Embedding Machine-Learned Anomaly Detection into Relevance-Driven Mission Operations Workflows

Daniele Bellomi
Intella
Milan, Italy
daniele.bellomi@intella.tech

Andrea Guzzo
Intella
Milan, Italy
andrea.guzzo@intella.tech

Valeria Zuccoli
Intella
Milan, Italy
valeria.zuccoli@intella.tech

Enza Magaudda
Intella
Milan, Italy
enza.magaudda@intella.tech

Edoardo Cocci
Telespazio Germany
Darmstadt, Germany
edoardo.cocci@telespazio.de

Jörg Bullmann
Telespazio Germany
Darmstadt, Germany
joerg.bullmann@telespazio.de

Alexandra Lora
Telespazio Germany
Darmstadt, Germany
alexandra.lora@telespazio.de

***Abstract*—Anomaly detection has become a central application of machine learning in satellite operations; however, when applied in isolation, machine learning based detections alone are insufficient to support scalable mission operations. As fleets grow, operators face increasing cognitive load not from anomaly frequency, but from the need to interpret weak signals, correlate them with operational context, and decide whether and how to act. This paper presents an operationally grounded approach in which machine-learned anomaly scores are treated as inputs to a relevance-driven decision workflow rather than as standalone alerts. We describe Mercury, an AI-based mission intelligence engine designed for continuous, mission-adaptive and unsupervised anomaly detection across telemetry streams, producing interpretable anomaly scores to characterize deviations from learned nominal behavior. These scores are not surfaced directly to operators. Instead, they are combined with additional signals from the ground segment—e.g. system state, operational modes, and scheduled activities—through a deterministic relevance-filtering layer. The system applies rule-based logic to combine anomaly probabilities with operational conditions, ensuring that only events with real operational impact are flagged. These operational events are enriched with contextual information required for investigation and decision-making. The architecture is natively integrated with EASE-Rise, a cloud-native mission control and operations platform developed by Telespazio Germany. Telemetry and operational context flow from mission control to the intelligence layer, while prioritized events and recommended command sequences are fed back into the mission control environment, preserving established procedures and operator authority. The approach has been evaluated on representative small satellite operational scenarios, where Mercury has demonstrated actual reductions in false escalations, investigation time, and routine monitoring effort, enabling operations to scale without linear increases in operator workload.**

***Index Terms*—Engineering, Technology, Systems Engineering & Integration, AI/ML in Satellite Data Missions, Ground Systems**

I. Introduction

The operational landscape of small satellites is shifting from single-mission management to the orchestration of large-scale constellations [1]. As telemetry data volume grows rapidly and operational complexity increases (more assets, modes, and mission timelines), traditional limit–checking is becoming insufficient for monitoring the health of complex constellations [2, 3]. While effective for known catastrophic failures, static limits often fail to detect weak signals, subtle deviations in component behavior that precede a failure but remain within absolute safety margins.

Machine Learning (ML) offers a promising solution to detect these non–linear deviations, thanks to a wide range of time series modeling methods which are suitable to manage high volumes of data [4, 5]. However, when applied in isolation, ML-based detections alone are insufficient to support scalable mission operations. In fact, probabilistic models often lack mission context, flagging *statistical* anomalies that are *operationally* nominal (e.g., transient deviations explained by spacecraft mode or scheduled activities) [6]. Flooding operators with non-actionable warnings degrades system trust and increases the likelihood that true anomalies will go unnoticed.

We introduce an approach that contextualizes ML anomaly scores through a relevance–driven workflow: a deterministic logic layer that translates probabilistic outputs into actionable events consistent with established procedures. *Mercury* implements this approach as a mission intelligence decision-support system composed of (i) an unsupervised ML scoring layer and (ii) a deterministic relevance–filtering workflow layer, designed to be integrated with mission control platforms – currently integrated with *EASE-Rise*.

