ICDS: Integrated Cyberattack Detection for Satellites

Marco Chilese, Giannis Mouzenidis, Richard Mitev, Patrick Jauernig
SANCTUARY Systems GmbH, Germany

Daniel Fortún Sánchez, Martín Bárez Alonso
GMV, Germany

***Abstract*—Next-generation satellites are evolving into highly interconnected cyber-physical platforms, where heterogeneous subsystems, widespread adoption of Commercial Off-The-Shelf (COTS) components, and emerging multi-tenant payload paradigms fundamentally reshape the spacecraft threat landscape. These trends erode traditional trust assumptions and necessitate autonomous onboard cybersecurity mechanisms capable of detecting, localising, and mitigating attacks without reliance on continuous ground intervention.**

This paper presents a novel integrated Monitoring and Control (M&C) satellite subsystem that introduces cybersecurity anomaly detection and recovery services to sustainably secure the rapidly evolving landscape of satellite architectures. The subsystem comprises deep learning-driven system bus monitoring and static rule enforcement, onboard computer (OBC) log monitoring, and recovery strategies for other subsystems. The deep learning-based monitoring module continuously observes system bus communications without perturbing nominal operations, enabling real-time situational awareness at the system level. The underlying model is extremely small, based on recurrent autoencoders with four layers and 8,465 trainable parameters, yet it precisely detects subtle structural and temporal deviations indicative of cyberattacks. Lightweight rule-based detectors and machine learning-based log analysis complement the learning-based bus monitoring, providing fast baseline detection and robustness against model blind spots. When an anomaly is detected, the framework performs automated root cause analysis to identify responsible communication, enabling fine-grained isolation and recovery of compromised subsystems. Two recovery approaches are presented, based on subsystem software redundancy, and a lightweight method leveraging the SpaceWire Remote Memory Access Protocol.

Extensive experimental evaluation demonstrates strong detection performance and operational feasibility. The system bus anomaly detector achieves a 97.18% true negative rate with low false positive rates on benign traffic, and F1 scores of 89.88% for denial-of-service attacks, 92.70% for full-payload interference, and 82.95% for partial frame corruption. Our breadboard prototype yields an inference latency of between 4ms to 240 ms per analysis window depending on the analysis stage, validating real-time onboard applicability under realistic resource constraints.

***Index Terms*—Anomaly detection, Small satellites, Satellites, Space technology, Machine learning, Embedded systems, Computer security, Security.**

I. Introduction

The proliferation of satellites, particularly those adopting Commercially Off-the-Shelf (COTS) components, is accelerating rapidly. However, this modernisation comes at the cost of an expanded attack surface: hardware and software originally designed for terrestrial environments often lack the resilience required against space-specific threats. Beyond individual components, satellites now function as highly complex cyber-physical systems, where heterogeneous subsystems must interoperate within distributed embedded architectures, fundamentally reshaping the spacecraft threat landscape.

Emerging trends such as multi-tenant payload configurations exacerbate these concerns. Thales' demonstration during ESA's OPS-SAT mission [1], where researchers escaped the payload bus via a Java-based experiment, vividly illustrates the capability to compromise satellite operations from within the payload environment. Critically, once an adversary gains a foothold in payload systems, they may pivot laterally into the satellite bus, endangering fundamental spacecraft control. Additionally, recent commercial innovations, exemplified by Amazon's AWS Ground Station as a service [2], have democratised access to satellite ground infrastructure. While facilitating easier mission operations, these platforms introduce new cyber risks where insecure Telemetry, Tracking, and Command (TT&C) links can compromise mission integrity even when transport-layer protections are in place. These developments underscore a shift in threat models: ground-to-space links are no longer the sole concern—autonomous onboard security is now imperative.

While ground link security benefits from established encryption and authentication mechanisms, onboard satellite defences are less extensively studied. The challenge is compounded by the resource constraints inherent to space-grade embedded systems, which often preclude the deployment of computationally heavy deep learning models or complex cryptographic protocols. Furthermore, existing research predominantly focuses on detection using telemetry [3], [4], [5], [6], [7], [8], [9] or sensor data [10], [11], [12], [13], often neglecting the critical need for immediate, automated recovery mechanisms that can contain threats before they propagate across the system. Most current frameworks rely heavily on ground intervention [14], [15], [16] for mitigation, introducing latency that is incompatible with rapid threat containment in orbit.

Consequently, robust, autonomous on-satellite security mechanisms are required. In response to these challenges, this paper presents ICDS, a novel Integrated Anomaly Detection and Recovery System tailored for satellite platforms. Unlike

traditional approaches that treat detection and recovery as separate phases, ICDS integrates a lightweight, multi-layered monitoring stack with an adaptive control module. The system employs a hybrid detection strategy combining rule-based analysis, log inspection, and deep learning-driven bus monitoring to identify anomalies in real-time. Crucially, upon detection, the Control Module orchestrates recovery actions based on the severity of the threat, ranging from autonomous isolation and partition switching (A/B redundancy) to safe-mode transitions, thereby ensuring resilience without necessary ground intervention.

