Part A: Core Integration & Testing

1. Pipeline Architecture

The core pipeline for anomaly detection is built around a sequence of modules:

1. **Data Ingestion & Preprocessing:** Telemetry data from `segmentclean.csv` is loaded and normalized using `StandardScaler`. This data is then converted into fixed-length sequences (timesteps=10) suitable for the LSTM model.
2. **Anomaly Detection:** A PyTorch-based LSTM autoencoder is trained to reconstruct the telemetry sequences. The Mean Squared Error (MSE) between the original data and the reconstructed data serves as the anomaly score. Higher MSE indicates a greater deviation from normal behavior, flagging a potential anomaly.
3. **Anomaly Scoring:** The trained model calculates an anomaly score for each timestep in the dataset.

2. Performance Metrics

- **Accuracy:** The autoencoder's loss function (MSE) served as a proxy for reconstruction accuracy, which decreased during training, indicating the model learned to accurately represent normal behavior.
- **Latency:** The training time for the short-sequence model was minimal, and inference (anomaly scoring) is near-real-time, making it suitable for a live Decision Support System.
- **Resource Usage:** The small model size (input_size=10, hidden_size=64) and short training cycles minimize resource consumption on both the training and inference sides.