ASRC FEDERAL

Developing A Robust Machine Learning Application for Satellite Operations in Operational Environments

Zhenping Li, Ph.D. | January 2022

Agenda

1. **ML Solutions**
2. **Operationalize ML solutions:**
   - Data Patterns and Trainings in Satellite Telemetry
   - ML Application system requirements
3. **Data training in satellite operation environments**
4. **Software Architecture for ML Applications**
5. **Application Deployment**
6. **Summary**

ML Solutions for Satellite Operations

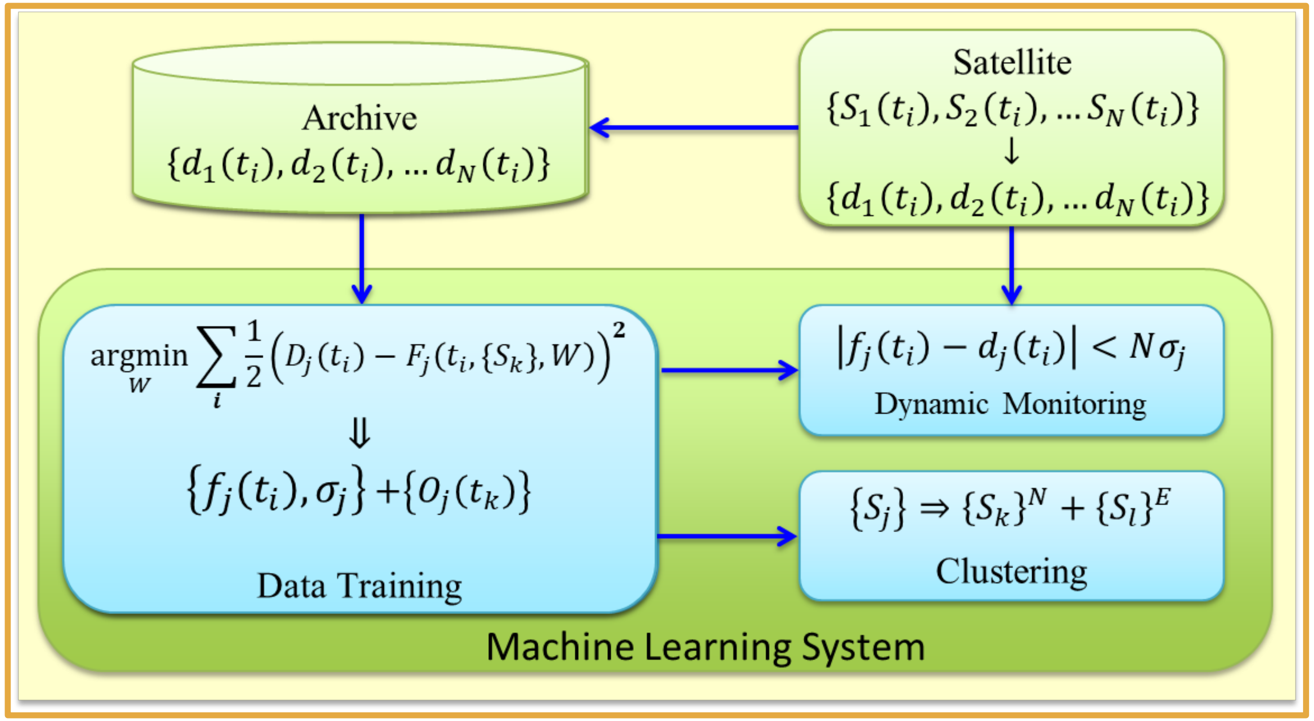
Creating a novel ML analysis tool to characterize and analyze satellite operations:

- Time dependent trends for telemetry consists of
  - Time dependent function that can predict the near future behavior
  - Standard deviations for given training periods.
    - Vs. statistical collections of data in short periods
- Data monitoring by comparing the data values with predictions of time dependent trends
  - Time dependent trends provide tight data bounds that are highly sensitive to deviations of predictions of time dependent trends
    - Vs. static limit monitoring
- Anomaly detections and characterizations that provide direct insights into root causes of anomalies
  - Greatly reduces turnaround time in resolving anomalies
    - Vs. manual troubleshooting in current satellite operations.
- Capture signatures of normal satellite operations in telemetry datasets

Common Architectural Model

- **All ML solutions for satellite operations follow the same ML processes and same functionalities.**
  - Satellites with different orbit may have different algorithm implementations.
- **The data training process obtains time dependent trends for each datasets and detects outliers in datasets**
- **Clustering process analyzes data-training outputs to evaluate data quality and detects anomalies.**

Dynamic monitoring compares the values of an incoming dataset with its predictions for anomaly detections in real or near real time



Data Patterns and Trainings of Satellite Telemetry

Data patterns in satellite datasets are based on orbital characteristics

- Data patterns for datasets with the same orbital characteristics, such as Low Earth Orbit (LEO), should be similar.
- ML algorithms for satellite telemetry are based on orbital characteristics

Data patterns in satellite datasets have both orbital (short-term) and seasonal (long-term) patterns

- Perform data training daily so that training outputs to be more adaptive to seasonal changes.