In this work, we make the following key contributions:

- We propose a novel, purely on-board anomaly detection and recovery framework designed to identify and mitigate cyber attacks targeting satellite systems autonomously.
- We introduce a tiered recovery mechanism that dynamically selects mitigation strategies (isolation, partition switching, or safe mode) based on the detected anomaly's severity level, balancing autonomy with necessary human oversight.
- We implement a multi-level detection approach incorporating bus metadata analysis, payload inspection, and onboard computer (OBC) log monitoring to provide comprehensive situational awareness.
- We demonstrate the operational feasibility of our solution through extensive experimentation, achieving high detection accuracy (F1-scores up to 92.70%) while maintaining low inference latency and minimal memory footprint suitable for resource-constrained satellite subsystems.

II. System Model

This section outlines the system model, introducing fundamental satellite concepts relevant to security analysis. Note that this model is intended to establish a shared conceptual framework; the developed approach remains agnostic to the specific subsystem configuration or architecture.

A typical satellite architecture comprises two primary components, as illustrated in Figure 1: the satellite bus and the payload. The satellite bus encompasses all subsystems required to operate and maintain satellite functions, including power management, thermal control, attitude determination and control systems (ADCS), command and data handling subsystems (CDHS), as well as telemetry, tracking, and command (TT&C). Vulnerabilities present within these subsystems pose a particularly severe threat due to their pivotal role in satellite operations; an attacker gaining access to any subsystem within the satellite bus can potentially exert comprehensive control over the satellite's core operations. Consequently, from a security standpoint, the satellite bus constitutes a critical attack vector, influencing the satellite's basic functionality and its communication interfaces with ground control systems and other spacecraft.

In contrast, the satellite payload contains equipment specifically designed to execute the satellite's primary mission objectives. For instance, Earth observation satellites may include high-resolution imaging sensors, whereas communication satellites utilize advanced radio-frequency equipment. Due to their specialized nature and critical role in mission success, payload components represent attractive targets for attackers aiming to disrupt or subvert mission goals. It is important to note that mission-specific payloads may contain subsystems functionally equivalent to those found on the satellite bus, such as dedicated TT&C units, necessitating their inclusion in monitoring strategies.

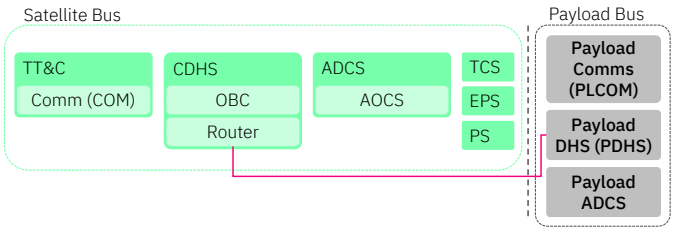

Fig. 1. Satellite system model illustrating satellite bus and payload connectivity.

However, the interconnectivity between the satellite bus and payload significantly enlarges the attack surface. Typically, these elements communicate through standardized communication buses, such as Controller Area Network (CAN), enabling efficient data exchange and functional coordination. This connectivity introduces a bidirectional threat vector: attackers compromising the payload may escalate their privileges by leveraging existing communication links to infiltrate the satellite bus; similarly, a successful breach of the satellite bus can enable an attacker to interfere with or disable the payload. Based on this architecture, our identified adversary models focus on three primary attack vectors:

- Lateral Movement via System Bus: Adversaries exploiting the CAN bus to pivot from a compromised payload (or less secured subsystem) into the critical satellite bus to gain root-level control.
- Payload Disruption: Direct attacks aimed at payload subsystems to manipulate mission data or disable operational capabilities without necessarily achieving full bus control.
- Interface Exploitation: Attacks targeting the communication protocols and interfaces between the bus and payload to inject malicious commands or intercept sensitive telemetry.

Thus, securing communication interfaces and protocols between the satellite bus and payload emerges as a critical measure for mitigating unauthorized access and limiting attack escalation risks.

III. Design

The Integrated Cyberattack Detection System for Satellites (ICDS) is a novel security framework designed to autonomously identify and mitigate cyber-physical threats in satellite architectures. Unlike traditional approaches, which either rely on ground-based intervention and mostly focus solely on detection[14], [15], [16], ICDS combines a lightweight,

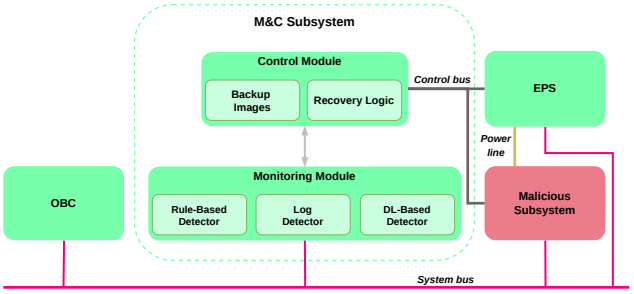

Fig. 2. High level overview of the Monitoring and Control Modules.

multi-layered monitoring stack with an adaptive control module capable of executing tiered recovery actions. This integrated design enables rapid threat containment and system restoration while respecting the stringent resource constraints of space-grade embedded systems.

