

Hera AI FDIR: On-Board Telemetry Anomaly Detection using Machine Learning

F. Voldrich, O. Luschykov, L. Malek, M. Boselli, V. Kudrins

Huld s.r.o., Czech Republic

October 15, 2025



Presentation Outline

- ➊ Introduction
- ➋ Mission Context
- ➌ HERA Dataset
- ➍ Evaluation Strategy
- ➎ Model Performance
- ➏ Uncertainty Quantification
- ➐ LEON3 Integration Challenges
- ➑ Future Improvements
- ➒ Contact

Supported by European Space Agency (ESA) under CCN to Contract No: 4000137902/22/NL/GLC/my

Introduction

- **Spacecraft must continuously monitor subsystem health**
- **Limited downlink capacity** delays ground-based analysis
- Traditional **Out-of-Limits (OOL) alarms** are insufficient for:
 - Anomalous behavior can occur within normal operating ranges
- **Real-time detection** without ground station dependency

Mission Context

- Originally trained on **MEX data** as HERA data were not available (2023 Research)
- Now transitioning to **HERA mission data** (launched October 2024)
- Implemented for **LEON 3** radiation-tolerant processor

Research Evolution

This presentation builds on top of the previous research with MEX data and provides insights regarding the data transition to HERA mission.

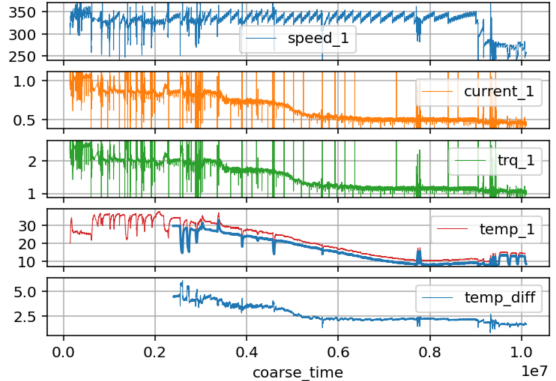
HERA Dataset

Available Data

- **First 4 months** of HERA flight (Oct 2024 - Feb 2025)
- **Reaction Wheel subsystem:** 4 wheels
- **Variables per wheel:** current, torque, speed, internal & external temperatures
- **Measurement interval:** 16 seconds

Key Challenges

- External temperature missing in first month
- Temperature influenced by current/torque
- **Operational command spikes** in data
- No real anomalies available



HERA Reaction Wheel Data

Variable Selection

Temperature Exclusion

- **External influences** on temperature readings
- **Missing data** in the first month
- **High correlation** with current/torque measurements

Final Dataset

- **3 variables per wheel**: current, torque, speed
- **4 wheels** total
- **12 channels** for anomaly detection

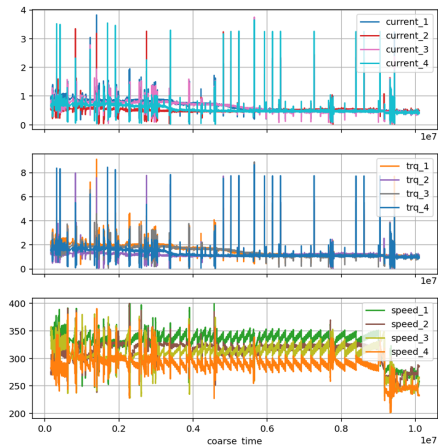
Window Function & Dataset Configuration

Window Function

- Needed to utilize sequential data
- **Sampling:** Every 40th measurement ($16s \times 40 = 640s$)
- **Subsequence length:** 36 (≈ 6 hours at 640s intervals)
- Data dimension: 36×12 (4 wheels $\times$ 3 variables each)
 - 36 samples chosen experimentally (yielding best results)

Dataset Configuration

- Training: **100,000** samples
- Validation: **20,000** samples
- Testing: **20,000** samples



Typical Local Patterns with Spikes

Evaluation Strategy

Synthetic Anomalies

- **Linear Trend** (added to 1 or 2 variables)
- **Random noise** (added to 1 or 2 variables)

Detection Challenge

- **High false positive risk** on legitimate commands
- **Solution:** Reweighting strategy (Boosting-Inspired Approach)

Reweighting Results

Iterative Reweighting Strategy

- 1 Run initial model on entire nominal dataset
- 2 Identify high-scoring subsequences
- 3 Preferentially include them in retraining

Metric	Before Reweighting	After Reweighting
F1 score	0.822	0.815
Precision	0.853	0.736
Recall	0.793	0.912

Performance Before vs After Reweighting

Model Performance - MEX

Two-Stage Approach

- **Siamese AE + KNN** clear winner
- Problem with two-stage approach:
 - Siamese AE is supervised only (need to create anomalies before training)
 - Does not offer connected computational graphs
 - No unified objective

Single-Stage Models

- **Unsupervised learning**
- **Gradient computation**
- **XAI opportunities**

→ For HERA data we move to single-stage models

Model Performance - MEX Results

Encoder	Detector	Precision	Recall	F1 score
Conv. AE	KNN	0.840	0.808	0.804
Conv. AE	I-forest	0.742	0.723	0.718
Conv. AE	Reconstruction error	0.766	0.764	0.764
Var. conv. AE	Reconstruction error	0.895	0.895	0.895
Siamese Conv. AE	KNN	0.995	0.995	0.995
Siamese LSTM	KNN	0.994	0.994	0.994