Data patterns in satellite datasets are diverse.

- No one-size-fits-all ML model

Datasets in OPS environments general contain outliers (or data pattern changes) that could distort data training outcomes.

- Data Training processes not only need to obtain accurate data models but also detect data pattern changes

Operationalize ML Solutions for Satellite Operations

Developing a ML application with following system requirements

- Scalability
  - handle $10^3 \sim 10^4$ or more datasets for a space mission.
- ML Processing Efficiency:
  - ML data training and analysis need to be completed in minutes rather than hours or days.
- Flexibility
  - Allows an ensemble of algorithms for telemetry datasets
  - An algorithm for a specific dataset is determined by its pattern complexity and noise level.
- Rapid deployment:
  - Provide 80% of ML solution, and 20% of customizations
  - Standard APIs for telemetry data inputs and satellite telemetry database inputs.
- Extensibility:
  - A Common ML Framework covers the full life cycle of space missions with different orbits

Incremental Training Operation Concepts

ML operations are running in periodic sessions.

- Data training and clustering processes are running consecutively in each session.

The model parameters generated in previous session are used as inputs for current sessions.

- Changes in data patterns are small from one session to the next.
- Changes in model parameters are small increments.
- Data retraining of models from previous sessions are much more efficient.
  - Critical for the data training of neural networks in OPS environments

Session overlaps in training period ensures stability of data training outcome

Iterative Data Training

Common Data Training process for all ML algorithms

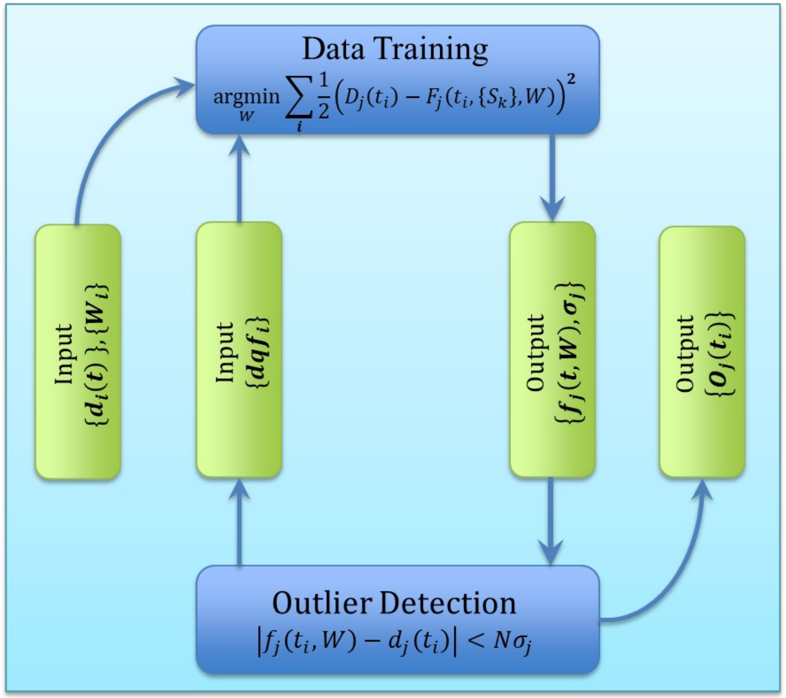
- Removes distortions due to the presence of outliers
- Detects the data pattern changes in data sets.

Input training sets must cover multi-orbit periods

- Pattern changes in one orbit are detected by the pattern differences in multiple orbits.