As illustrated in Figure 2, the ICDS architecture is centred around an independent security subsystem directly connected to the satellite's main system bus. This design is aligned with the SAVOIR[1] Monitoring and Control (M&C) reference architecture, in which the proposed solution operates as a distinct subsystem capable of functioning in both active and passive modes, either actively requesting data or passively monitoring bus communications. The framework is composed of two primary components: the Monitoring Module, responsible for threat detection, and the Control Module, responsible for executing recovery actions.

A. Monitoring Module

The Monitoring Module is responsible for continuous assessment of the spacecraft's information flow to ensure situational awareness. To achieve robust detection capabilities, the module employs a multi-layered approach utilising three types of detectors, including rule-based analysis, ML-powered log analysis, and ML-powered packet analysis leveraging metadata and payload information.

First, a rule-based detector is implemented as first stage (Stage 1) of the monitoring process. This component leverages expert knowledge and field experience to define a set of deterministic rules where any violation signals potential system compromise. Usually, these variables are already known due to safety requirements. Instead of relying solely on learned patterns, this approach utilises statistical analysis, cumulative message counters, and predefined thresholds to detect irregularities such as deviations from expected message patterns, traffic volume anomalies (e.g., flooding), or violations of operational limits trained on benign traffic data (cf. Section V). By processing these parameters in real-time, the rule-based detector identifies coarse-grain anomalies almost instantaneously without requiring resource-intensive analysis, effectively serving as a lightweight first line of defence against known threat signatures before engaging more detection mechanisms. Given the deterministic and low false positive assessments nature of this detector, it issues high severity alerts.

Complementing this, an ML-powered Log Detector continuously monitors system logs generated by the On-Board Computer (OBC) and other critical subsystems to identify operational irregularities that may not be immediately visible in bus traffic. Unlike the rule-based layer which focuses on network metrics, the Log Detector analyses textual logs, structured events, and execution traces to uncover subtle deviations in system behaviour, such as anomalous command sequences or unauthorised file system manipulations. This component utilises a light machine learning model, specifically a One-Class SVM trained on simulated F Prime instance logs, to distinguish between normal operational patterns and potential compromise indicators. By correlating these log-based findings with bus traffic anomalies, the Log Detector provides a secondary verification layer that enhances situational awareness and significantly reduces the overall False Positive Rate in complex attack scenarios. Differently from the rule-based detector, the log detector issues low severity alerts.

For deeper inspection, implemented as Stage 2 of the monitoring process, the Monitoring Module incorporates a unified detection mechanism that jointly analyses both metadata and payload content extracted directly from the satellite's System Bus traffic using a lightweight deep learning (DL) model, represented in Figure 3. Rather than employing separate detectors for each data type or relying solely on high-level telemetry, the proposed approach fuses these information streams into a single data representation. This enables the model to exploit a broader contextual view and capture correlations between metadata and payload, resulting in a more comprehensive assessment of communication patterns. Specifically, the DL model is based on a Recurrent Autoencoder (RAE) architecture with Gated Recurrent Unit (GRU) layers. It processes sliding windows of 64 consecutive CAN messages, allowing it to capture temporal dependencies and structural characteristics inherent in bus communications. By jointly ingesting metadata (e.g., timing information and message identifiers) and raw payload content, the model enhances its sensitivity to subtle, stealthy, or coordinated anomalies that may be overlooked when these features are analysed in isolation. The RAE is trained in an unsupervised manner using only benign data, with anomaly detection triggered when the reconstruction error exceeds a calibrated threshold. This design ensures high sensitivity to complex cyber-physical attacks while maintaining minimal communication and computational overhead.

Crucially, upon detecting an anomalous window, the framework performs automated root cause analysis to identify the specific responsible subsystem. By leveraging knowledge of the underlying communication graph, or in scenarios that leverage directed communication protocols such as libCSP [17], the system traces back the most likely events within the reconstructed window that contributed to the abnormal anomaly score. This diagnostic capability (i.e., root-cause analysis) pinpoints the source of the compromise, such as a specific transmitter ID or a particular subsystem interaction,

[1]https://savoir.estec.esa.int/

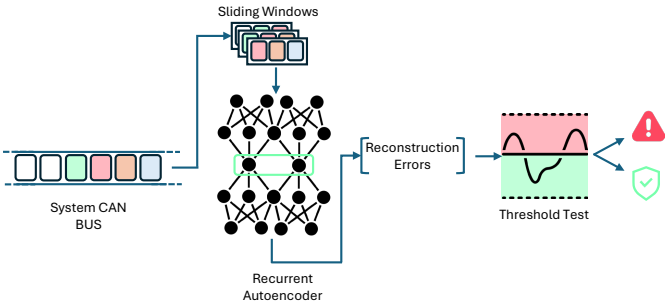

Fig. 3. High level overview of the Deep Learning-Baed detector.