The remainder of this paper is structured as follows: Section II defines the operational problem. Section III details the system architecture of *Mercury* and its integration with

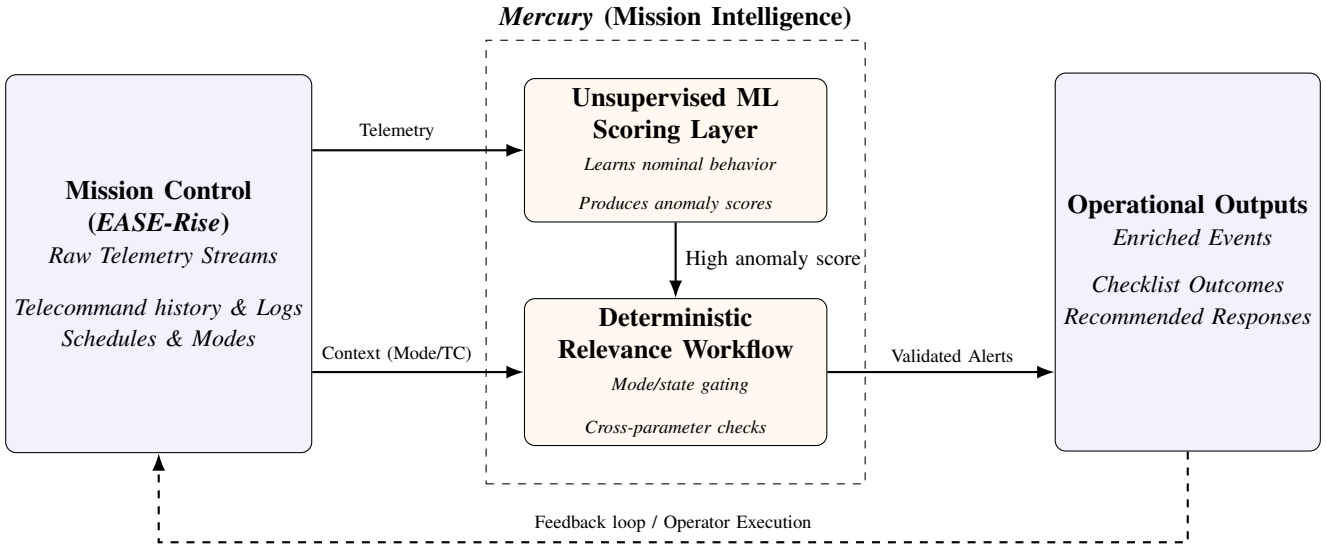

Fig. 1. System architecture illustrating the integration of *Mercury*'s unsupervised ML scoring and deterministic relevance workflow with the *EASE-Rise* mission control platform.

EASE-Rise. Section IV describes the relevance–driven decision workflow that translates ML anomaly scores into actionable events. Section V reports a quantitative evaluation of the unsupervised ML scoring layer using the ESA Anomaly Detection Benchmark as a baseline. Finally, Section VI discusses how this baseline is contextualized through deterministic relevance logic and event construction, and outlines the path toward end–to–end evaluation.

II. The operational challenge

Current satellite operations are heavily based on deterministic rule–based monitoring systems that verify telemetry against predefined thresholds. These systems produce binary "success/fail" outputs: when a parameter exceeds a predefined threshold, an alarm is triggered.

This approach introduces a critical gap: contextual reconstruction across parameters and systems. Specifically, when a binary alarm is triggered, the automated system provides limited insight into cross-parameter correlations and operational relevance. The burden is entirely on the human operator, who must manually reconstruct the context to determine whether the "fail" state is a genuine anomaly or an artifact of known conditions (e.g., mode transitions or scheduled activities). This manual investigation workflow—- retrieving procedures, performing cross–parameter checks, cross–referencing flight plans, and selecting recovery protocols-—is slow, error–prone, and scales poorly with increasing constellation sizes [7, 8]. To reduce false positives and preserve operator trust, probabilistic anomaly indications must be interpreted through a deterministic logic that incorporates mission context and cross–parameter checks.

The ultimate goal is to transition from simple error flagging to relevance–driven decision support systems that can: 1) automatically correlate telemetry deviations with proper context (e.g., satellite states and modes) and, where available, external environment indicators (e.g., radiation spikes) to filter false positives; 2) recommend response procedures directly within the workflow, reducing operator cognitive load [9].

III. System architecture

Mercury, as decision–support system, is architected to reconcile the probabilistic nature of unsupervised machine learning (ML) with the deterministic requirements of satellite operations. As illustrated in Fig. 1, the system is composed of two synergistic engines: an *unsupervised machine learning core* that detects statistical deviations, and a *deterministic relevance-filtering layer* that assesses operational relevance against the mission context.

