Matthew Lewis

Intelligence Data Pipeline

Project Introduction:

In the fast-paced domain of defense and aerospace intelligence, processing aircraft telemetry data efficiently is essential for identifying anomalies and supporting mission-critical decisions. This project is my effort to build a lightweight, self-contained data pipeline that simulates telemetry ingestion, processing, and analysis without relying on real-world data sources, which can be sensitive or unpredictable. By generating synthetic aircraft data and applying machine learning for anomaly detection, it creates a controlled environment for testing intelligence workflows, making it a valuable tool for prototyping defense systems.

The core focus is on mimicking real telemetry pipelines such as those used in ISR (Intelligence, Surveillance, and Reconnaissance) while enabling quick iteration on features like data engineering and model tuning. This serves as a bridge between theoretical data science concepts and practical applications in high-stakes sectors like aviation security or threat monitoring.

Solution and Core Concept:

To deliver an effective simulation, the project establishes a full ETL (Extract, Transform, Load) pipeline: from synthetic data generation to anomaly scoring, storage, and visualization. It produces telemetry for multiple flights (80% normal, 20% anomalous) with attributes like timestamp, position, altitude, speed, and status, then engineers features such as speed and altitude differences to feed an Isolation Forest model for detecting outliers.

The detection combines unsupervised ML for pattern recognition with configurable thresholds, generating alerts with scores and labels. Results are stored in a lightweight database and served through APIs and a dynamic dashboard, allowing for experimentation without external dependencies. This modular, centralized design reduces complexity while maintaining key causal relationships in data processing, positioning it as a reasoning tool for intelligence pipeline design rather than a full-scale operational system.

Project Goals:

- Create a synthetic telemetry generator for aircraft data
- Perform feature engineering and anomaly detection using ML
- Build a storage system for processed results
- Develop APIs and a web dashboard for real-time monitoring and querying
- Enable easy iteration through scripts, tests, and configurable parameters

- Ensure deployability with Docker and CI/CD for consistent environments

Scope Clarification and Non-Goals:

This project prioritizes simulation efficiency and interpretability over production-scale features. It does not integrate with live telemetry feeds, handle encrypted data, or perform advanced visualizations like geospatial mapping, focusing instead on abstracted data flows to keep it compact and educational. These choices make it a design tool for hypothesizing intelligence workflows, not a deployable mission system.

Tech Stack:

- Python (core language for all components)
- Pandas and PyArrow (data processing and Parquet handling)
- Scikit-learn (for Isolation Forest ML detection)
- SQLite (data storage)
- FastAPI and Uvicorn (web server and API)
- Jinja2 (HTML templating)
- Pytest (unit testing)
- Docker (containerization)

Implementation:

The pipeline is structured as a modular workflow, with key scripts handling specific stages:

1. Data Generation (pipeline/ingest.py): Creates CSV files with timestamped telemetry for multiple flights, including normal and anomalous patterns.
2. Feature Engineering (pipeline/process.py): Loads data into DataFrames, computes differences like speed_diff, sorts by flight, and prepares for modeling.
3. Anomaly Detection (pipeline/model.py): Trains an Isolation Forest, scores anomalies, and stores results in SQLite with flags for quick querying.
4. Web Server and Dashboard (app/main.py, templates/index.html): Serves APIs for JSON anomalies and renders an interactive HTML UI with tables highlighting outliers and summary stats.

Usage is flexible: Run the full pipeline via CLI for tests, APIs for integration, or the dashboard for monitoring. Docker enables easy deployment, with GitHub Actions ensuring CI/CD reliability. A live demo is hosted at https://intelligencedatapipeline-lewis.onrender.com for interactive exploration.

This project demonstrates my ability to build end-to-end intelligence tools, blending data engineering, machine learning, and web services to tackle practical challenges in defense telemetry processing.