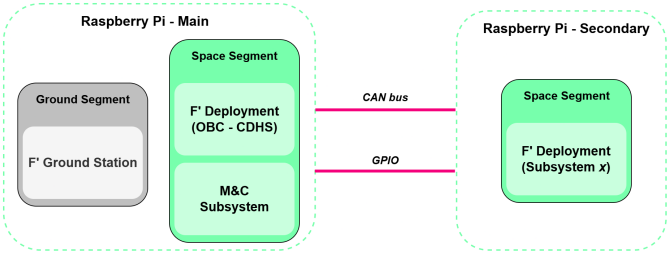

Fig. 4. High level overview of the proof-of-concept.

which is then immediately reported to the Control Module. This precise attribution enables the execution of fine-grained recovery actions, ensuring that mitigation strategies target only the affected elements rather than imposing broad, disruptive resets.

In general, when an anomaly is detected, the system assigns a severity level alongside the anomaly information to facilitate appropriate response actions. Specifically, detections from the rule-based detector are immediately flagged with a high severity level, reflecting its deterministic and non-probabilistic nature which indicates a confirmed violation of established rules. In contrast, anomalies identified by the log and CAN packets detects are assigned with a lower severity level, as these approaches rely on probabilistic assessments of deviations from learned normal behaviour. These distinct severity levels serve as critical inputs for the Control Module, enabling it to orchestrate recovery steps that are proportionate to the confidence and urgency of the detected threat.

B. Control Module

Complementing the detection capabilities of the monitoring layer, the Control Module orchestrates recovery mechanisms tailored to the severity level of the detected anomaly. For minor faults or low-severity alerts, the framework executes autonomous module-level responses, such as resetting or shutting down specific components, to restore functionality without external intervention. In more severe scenarios involving persistent compromise or complex adversarial attacks, the system escalates to partition-level recovery strategies. This includes switching between redundant A/B partitions, effectively rebooting the affected environment from a known clean state while isolating the compromised partition for later reflashing using a backup image through, e.g., SpaceWire Remote Memory Access Protocol (RMAP). For the most critical incidents, this tiered approach may transition the satellite into a safe mode or initiate extensive procedures requiring ground station coordination to ensure a secure return to a trusted baseline. This stepped strategy ensures that responses are proportionate to the threat, balancing rapid autonomy with necessary oversight for high-stakes security events. Note that due to the highly mission-specific design of satellite architectures, these recovery methods are rather a baseline of possible approaches. For instance, with redundant subsystems within an ADHA (Advanced Data Handling Architecture)-based architecture [18], a compromised subsystem can also be powered off and its backup system can take over.

IV. Implementation

The proposed ICDS prototype constitutes a comprehensive, layered security architecture specifically designed for resource-constrained SmallSats and CubeSats. In contrast to traditional monolithic safety systems, the framework operates as an independent security subsystem running in parallel with the primary mission computer. It interfaces directly with the satellite backbone via the Controller Area Network (CAN) bus, leveraging its broadcast nature to passively monitor inter-component communications without requiring intrusive modifications to legacy firmware.

The prototype was implemented as a distributed embedded testbed composed of two Raspberry Pi nodes, represented in Figure 4 emulating a satellite architecture. The use of Raspberry Pi platforms enables rapid prototyping while preserving architectural fidelity with space-grade distributed systems. Specifically, one Raspberry Pi acts as the On-Board Computer (OBC), running the F Prime (F ') flight software stack[2], while the second node emulates payload and subsystem components interconnected via a physical CAN bus. The proposed ICDS security subsystem is deployed on the main node (i.e., the one simulating the OBC) and is connected to the same CAN bus, enabling passive monitoring of all inter-subsystem communications. Furthermore, the secondary node also implements the ground segment, generating Telecommands (TCs) that are injected into the system through the OBC, thereby reproducing realistic operational and adversarial scenarios.

The core flight software stack is built upon F', a NASA-developed, object-oriented framework widely adopted in CubeSat missions due to its modularity and real-time capabilities. The security subsystem integrates with the F' command-and-telemetry infrastructure, enabling the interception and inspection of CAN frames exchanged with critical payloads. To evaluate the architecture under realistic stress conditions, the ground segment is implemented as a dedicated simulation environment hosted on the secondary Raspberry Pi, which acts as a ground station emulator capable of issuing telecommands to reproduce both nominal and anomalous operational traffic.

[2]https://fprime.jpl.nasa.gov/

The software stack follows a modular design that clearly separates detection from recovery functionalities. The Monitoring Module is composed of three parallel layers. The first layer, a Rule-Based Engine, employs deterministic state machines and statistical thresholds (e.g., cumulative counters) to promptly detect conditions such as traffic flooding or unauthorised identifier sequences, without relying on complex models. The second layer, the Log Analyzer, aggregates On-Board Computer (OBC) logs and leverages scikit-learn[3] to implement a One-Class SVM, enabling the identification of latent anomalies such as irregular command patterns or file system manipulations through the analysis of both textual and structured log data. The third layer, the Deep Learning Agent, performs packet-level analysis using a GRU-based recurrent autoencoder implemented in TensorFlow[4]. By operating on sliding windows of 64 messages, it jointly processes metadata and raw payload bytes to capture complex temporal dependencies.

