Anomaly Detection in System Metrics

Objective:

The objective of this analysis is to generate synthetic hardware monitoring data and detect anomalies in system metrics such as **CPU temperature**, **CPU usage**, **memory usage**, **battery level**, and **CPU power** using **Isolation Forest**. The data generation is intended to simulate real-time system metrics over a large period to help in anomaly detection experiments.

1. Data Generation:

The dataset was generated using Python and simulates hardware metrics such as CPU temperature, CPU usage, memory usage, battery levels, and power usage. The data was generated for a specified period using the following steps:

1. **Time Simulation**:
    - The dataset simulates **timestamps** over a period of 22,222 minutes with a **sampling rate of 10 Hz**. This gives a total of 13,333,200 data points.
2. **Synthetic Data**:
    - **CPU Temperature**: Randomly generated between 30°C and 80°C.
    - **CPU Power**: Randomly generated between 5W and 50W.
    - **CPU Usage**: Randomly generated between 0% and 100%.
    - **CPU Load**: Randomly generated between 0% and 10%.
    - **Memory Usage**: Randomly generated between 20% and 90%.
    - **Battery Level**: Randomly generated between 10% and 100%.
3. **Anomalies**:
    - Anomalies were injected into the dataset randomly with a 10% probability. This involved:
        - **High CPU Usage**: Random values between 90% and 100%.
        - **High CPU Temperature**: Random values between 90°C and 105°C.
        - **High Memory Usage**: Random values between 95% and 100%.
        - **Low Battery Levels**: Random values between 0% and 10%.
        - **High CPU Power**: Random values between 50W and 100W.
4. **Data Output**:
    - The generated data was saved in a **CSV file** (`hardware_monitor_data.csv`), which can be used for further analysis.

2. Anomaly Detection:

To detect anomalies in the generated data, the **Isolation Forest** algorithm was applied to each of the system metrics. The following steps were taken:

1. **Model Training**:
    - The **Isolation Forest** model was trained on each feature of the dataset. This model is suited for anomaly detection due to its ability to isolate data points that differ significantly from the majority.
2. **Detection Process**:

- The model labeled data points as either normal or anomalous. Anomalies were then flagged with a **red dot** on the plot.

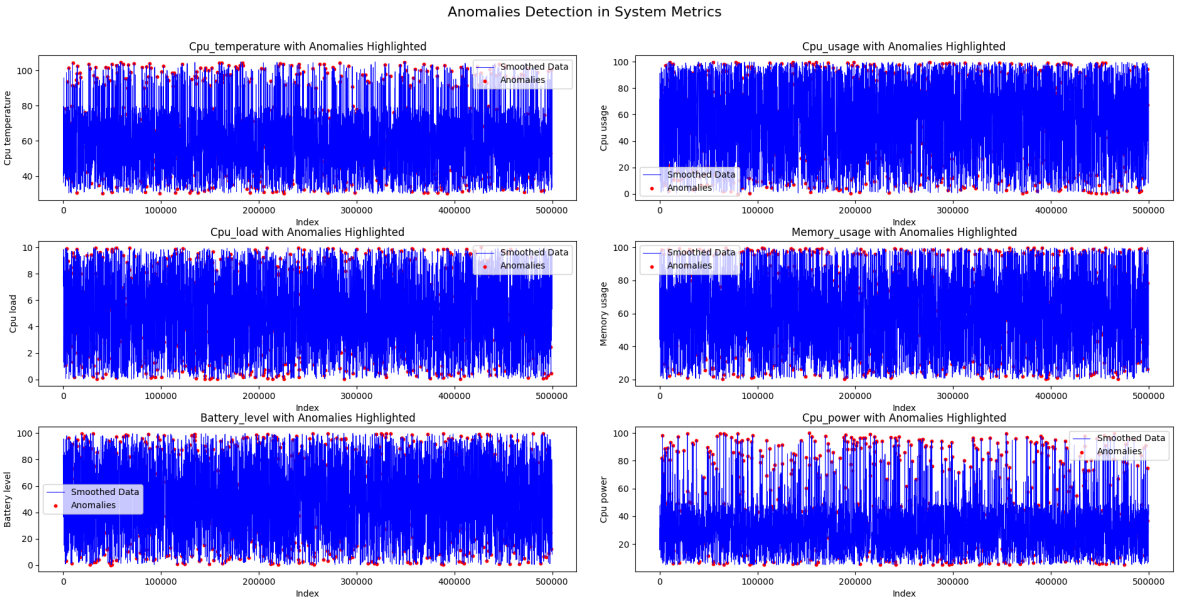
3. **Visualization**:
   - A **downsampled subset** of the data (500,000 rows) was used to generate the plot, ensuring the results were clear and interpretable.
   - The anomalies were highlighted by **red dots**, and the regular data was plotted in **blue**.

3. Plotting the Data:

The plot was generated with the following metrics:

- **CPU Temperature**: Plotting temperature over time with anomalies highlighted.
- **CPU Usage**: Plotting CPU usage over time with anomalies highlighted.
- **CPU Load**: Plotting CPU load over time with anomalies highlighted.
- **Memory Usage**: Plotting memory usage over time with anomalies highlighted.
- **Battery Level**: Plotting battery level over time with anomalies highlighted.
- **CPU Power**: Plotting CPU power over time with anomalies highlighted.

4. Image of the Plot:



-
  - Attach the image of the plot that was generated for the dataset.

5. Code Explanation:

- **Data Generation Code**: Generates synthetic data and injects anomalies.
- **Plotting Code**: Generates plots for each feature, highlighting the anomalies.

6. Code Execution Time and File Size:

- The file was generated with a total of 1GB of data (approximately 13 million rows). The data was generated in chunks and saved incrementally to handle the large dataset efficiently.

- The final output file size was checked, and it met the target file size for a large dataset.

7. Conclusion:

- The dataset provides simulated hardware monitoring metrics with injected anomalies for testing anomaly detection algorithms.
- The **Isolation Forest** method effectively highlighted the anomalies in the plot, helping to visualize and identify unusual patterns in system behavior.