A. Unsupervised ML engine

The unsupervised ML engine within *Mercury* is designed as a mission–agnostic, unsupervised anomaly detection system. We prioritize an unsupervised approach for three operational reasons:

- **Applicability**: new spacecraft or components often lack the labeled failure examples required for supervised training [4]. An unsupervised model self-learns normal behavior from nominal telemetry.
- **Scalability**: supervised models may not generalize when hardware batches or orbital parameters change slightly, often requiring retraining. Unsupervised learning adapts to the specific statistical footprint of each sensor/satellite combination.
- **Novel anomaly detection**: by focusing on deviations from normality rather than matching known failure patterns, the system can detect previously unseen deviations and novel anomalous behaviors.

Mercury employs cross–sensor learning, correlating data across multiple telemetry channels to establish a holistic view

of the satellite's state [10]. The system computes an *anomaly score* that quantifies the deviation of current observations from the learned baseline. To ensure stability in an operational environment, the scoring logic implements specific heuristics to filter noise:

- **Robust aggregation**: in our current implementation, the system records the third–highest anomaly score among monitored channels per hour, rather than the absolute maximum. This heuristic is motivated by experience with non–curated datasets, where single–point spikes often indicate telemetry corruption or transmission errors rather than physical anomalies.
- **Historical comparison**: this hourly score is compared to the peak values recorded over a rolling historical window. A detection is flagged only if the new score exceeds the historical maximum by a configurable detection factor R (e.g., $R = 1.5$).

B. The relevance-filtering layer

While the unsupervised ML engine provides anomaly score, the relevance–filtering layer provides the operational context. This deterministic engine acts as a logic gate between probabilistic anomaly indications and the operator, producing actionable events aligned with established procedures.

This layer aggregates diverse data sources to assess the operational relevance of detections:

- **telemetry & anomaly scores:** real–time sensor values and their computed ML scores;
- **system events:** reboot counts, mode changes, and telecommand history ingested from the mission control system;
- **environmental indicators:** space weather data (e.g., $K-$index, solar flux) to correlate internal upsets with external radiation events.

Relevance is determined through rule–based conditions that combine anomaly scores with operational state, mode gating, temporal persistence, and cross–parameter checks. When conditions are satisfied, the layer promotes detections into actionable events enriched with investigation context and traceability to the triggering logic.

IV. Workflow implementation

The operational workflow transforms raw data into a decision through a three–phase process. This structure is specifically designed to tame the stochastic nature of machine learning outputs, converting probabilistic indications into deterministic operational reasoning and actionable events.

A. Phase 1: rule logic construction

To bridge the gap between a raw *statistical anomaly* and a valid *operational event*, operators define relevance conditions within *Mercury*, ingesting data from *EASE-Rise*. These conditions act as a stabilization layer for the ML scores and enable cross–parameter checks:

- **Comparison-based:** combines standard hard limits with ML probabilities (e.g., "Warn if ML anomaly score > 0.9 AND battery temperature $> 70\,°C$").
- **Time-based (persistence logic):** this is the primary mechanism for smoothing probabilistic volatility. Operators define temporal persistence conditions (e.g., "Alert only if anomaly score > 0.8 for > 3 consecutive minutes"). This acts as a low-pass filter suppressing transient spikes caused by telemetry noise while capturing sustained deviations that indicate genuine instability.
- **Operational context:** filters based on operational state to prevent false positives during known dynamic events (e.g., "ignore attitude deviation if Mode == Slew Maneuver").

B. Phase 2: actionable outputs

When the conditions encoded in the rule-based logic are satisfied, the system generates one or more actionable outputs:

- **Operational events**: creation of structured events with predefined fields, e.g. status (confirmed/deleted) , importance factor (low/critical), description, which becomes the primary artifact for triage, traceability, and downstream actions.
- **Notifications:** multi-channel alerts routed to the correct role in the operations team to prompt timely review of the event.
- **Checklists items**: explicit success/fail outcomes of formalized rule checks, attached to the event for operator review and auditability.
- **API Hooks:** interfaces to push events, checklist outcomes, and recommended responses/command sequences back to *EASE-Rise* (or other ground systems).

C. Phase 3: event triage and knowledge retention

Operators interact with the generated operational event to classify and annotate the anomaly. By storing investigation comments and preserving traceability to the triggering logic, the operations team builds a searchable logbook. This preserves institutional knowledge, allowing operators to rapidly retrieve past resolutions and significantly accelerate the triage of recurring signatures.