The Control Module acts as the central orchestrator of recovery actions, mapping severity levels generated by the detection layers to a structured escalation strategy. Low-severity alerts, typically produced by probabilistic components such as the Log Analyzer and the Deep Learning Agent, trigger autonomous, localised responses, including component isolation or targeted system resets. Conversely, high-severity events initiate A/B partition switching, allowing compromised subsystems to be rebooted from trusted images. In the proof-of-concept implementation, this mechanism is realised via GPIO control lines that emulate power cycling and instruct the bootloader to switch to a backup partition. Additionally, critical failures trigger a transition to safe mode, suspending non-essential payload operations to enable ground intervention.

V. Evaluation

This section presents the comprehensive evaluation of ICDS. We detail the experimental setup, including the adaptation of datasets to a realistic space-domain simulation, and describe the methodology used to validate the full security framework. The evaluation covers the performance of the multi-layered monitoring stack, comprising rule-based, log, and deep learning detectors, and assesses the efficacy of the Control Module in orchestrating recovery actions based on threat severity.

Dataset and Experimental Setup. Our evaluation relies on the publicly available OTIDS CAN dataset [19], containing 2.37M recorded messages. To adapt this automotive data for the space domain, we replayed the traffic through a physical CAN connection between two Raspberry Pi 4 units using the F' (F Prime) flight software (cf. Section IV). This setup mimics spacecraft message timing while introducing a processing layer that replicates onboard behaviour. For the machine learning component, we extracted lightweight features including inter-arrival time, payload entropy, unique byte counts, and individual byte values. Additionally, to evaluate the rule-based and log detectors, we simulated specific system logs and metadata patterns typical of satellite bus operations.

To assess the robustness of the detection framework, we designed three attack scenarios:

1) **Denial-of-Service (DoS):** The adversary floods the channel with random messages to delay legitimate traffic. We injected 32 high-frequency random messages into 10 benign segments.
2) **Full Interference (8B):** The attacker replaces the entire 8-byte payload with random noise while preserving CAN IDs and timing.
3) **Partial Interference (4B):** A stealthier attack where only four randomly selected bytes in the payload are corrupted.

Detection Performance. The evaluation confirms the efficacy of the multi-detector architecture. The static threshold for the RAE was set at the mean reconstruction error plus two standard deviations. On benign data, the RAE achieved a True Negative Rate (TNR) of 97.18% and a False Positive Rate (FPR) of 2.81%.

Table I summarizes the results across all attack types. The model demonstrates strong resilience, achieving an F1-score of 89.88% for DoS attacks and 92.70% for full interference. Notably, even against the challenging partial interference (4B) scenario, the system maintains an F1-score of 82.95% with perfect precision (100.0%), indicating a conservative bias that prioritizes minimizing false alarms.

TABLE I
Evaluation results for DoS and interference attacks. Results are averaged over 10 experiments.

Attack Type	Accuracy	Precision	Recall	F1-Score
DoS	93.49%	91.70%	90.95%	89.88%
Interference 8B	92.21%	86.39%	100.0%	92.70%
Interference 4B	85.60%	100.0%	70.87%	82.95%

Beyond the deep learning model, the integration of rule-based and log detectors significantly reduced detection latency for known threat signatures. The rule-based detector successfully identified immediate Denial-of-Service (DoS) flooding patterns within 1 ms. Complementing this, the Log Detector was evaluated using a dataset of 300 event and command logs from a simulated onboard instance. It employed a One-Class SVM to identify anomalous command sequences correlating with data and file system manipulations, contributing to a secondary verification layer that reduced the overall False Positive Rate in complex scenarios. Performance benchmarks indicate that parsed log entries consume minimal memory (e.g., $\sim$ 0.94 KB per event entry), with average 38 ms processing time recorded during evaluation. However, it is important to note that the current detection accuracy of approximately 75% serves as an initial proof-of-concept driven by data scarcity inherent to the simulation environment. These results demonstrate feasibility rather than final optimisation; the specific algorithmic implementation is modular and can

[3]https://scikit-learn.org/
[4]https://www.tensorflow.org/

be readily replaced or enhanced with advanced methodologies proposed in existing literature to further improve robustness, accuracy, and reduce false positives as more realistic datasets become available [20], [21], [22], [23].

Control Module and Recovery Evaluation. We further evaluated the Control Module's ability to orchestrate recovery actions based on severity levels. In low-severity tests (e.g., isolated DoS bursts), the module executed full autonomous resets at the subsystem level, restarting the affected software processes and clearing volatile states without ground intervention. This approach restored normal bus traffic and complete reachability within 32.20 seconds. Unlike high-severity responses, these low-severity events do not require altering the persistent storage state.

