**Textual Explanation for Time Series Fault Detection**

**Dataset can be found here:** [**https://www.kaggle.com/datasets/raysentinel/metropt-2022**](https://www.kaggle.com/datasets/raysentinel/metropt-2022)

**1. Overview**

This project is to design and implement a system that can translate complex time series fault detection data into user-readable text explanations.

We have provided the MetroPT dataset as an example. The dataset includes time series sensor data from a train and a PDF listing fault events (shown below) along with other dataset information.

Your system should allow users to explore and understand the dataset in plain language by automatically generating clear, concise summaries and explanations of trends, anomalies, and fault events.

It is encouraged to design the solution so that it runs with lower compute requirements. Lightweight approaches, efficient data handling, and resource-friendly methods are preferred over large or complex solutions that are difficult to run on minimal hardware.

A table with numbers and a number on it

AI-generated content may be incorrect.

**2. Deliverables**

**A. Working Prototype**

* A system that can explain dataset information in natural, user-readable text.
* Prototype should be runnable in Google Colab.
* The system should demonstrate:
  + Fault detection insights (for example, when and where faults occur).
  + Explanations of time series trends (for example, unusual spikes, sensor patterns).
  + Plain-language explanations suitable for non-technical users.

**B. Documentation**

* A README file including:
  + Setup instructions.
  + Dependencies (libraries, frameworks, models).
* A diagram explaining:
  + Data processing pipeline (for example, Pandas, NumPy, statistical or ML methods).
  + Any supporting tools (visualization libraries, summarization approaches).
  + Data flow from raw dataset → text output → user explanation.

**C. Demonstration**

* A short video (about 1 minute) showing the system in action:
  + Loading and processing the dataset.
  + Generating example text explanations.

**D. Write-up**

* A report (1–4 pages) covering:
  1. What works: Strengths, features completed.
  2. What doesn’t work: Limitations, obstacles faced.
  3. Why this approach: Rationale vs. alternatives (for example, rule-based vs. statistical vs. ML).
  4. Next steps: Possible future improvements if more time or resources were available.

**E. Optional Extensions**

If time allows, explore bonus features such as:

* Interactive Chatbot for explaining the data
* Fault visualization dashboards with text summaries.
* Predictive failure modeling with textual explanations.
* Real-time interactive summaries of live data streams.
* Multimodal features combining text with charts or tables.