V. Evaluation on ESA benchmark

To establish a quantitative baseline for our architecture, this section evaluates the standalone unsupervised ML detection capabilities prior to any workflow integration. We adopted the ESA Anomaly Detection Benchmark [ESA–ADB; 11], focusing on *Mission1*– a dataset covering approximately 14 years of operation ($\sim 105,000$ hours), with 186 labeled anomalies vetted ex–post by Spacecraft Operations Engineers (SOEs). The end–to–end impact of the relevance–driven workflow on false escalations and investigation time is discussed qualitatively in Section VI and is not quantified by ESA–ADB.

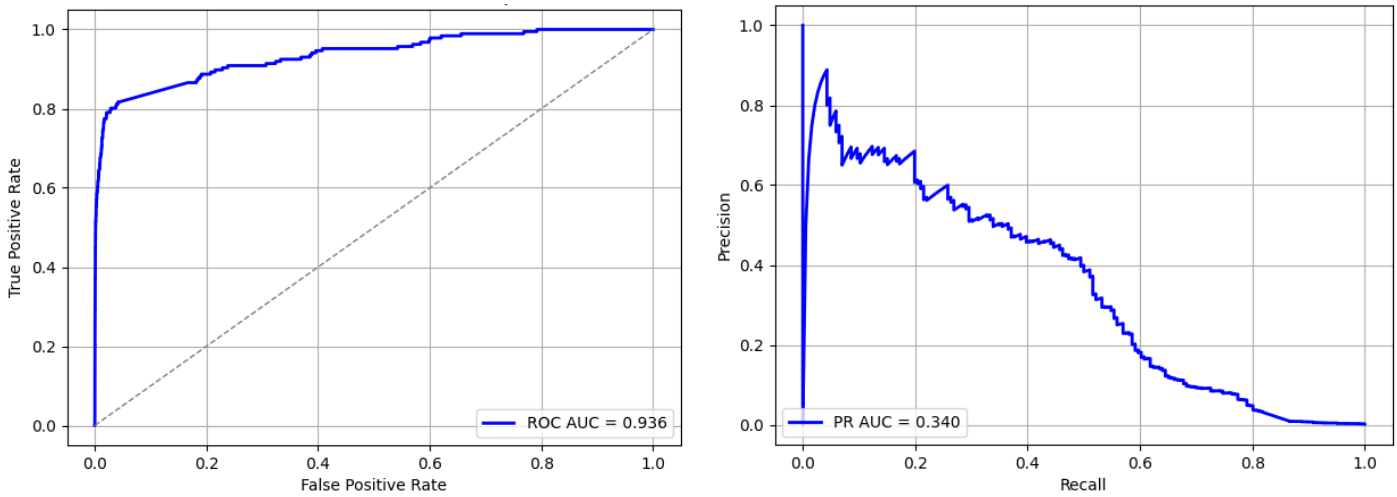

Fig. 2. *Left*: ROC-AUC curve across detection thresholds, demonstrating consistent model discriminability over the full range of operating points. *Right*: Precision-Recall AUC curve. The area under the curve confirms that the model maintains a meaningful precision-recall trade-off, particularly at operationally relevant recall levels.

A. Benchmark setup

The evaluation was designed to simulate a realistic worst–case scenario. To optimize the use of this amount of data relative to the computational resources, the unsupervised ML scoring layer was run with a conservative model configuration. The simulation replicated the full operational lifecycle of the models without any manual intervention-—no operator–triggered retraining, no parameter tuning mid–run.

Key experimental constraints include:

- **Model lifecycle:** models were retrained every six months, compared to monthly retraining in production. This represents a worst–case staleness scenario.
- **Training window:** a sliding window of up to two years of historical data was used, where in early mission years, shorter windows were adopted as data accumulated.
- **No supervision:** the system operated in a fully unsupervised mode, with no access to labeled anomaly data at any point during training or inference.

A successful detection (True Positive) was defined as a flag raised by any monitored channel within a window spanning 12 hours before the recorded anomaly onset up to 2 hours after. A labeled anomaly is identified by its ID in the label file; where the dataset breaks anomalies down by sensor we take the superset of all time intervals, meaning that as soon as one sensor triggers, the anomaly is considered to have begun.