For critical simulations involving persistent payload corruption where the local file system or binary integrity was compromised, the system escalated to the A/B partition switching mechanism as the event is detected by both the rule-based detector (i.e., payload consistency check) and by the deep learning based detector. Here, the Control Module detected the deep-seated compromise, isolated the current running partition, and rebooted the entire platform from a redundant, verified backup image in under 20.06 seconds begin operative shortly afterwards. This distinction ensures that while minor faults are corrected via lightweight reinstatement, severe corruptions are mitigated by restoring a known-clean base state, effectively halting attack propagation and validating the tiered response strategy. Additionally, the transition to safe mode was correctly triggered during simulated critical failures, demonstrating the system's capacity to balance autonomy with strict safety protocols.

Deployment Feasibility. Finally, we assessed the resource requirements for on-board deployment. Benchmarks were conducted on a Raspberry Pi 3 Model B+ (Cortex-A53 @1.4GHz, 1GB RAM), which uses the same CPU cores as space-grade embedded systems such as the Genesys ZU Zynq Ultrascale+ MPSoC. The evaluation covered both the rule-based and machine learning detectors. The rule-based detector demonstrated high efficiency, with a total median processing time of 40.24 ms per 300 CAN frames and a minimal memory footprint of approximately 9.08 KB per processed event. Complementing this, the log detector achieved an inference time of 3.96 ms per sample (log event) with a memory footprint of approximately 118.6 KB. For the more computationally intensive ML-based detector (RAE), the total memory footprint was measured at approximately 240.63 KB, primarily driven by the event window and preprocessed data structures. In terms of latency, the ML detector achieved a median total processing time of 240.34 ms, with the detection function itself consuming the majority of this time (approx. 205.48 ms). These metrics confirm that the combined Monitoring and Control modules can operate in real-time within the strict power and memory constraints typical of modern satellites.

VI. Related Work

Existing research on anomaly detection in spacecraft systems predominantly focuses on data derived from three principal sources: telemetry, transponder frequency, and sensor data related to physical faults.

Telemetry-based Detection. Anomaly detection using spacecraft telemetry data has extensively employed a wide range of machine learning (ML) methodologies. In particular, recent works increasingly leverage deep learning architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), generative adversarial networks (GANs), and transformers. Baireddy et al. [24] propose anomaly detection via transfer learning, fine-tuning a generalized ML model across telemetry channels. Similarly, Yu et al. [3] employ a GAN-based approach combining CNNs and RNNs to quantify anomaly severity. Wang et al. [4] integrate nearest neighbour algorithms with Long Short-Term Memory (LSTM) networks and Gaussian models to distinguish false from true positives. Zeng et al. [5] and Liu et al. [6] further enhance predictive capabilities through attention mechanisms and temporal convolutional networks (TCNs), respectively. Meanwhile, Xu et al. [7] propose a hybrid statistical and Support Vector Machine (SVM) approach, whereas Cuellar et al. [8] introduce explainable AI techniques to complement LSTM-based anomaly detection. More recent advancements include diffusion probabilistic models [9] and transformer architectures specifically tailored for satellite telemetry [25]. Despite these advances, telemetry-based methodologies exhibit several limitations. A common assumption is that telemetry channels share relatively uniform data distributions, which may not hold in heterogeneous satellite systems and can lead to detection inaccuracies. Moreover, the inherent complexity and computational demands of deep learning models—particularly RNNs, GANs, and transformers—raise significant concerns regarding scalability and deployment feasibility on resource-constrained onboard platforms. In addition, some widely used benchmark datasets and associated methodologies introduce simplifying assumptions that may not reflect realistic adversarial behaviour. For instance, the OTIDS dataset proposed by Lee et al. [19] models Denial-of-Service (DoS) attacks using empty payload frames as a distinguishing characteristic. While suitable for initial evaluation, this assumption renders such attacks trivial to detect based solely on payload content or size. In practice, adversaries can generate high-rate traffic with syntactically valid payloads, thereby evading simple payload-based or volume-based detection mechanisms. This highlights the need for detection approaches that capture richer temporal and structural characteristics of communication patterns, as pursued in this work.

Transponder Frequencies. Complementary approaches investigate transponder frequencies for anomaly detection, with the advantage of ground-based processing that conserves spacecraft resources. Gunn et al. [26] employ unsupervised LSTM-based RNN models for outlier detection in high-dimensional transponder frequency data. However, this method

does not facilitate anomaly labelling, thus providing limited insights into anomaly nature or intent.

Physical Fault Detection. Anomaly detection based on sensor data, particularly targeting physical faults, represents another significant research direction. Codetta et al. [10] develop a fault detection and autonomous recovery system using dynamic Bayesian networks. Naik et al. [11] demonstrate the effectiveness of traditional ML approaches such as random forests in anticipating physical anomalies like increased friction in reaction wheels. Additionally, Stottler et al. [12] describe onboard autonomous diagnosis systems designed to identify and rectify physical faults. Wiatrek et al. [13] propose a deterministic graph-based method leveraging sensor meta-data to achieve low-latency anomaly detection. Although these studies confirm the suitability of traditional and graph-based ML methods for detecting and identifying sensor-related faults, they can not trivially be extended to cover cybersecurity events. However, they could serve as an additional input in future anomaly detection works.