Iterative Training:

- Associate each data point with a data quality flag ⟦***dqf***⟧_***i*** initialized to 1
- Data training generates time dependent trend {***f_j*** (***t***,***W***),***σ_j*** }
- Outlier detection detects outliers and update data quality flags for each data point
- Perform data training again with updated data quality flags

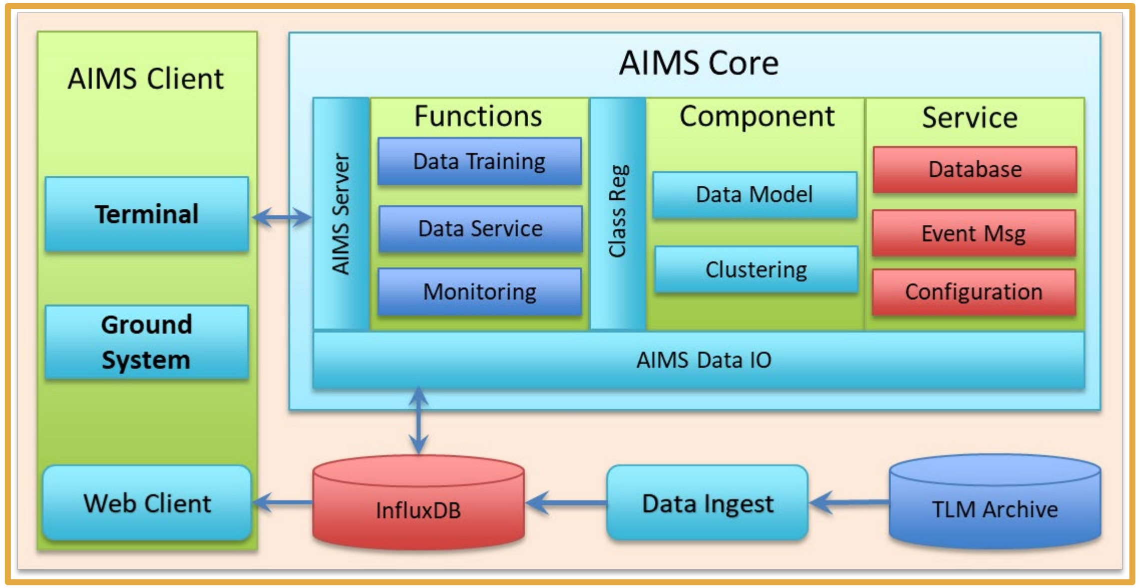


Scalable and Extensible Component Architecture

- **Separate the common services and infrastructure from the mission specific algorithm components**
- **ML algorithms for data training and clustering are treated as plugin and play components with standard API.**
- **Provides flexibility to select pattern specific algorithms for datasets.**
- **Multi-thread processes in data trainings**
- **Global services through standard API:**
  - Database, Global configuration, and event message.

Common Infrastructure:

- Component Container
- Data Input
- Training Output IO
- Data Archive:
  - Time series database



Algorithm Components and ML Database

Two Types of Components:

- ML Model Components:
  - Two subcomponents: inference and data training
  - Implements factory design pattern.
  - Data pattern specific.
- Clustering components:
  - Mission specific.

Component registry:

- Binding algorithm name with class objects.
- Provides API to retrieve an algorithm component for a given algorithm name

ML database contains mission specific information needed for data training

- Data definition:
  - Mapping the TLM hierarchy in a satellite to ML hierarchy
    - $\{subsystem, mnemonic\} \Leftrightarrow \{group, mnemonic, index\}$
- ML attributes for data training and monitoring
  - ML algorithm name and algorithm structure.

Rapid Application Deployment

Developing a ML algorithm repository that covers most data patterns in satellite datasets

Setup ML database for a mission:

- ML database for common telemetry:
  - All missions have common subsystems, such as ephemeris, reaction-wheel, star-tracker, gyro, power, thermal, propulsion, and comm.
    - Each mission may have its own naming conventions
- Mission specific telemetry: telemetry related to payload.
- Map native telemetry database to ML database

Develop an interface between native TLM archive and InfluxDB

- A common interface for EGS ground system

Perform initial training to establish a baseline ML models used in the increment training in normal operations

Summary

The solution has been extended to

- LEO and GEO satellite health and safety monitoring
- Instrument calibration monitoring
- Launch vehicle monitoring

The data training of satellite datasets involves both data training and detections of data pattern changes

- Implements iterative training approach.

System Requirements for a ML software tool for satellite operations:

- Scalability, Extensibility, ML processing efficiency, Flexibility, Rapid deployment.

The approach to address these requirements:

- Increment Data Training for ML processing efficiency.
- Component software architecture for scalability, extensibility, flexibility, and rapid deployment
- ML algorithm component repository and common ML database for rapid deployments.

Quyanaq !

Thank you!

Zhenping Li

zli@asrcfederal.com