies in power output using thresholds.
6. Save Results: Export cleaned data, statistics, and anomalies to CSV files.
7. Visualization: Create time-series and bar plots for analysis.

3. Architecture

The architecture of the pipeline follows a structured data processing approach:



4. Why Use 1.5 Instead of 2?

The threshold of 1.5× standard deviation (std) instead of 2× is chosen to
balance sensitivity and specificity in anomaly detection. A threshold of 2× may only flag extreme outliers,
missing early warning signs of turbine failure. Using 1.5× allows for detecting small but significant deviations,
ensuring proactive maintenance.

Comparison:
- **1.5 × std:** More sensitive, detects minor irregularities before failure.
- **2.0 × std:** Less sensitive, only catches major anomalies.
Using 1.5× helps detect potential turbine issues early while reducing false positives.

5. Outcomes

1. Cleaned and processed turbine data stored in CSV format.
2. Statistical summaries (min, max, mean, std) for each turbine.
3. Detected anomalies in power output for further investigation.
4. Visualizations highlighting anomalies and turbine performance.
5. Scalable pipeline for analyzing large datasets and detecting faults.

6. Scalability: Python, PySpark, CI/CD, and Workflow Automation

This pipeline is implemented using **Python**, but it can also be adapted to:
- **PySpark**: For handling large-scale turbine datasets efficiently.
- **CI/CD Pipelines**: Automating data ingestion, processing, and reporting.
- **Workflow Automation**: Streamlining tasks for real-time monitoring and alerts.
- **Azure Databricks**: Leveraging cloud-based big data processing and machine learning capabilities.