B. Results: detection performance across thresholds

Table I reports the results of the confusion matrix in three reference detection factors: $R \in \{1.5, 1.75, 3.0\}$, using the $[-12\text{h}, +2\text{h}]$ detection window.

The parameter R acts as a sensitivity dial that operators can adjust according to their operational risk tolerance. Adjusting R yields distinct trade-offs, shifting the operating point from high-sensitivity monitoring to strict, conservative alerting:

TABLE I
CONFUSION MATRIX FOR $R \in \{1.5, 1.75, 3.0\}$ AND DETECTION WINDOW $[-12\text{H}, +2\text{H}]$.

	True Anomaly	**True Nominal**
R = 1.5		
Predicted Anomaly	109 (TP)	407 (FP)
Predicted Nominal	77 (FN)	104,895 (TN)
R = 1.75		
Predicted Anomaly	103 (TP)	271 (FP)
Predicted Nominal	83 (FN)	105,031 (TN)
R = 3.0		
Predicted Anomaly	72 (TP)	83 (FP)
Predicted Nominal	114 (FN)	105,219 (TN)

- **Baseline benchmark** ($R = 1.5$): a reference configuration. It successfully captures 109 true positives (detecting nearly 60% of all mission anomalies), but generates 407 false positives over the 14−year mission life.
- **Intermediate operating point** ($R = 1.75$): this setting reduces false positives with preserving most of the baseline recall. By slightly reducing the sensitivity, false positives drop dramatically by 33 % (down to 271, averaging fewer than 1.6 false alarms per month), while only sacrificing 6 true positives compared to the baseline.
- **Conservative alerting** ($R = 3.0$): represents a strict setup for risk-averse operations. False positives are severely suppressed down to just 83 occurrences in 14 years, although the system misses more weak signals, reducing true detections to 72 anomalies.

C. Overall model performance (threshold-independent)

In addition to the results of the confusion matrices in Table I, to evaluate the model's performance across all possible decision thresholds, we report in Figure 2 the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) and the

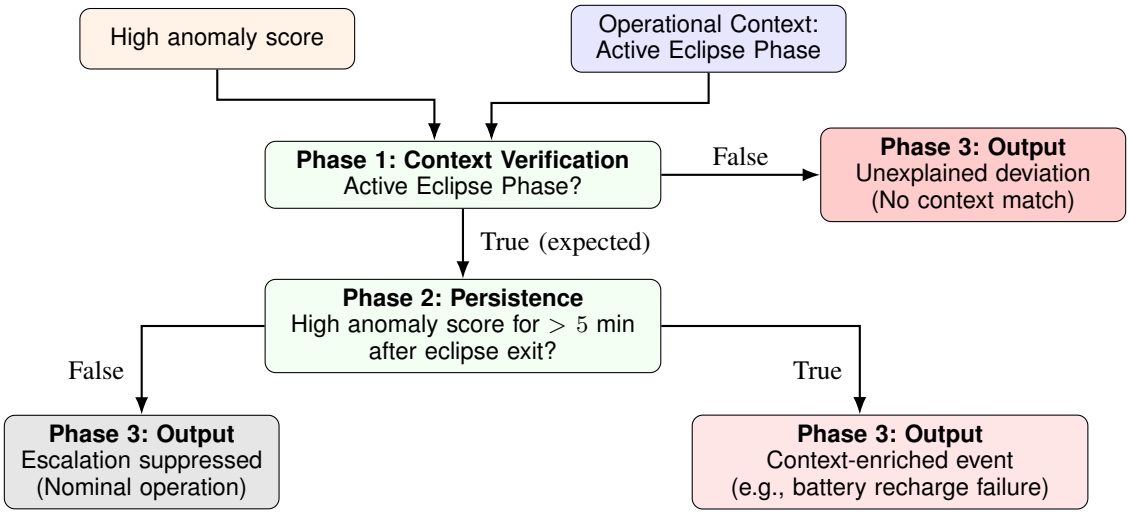

Fig. 3. Relevance-driven workflow example (Section VI-A). A high ML anomaly score acts as a broad-spectrum indicator of statistical deviations. The workflow first verifies whether the deviation is explained by the current mission control context (Phase 1). If unexplained, it is escalated for operator review. If expected (e.g. active eclipse), a temporal persistence condition is applied (Phase 2) to either suppress non-actionable escalation or promote a context-enriched operational event (Phase 3).