MEX Model Comparison

Model Performance - HERA

Results

- **Two-stage:** Siamese still works best
- **Single-stage:** CSBSVDD wins

	Model	F1	Precision	Recall
CAKNN	Convolutional Autoencoder with KNN	0.38	0.39	0.37
RAKNN	Recurrent Autoencoder with KNN	0.42	0.84	0.27
SCEKNN	Siamese convolutional encoder with KNN	0.97	0.99	0.95
SREKNN	Siamese recurrent encoder with KNN	0.94	0.96	0.92
CSBSVDD	Convolutional soft-boundary SVDD	0.84	0.90	0.79
COCSVDD	Convolutional one-class SVDD	0.44	1.00	0.28
RSBSVDD	Recurrent soft-boundary SVDD	0.82	0.89	0.77

HERA Model Comparison

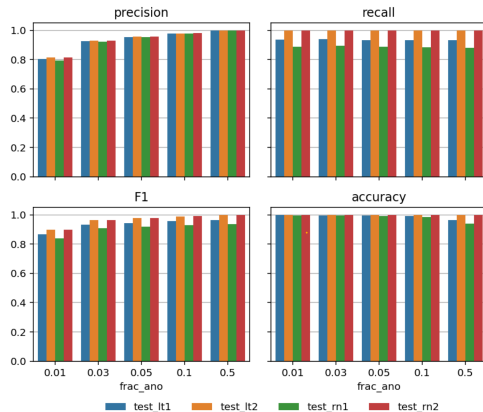
Deep SVDD Anomaly Tests

Test Cases

- **test_lt1/2**: Linear trend added to 1 or 2 variable(s)
- **test_rn1/2**: Random noise added to 1 or 2 variable(s)

Outcomes

- Anomalies affecting two variables are typically easier to detect
- Random noise on single variable (test_rn1) most challenging
- High accuracy due to nominal samples heavily outnumbering anomalous



CSBSVDD Anomaly Detection Results

Uncertainty Quantification (UQ)

Considered but not implemented

- **Dropout**: Not supported by TFLM (TensorFlow Lite for Microcontrollers)
- **Ensemble methods**: Computationally heavy, complicated to decide which model to trust
- **Data augmentation**: Computationally heavy, complicated for data with no anomalies/labels

Implemented

- **Element-wise confidence**: Likelihood of consistent predictions under data variations
- **Inductive Conformal Anomaly Detection**: P-values computed to determine if predictions fall into expected distribution (online/offline approach)

Anomaly Prediction Strategies

Three Outputs

- Model prediction
- Element-wise confidence
- Conformal prediction

Decision Strategies

- **Reckless:** Both detector and conformal predictor agree
- **Cautious:** Either low confidence or detector/conformal signals anomaly

Evaluation

- 7 artificial anomalies: dropout, correlation break, spike, drift, static noise, dynamic noise, channel saturation
- **Cautious methods improve recall** by reducing false negatives (at cost of higher false positives)
- Reflects expected trade-off between sensitivity and specificity

UQ Performance Results

Method	TP	TN	FP	FN
detector_only	466	537	18	89
reckless_online	275	548	7	280
reckless_offline	466	537	18	89
cautious_offline	521	496	59	34
cautious_online	504	512	43	51

Confusion Matrix

Method	Rec.	Prec.	Spec.	FPR	Acc.
detector_only	0.84	0.96	0.97	0.03	0.90
reckless_online	0.50	0.98	0.99	0.01	0.74
reckless_offline	0.84	0.96	0.97	0.03	0.90
cautious_offline	0.94	0.90	0.89	0.11	0.92
cautious_online	0.91	0.92	0.92	0.08	0.92

Performance Metrics

LEON3 Integration Challenges

Technical Challenges

- Models trained in JAX+Flax can be exported to TensorFlow format and run in TFLite but cannot run in TFLite for microcontrollers (TFLM)
- **Big endian support issues** on LEON - byte swapping needed

Solution: New Patch Delivered

- Leverages TFLM's memory management classes
- Ensures compatibility with latest TFLM versions
- Maintains strict static memory allocation
- **Offline byte swapping** - no runtime resource consumption

Memory Footprint

Benchmark Evaluation

- Minimal demo running inference on LEON3
- Memory footprint highly influenced by model operations (kernels)
- **TensorFlow arena**: Only 15.78 KB required
- **SVDD inference faster** than Siamese network from previous research

Operation	Time [ms]
Input prep.	0.047
SVDD inference	99.890
Uncertainty quant.	23.007
Total time	123.082

Time Breakdown

Scenario	text [KB]	data [KB]	bss [KB]	total [KB]
Baseline	10.15	0.14	0.36	1.88
TFLM	245.44	2.34	0.23	248.00
Demo	1140.68	952.87	24.27	2117.56

Memory Footprint

Future Improvements

- Use data for **multiple different subsystems** (not only reaction wheels)
- **Increase number of variables** for each subsystem
- Focus on **explainability** (was not possible in latent space of two-stage model)
- Deploy it on **HERA's second core**

Lukas Malek

Huld s.r.o.

`lukas.malek@huld.io`