Recovery and Mitigation Strategies. While detection is well-represented, research specifically addressing automated recovery and mitigation in response to cyber-physical attacks remains less explored compared to physical fault handling [27]. Existing literature often treats fault recovery in CPSs as a secondary step [28] or, in the case of spacecrafts, relies heavily on ground intervention, which introduces latency incompatible with rapid threat containment [29]. Many current approaches focus on specific countermeasures or decision policies rather than a unified severity-aware response hierarchy [30].

For instance, some frameworks propose switching to redundant partitions such as the one presented by Paridari et al. [31] an attack-resilient control framework that couples anomaly detection with controller reconfiguration to maintain closed-loop stability under attack, or, Chaves et al. [32] who strengthen redundancy-based resilience in industrial control systems through active defence, aiming to prevent common-cause compromise of primary and backup PLCs. Rushby [33], by contrast, addresses partitioning as a foundational isolation mechanism in avionics, ensuring strong fault containment between critical subsystems.

Other propose entering safe modes, such as Thummala et al. [29] who introduce a mission-aware cyber safe mode for spacecraft that autonomously contains intrusions while preserving essential control functions.

Existing work is fragmented across detection, controller reconfiguration, and resilience mechanisms, and comparatively few architectures integrate these into a single runtime orchestration framework.

ICDS aims to bridge this gap by integrating lightweight, multi-layered detection (including rule-based and log analysis) with an adaptive control module capable of executing isolation, partition switching, and safe-mode transitions autonomously. This approach contrasts with traditional methods that either rely on computationally heavy models unsuitable for edge deployment or fail to implement immediate, proportional recovery actions without external oversight.

VII. Conclusion

In this paper, we presented ICDS, an integrated anomaly detection and recovery system, a comprehensive security framework designed to autonomously identify and mitigate cyberattacks on next-generation satellite platforms. Unlike traditional approaches that rely heavily on ground intervention or focus solely on detection, ICDS integrates a lightweight, multi-layered monitoring stack with an adaptive control module capable of executing tiered recovery actions. The system operates entirely onboard, utilising a hybrid detection strategy that combines rule-based analysis, log inspection, and deep learning-driven bus monitoring to provide real-time situational awareness without requiring modifications to existing subsystems.

Our evaluation demonstrated the efficacy of this integrated approach across multiple attack vectors, including Denial-of-Service (DoS) and payload interference attacks. The Recurrent Autoencoder (RAE) detector achieved high accuracy (F1-scores up to 92.70%) while maintaining low latency and a minimal memory footprint suitable for resource-constrained embedded systems. Furthermore, we validated the Control Module's ability to orchestrate recovery strategies based on threat severity. This includes immediate autonomous isolation for minor anomalies, A/B partition switching to restore functionality from a known clean state for persistent compromises, and safe-mode transitions for critical incidents requiring ground oversight. These results confirm that ICDS can effectively contain threats and restore operations within strict time constraints, significantly enhancing satellite resilience.

Future work will focus on refining the recovery mechanisms and expanding the system's adaptability. We plan to further integrate advanced SpaceWire Remote Memory Access Protocol (RMAP) capabilities to facilitate secure key rotation and filesystem updates following a detected compromise, ensuring long-term immunity against persistent threats. Additionally, we aim to enhance the log-based detector to support more complex query patterns across diverse subsystem architectures and explore the application of transfer learning techniques to improve the RAE's generalisation capabilities across different satellite missions. Finally, ongoing research will address the optimisation of the human-in-the-loop interface, ensuring seamless coordination between autonomous onboard responses and ground station verification during high-severity events.

References

[1] Thales Group, "Thales seizes control of ESA demonstration satellite in first cybersecurity exercise of its kind," Press release, Thales Group, Apr. 2023, accessed: 2025-07-07. [Online]. Available: https://www.thalesgroup.com/en/worldwide/security/press_release/thales-seizes-control-esa-demonstration-satellite-first

[2] Amazon Web Services, Inc., "AWS Ground Station," https://aws.amazon.com/ground-station/, 2025, accessed: 2025-07-07.

[3] J. Yu, T. Zhang, W. Liu, X. Li, K. Sun, J. Wang, and X. Qiu, "Telemetry data-based spacecraft anomaly detection with spatial–temporal generative adversarial networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[4] Y. Wang, Z. Sun, T. Li, J. Liu, P. Zhang, and Y. Xu, "A deep learning anomaly detection framework for satellite telemetry with fake anomalies," *International Journal of Aerospace Engineering*, vol. 2022, no. 1, p. 1676933, 2022.

[5] Z. Zeng, X. Li, J. Chen, R. Wang, and B. Guo, "Satellite telemetry data anomaly detection using causal network and feature-attention-based lstm," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–21, 2022.

[6] L. Liu, L. Tian, Z. Kang, and T. Wan, "Spacecraft anomaly detection with attention temporal convolution networks," *Neural Computing and Applications*, vol. 35, no. 13, pp. 9753–9761, 2023.

[7] Z. Xu, Z. Cheng, and B. Guo, "A hybrid data-driven framework for satellite telemetry data anomaly detection," *Acta Astronautica*, vol. 205, pp. 281–294, 2023.