Precision-Recall Area Under the Curve (PR-AUC). While the ROC-AUC indicates the model's ability to distinguish nominal from anomalous states, PR-AUC demonstrates the model's robustness against the highly imbalanced nature of spacecraft telemetry datasets, where nominal data vastly outnumber true anomalies [5].

The system achieved a ROC-AUC of 0.936, confirming a highly robust separation between nominal and anomalous states. The resulting PR-AUC of 0.340 demonstrates that the unsupervised model captured meaningful structure in the telemetry distribution and outperforms random chance in a highly skewed dataset.

D. Operational impact analysis (false alarm burden)

At $R = 1.5$, the system generated 407 false positives over 14 years, averaging roughly one false alarm every two weeks (~ 29 per year). This rate should be interpreted as a baseline for standalone ML scoring. While ESA–ADB evaluates only the ML layer, the relevance-driven workflow described in Section IV is designed to reduce the operational burden of false positives through deterministic relevance conditions (e.g., mode/state gating, telecommand history checks, and optimal environmental indicators), providing structured context for triage.

These results quantify the baseline false–positive burden of standalone ML scoring, motivating the deterministic relevance layer introduced in Section IV to operationalize anomaly scores into actionable events.

VI. Translating Statistical Anomalies into Operational Events

While Section V validates the unsupervised ML detection capabilities, the core operational value of the proposed architecture lies in the deterministic relevance–filtering layer. Because the ESA-ADB dataset evaluates anomaly onset only rather than ground–segment workflow interactions, we present a qualitative operational scenario to demonstrate how the deterministic logic (described in Section IV) practically translates statistical deviations into actionable events, thereby mitigating alert fatigue.

A. Scenario definition: the false positive trap

Consider a routine orbital event: a satellite entering an eclipse phase. This transition causes rapid, highly correlated deviations across multiple subsystems—solar array currents drop to zero, battery discharge rates spike, and external panel temperatures plummet.

From a purely statistical perspective, this multivariate deviation is highly unusual compared to the sunlit baseline. This highlights the primary role of the ML engine: detecting any deviation, including deviations that predefined rules cannot anticipate. However, in a traditional standalone ML deployment, detecting this transition independent of spacecraft operational state (e.g. eclipse/sunlit) would immediately trigger a critical alarm, forcing the SPACON to drop their current tasks to investigate what is, in reality, a nominal operational event [8].

With the relevance-filtering layer within *Mercury*, ingesting mission control context, the unsupervised ML detection is complemented by the three-phase workflow (see the sequence diagram in Fig. 3):

1) **Operational context verification:** the rule engine queries the mission control system for the current operational state. Upon identifying an active eclipse phase, the deterministic logic dictates that thermal and power deviations are expected. In contrast, if the context indicates a nominal sunlit state, the rule engine cannot explain the deviation, therefore it classifies the deviation as unexplained by known operational context and it escalates to operator review.

2) **Persistence logic (time-based):** to ensure the system does not ignore a genuine failure that coincidentally happens during an eclipse, the workflow applies a time-based persistence rule: "suppress power anomaly alerts during eclipse mode UNLESS the anomaly score remains critical for > 5 minutes AFTER the eclipse exit."
3) **Actionable output generation:** since the satellite exits the eclipse nominally and the telemetry returns to baseline, the condition is not met. The alert is successfully suppressed.

In practice, a high anomaly score can lead to three triage outcomes:

(i) explained by known operational context $\rightarrow$ suppression;
(ii) unexplained by known context $\rightarrow$ escalation for review as an *unknown* event;
(iii) explained and actionable (known class) $\rightarrow$ escalation as a context–enriched operational event with traceability and recommended procedures (see example in Fig. 3).

B. Operational outcome

Returning to the example in Section VI-A, by bridging the unsupervised ML output with deterministic operational context the workflow yields two key outcomes. First, it stops a purely statistical–yet nominal– deviation from becoming a non-actionable escalation. Second, if a true failure occurred (e.g., the battery failing to recharge), the operator would not receive a generic "*high anomaly score*", but rather a fully contextualized alert: "*Post-eclipse battery anomaly: expected recharge current not observed. Suggested procedure: load-shedding checklist*".

