Documentation

Real-time Anomaly Detection Dashboard

This Python script implements a real-time anomaly detection dashboard using the Dash framework and Plotly library. The application generates synthetic sensor data, simulates anomalies, and visualizes the data points on a web page. The anomalies are detected based on Z-Score thresholding.

>Data Generation and Anomaly Simulation

The application generates synthetic sensor data with a combination of a pattern, seasonal variation, and random noise. Anomalies are introduced randomly with a 7% probability, simulating unexpected spikes in the sensor readings.

>Dash Layout

The layout consists of a title, a real-time scatter plot, and a line plot. An interval component triggers the update of the graph every 500 milliseconds.

i) Components

- Title: "Real-time Anomaly Detection" is displayed at the center of the page with specific styling.
- Graph: The scatter plot shows sensor data points, colored based on whether they are anomalies or not. The line plot represents the sensor data over time.
- Interval Component: It triggers the callback function to update the graph at regular intervals.

ii)Callback Function

The `update_graph` function is a Dash callback that updates the graph every time the interval component triggers. It adds a new data point, calculates mean and standard

deviation, and identifies anomalies based on Z-Scores. The graph is updated with the latest data points, and anomalies are highlighted.

iii) Graph Components

- **Scatter Plot (dots_graph)**: Displays sensor data points with colors indicating anomalies and non-anomalies. Anomalies are highlighted with a larger circle and an orange color.
- **Line Plot (line_graph)**: Represents the sensor data with a green line.
- **Anomaly Circle (anomaly_circle)**: Displays a transparent red circle around anomaly points for emphasis.

iv)Layout Settings

- X-Axis and Y-Axis: Customized range and titles for better visualization.
- Background Color: Light gray background and plot area colors.

v) To Run the Application

If executed as a standalone script, the application runs a local server. Open a web browser and navigate to the provided address (usually http://127.0.0.1:8050/) to view the real-time anomaly detection dashboard.