[8] S. Cuéllar, J. Fernández, E. García, and M. López, "Explainable anomaly detection in spacecraft telemetry," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108083, 2024.

[9] J. Sui, H. Zhao, Y. Li, and C. Liu, "Anomaly detection for telemetry time series using a denoising diffusion probabilistic model," *IEEE Sensors Journal*, 2024.

[10] D. Codetta-Raiteri and L. Portinale, "Dynamic bayesian networks for fault detection, identification, and recovery in autonomous spacecraft," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 13–24, 2014.

[11] K. Naik, A. Holmgren, and J. Kenworthy, "Using machine learning to automatically detect anomalous spacecraft behavior from telemetry data," in *IEEE Aerospace Conference*. IEEE, 2020.

[12] D. Stottler, X. Zhang, R. Kumar, and S. Evans, "On-board, autonomous, hybrid spacecraft subsystem fault and anomaly detection, diagnosis, and recovery," in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2020.

[13] N. Wiatrek, L. Smith, H. Chen, and A. Patel, "Advancing spacecraft security through anomaly detection," in *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. IEEE, 2024.

[14] L. Schlag, C. Schefels, and K. Helmsauer, "Applying machine learning to routine satellite ground segment operations by means of automated anomaly detection," *Aerospace Europe Conference 2023*, 2023.

[15] O. Driouch, S. Bah, and Z. Guennoun, "Distributed intrusion detection system for cubesats, based on deep learning packets classification model," in *Proceedings of the 2024 Security for Space Systems (3S)*. Noordwijk, Netherlands: IEEE & ESA, May 2024. [Online]. Available: https://indico.esa.int/event/528/attachments/5988/10192/Distributed_intrusion_detection_system_for_CubeSats_based_on_deep_learning_packets_classification_model.pdf

[16] D. S. Reddy, H. Saxena, and S. S. Solanki, "Satellite attitude anomaly detection in ground data processing," *Space: Science & Technology*, vol. 6, p. 0328, 2026.

[17] libcsp contributors, "The Cubesat Space Protocol (libcsp) Documentation," https://libcsp.github.io/libcsp/, 2025.

[18] European Space Agency, "Advanced data handling architecture (adha)," https://technology.esa.int/page/advanced-data-handling-architecture-adha, 2026, accessed: 2026-04-13.

[19] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2017, pp. 57–5709.

[20] H. Guo, S. Yuan, and X. Wu, "Logbert: Log anomaly detection via bert," in *2021 international joint conference on neural networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[21] M. Catillo, A. Pecchia, and U. Villano, "Autolog: Anomaly detection by deep autoencoding of system logs," *Expert Systems with Applications*, vol. 191, p. 116263, 2022.

[22] J. Wang, C. Zhao, S. He, Y. Gu *et al.*, "Loguad: Log unsupervised anomaly detection based on word2vec." *Computer Systems Science & Engineering*, vol. 41, no. 3, 2022.

[23] C. Almodovar, F. Sabrina, S. Karimi, and S. Azad, "Logfit: Log anomaly detection using fine-tuned language models," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1715–1723, 2024.

[24] S. Baireddy, S. R. Desai, J. L. Mathieson, R. H. Foster, M. W. Chan, M. L. Comer, and E. J. Delp, "Spacecraft time-series anomaly detection using transfer learning," in *CVPR Workshops*, 2021, pp. 1951–1960.

[25] H. Zhao, J. Sui, Y. Wang, and C. Li, "Satellite early anomaly detection using an advanced transformer architecture for non-stationary telemetry data," *IEEE Transactions on Consumer Electronics*, 2024.

[26] L. Gunn, M. Hughes, R. Patel, and E. Stewart, "Anomaly detection in satellite communications systems using lstm networks," in *Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2018.

[27] L. F. Cómbita, J. Giraldo, A. A. Cárdenas, and N. Quijano, "Response and reconfiguration of cyber-physical control systems: A survey," in *2015 IEEE 2nd Colombian conference on automatic control (CCAC)*. IEEE, 2015, pp. 1–6.

[28] P. Lu, L. Zhang, M. Liu, K. Sridhar, O. Sokolsky, F. Kong, and I. Lee, "Recovery from adversarial attacks in cyber-physical systems: Shallow, deep, and exploratory works," *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–31, 2024.

[29] R. K. Thummala, G. J. Falco, B. E. Schake, D. L. Pollock, D. J. Melander, C. P. Banh, B. Pendleton, and R. A. Procell, "Mission aware cyber safe mode for spacecraft," Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2025.

[30] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–29, 2014.

[31] K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 113–128, 2017.

[32] A. Chaves, M. Rice, S. Dunlap, and J. Pecarina, "Improving the cyber resilience of industrial control systems," *International journal of critical infrastructure protection*, vol. 17, pp. 30–48, 2017.

[33] J. Rushby, "Partitioning in avionics architectures: Requirements, mechanisms, and assurance," NASA, Tech. Rep., 1999.