This scenario illustrates how the approach described in this paper shifts the burden of context reconstruction from the human operator to the deterministic workflow layer, supporting the architectural design.

VII. Conclusion

This paper presented a relevance–driven mission operations approach that translates machine–learned anomaly scores into actionable events through deterministic logic. *Mercury* implements this approach as a decision–support system comprising an unsupervised ML scoring layer and a deterministic relevance–filtering workflow. By integrating with mission control platforms (in this work, *EASE-Rise*), the system ingests telemetry context and feeds back prioritized events with recommended responses, all while preserving established procedures and operator authority.

Using the ESA Anomaly Detection Benchmark (ESA-ADB) *Mission1*–a dataset (14 years, $105,000$ hours, 186 labeled anomalies), we established a quantitative baseline for the standalone unsupervised ML scoring layer under conservative lifecycle assumptions (six-month retraining cadence, no manual intervention). At $R = 1.5$, the model detected 109 of 186 labeled anomalies while generating 407 false positives over 14 years (~ 29 per year). Varying the detection factor R provides a tunable precision—recall trade–off, from higher sensitivity configurations to more risk–averse operating points (e.g., $R = 3.0$), which substantially reduce false positives at the cost of lower recall.

These results quantify the false positive burden of standalone ML scoring and motivate the core contribution of this work: a deterministic, relevance–driven workflow that surrounds ML outputs and determines operational significance using mission context, temporal persistence, and cross–parameter conditions.

As illustrated through the qualitative scenario in Section VI, deterministic relevance logic can suppress non–actionable escalations (e.g., deviations explained by known operational states) and, when escalation is warranted, promote detections into structured operational events with traceability to triggering logic and associated checklist outcomes.

Further development focuses on end–to–end operational evaluation of the complete ML+workflow approach in mission–specific deployments, including quantifying reductions in false escalations and investigation time under real mission context procedures. We also plan to extend the approach to fleet–level analysis, where per–satellite anomaly scoring combined with aggregation across assets may enable detection of cross–platform patterns that are not observable at single–satellite scope. The broader insight is that operational trust in AI–driven monitoring is enabled not by model accuracy alone [12], but by system design that respects operator workflows, provides actionable context, and preserves human authority over final decisions.

References

[1] M. M. Rahman et al., "Prompt anomaly detection for small satellites in low-earth orbit constellations: A machine learning approach," in 38th Annu. Small Satellite Conf., 2024.
[2] L. Herrmann et al., "Unmasking overestimation: a re-evaluation of deep anomaly detection in spacecraft telemetry," CEAS Space J., vol. 16, no. 225–237, 2024.
[3] A. Fejjari et al., "A review of anomaly detection in spacecraft telemetry data," Appl. Sci., vol. 15, no. 10, p. 5653, 2025.
[4] K. Hundman et al., "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in Proc. 24th ACM SIGKDD Int. Conf., 2018, pp. 387–395.
[5] M. A. Obied et al., "Deep clustering-based anomaly detection and health monitoring for satellite telemetry," Big Data and Cognitive Computing, vol. 7, no. 1, 2023.
[6] A. Pilastre et al., "Anomaly detection in mixed telemetry data using a generative adversarial network," Aerospace, vol. 7, no. 12, 2020.
[7] D. Heinrich et al., "Human factors considerations in satellite operations human-computer interaction technologies: A review of current applications and theory," Int. J. of Managing Information Technology, vol. 13, pp. 23–43, 2021.
[8] C. Schmitt et al., "Applying machine learning to routine satellite ground segment operations by means of automated anomaly detection," in 9th Eur. Conf. for Aeronautics and Space Sciences, 2022.
[9] G. De Canio et al., "Development of an actionable ai roadmap for automating mission operations," in 17th Int. Conf. on Space Operations, 2023.
[10] Y. Pan et al., "Detecting anomalies in satellite telemetry data based on causal multivariate temporal convolutional network," in IEEE Conf. on Telemetry, 2022.
[11] K. Kotowski et al., "European space agency benchmark for anomaly detection in satellite telemetry," 2025.
[12] H. A. Tahir et al., "Toward explainable ai in spacecraft health monitoring: Comparative benchmarking of anomaly detection models and open-source datasets," IEEE Access, vol. 13, pp. 209 368–209 398, 2025.