UDC 629.78
DOI: 10.15587/1729-4061.2026.358310

DEVELOPMENT OF A HYBRID ONBOARD DIAGNOSTIC ARCHITECTURE WITH LIGHTWEIGHT MACHINE LEARNING FOR RESOURCE-CONSTRAINED CUBESAT SYSTEMS

Ainur Kuttybayeva
PhD*, Associate Professor**
ORCID: https://orcid.org/0000-0001-7281-3690
Samal Zhamalova
Master of Pedagogical Sciences
Pedagogical Institute
Astana International University
Qabanbay Batyr ave., 8, Astana, Republic of Kazakhstan, 020000
ORCID: https://orcid.org/0009-0000-7310-9398
Anargul Boranbayeva
Corresponding author
Doctoral Student*, Senior Lecture***
E-mail: boranbayeva.anargul.talgatkyzy@gmail.com
ORCID: https://orcid.org/0000-0003-2221-9604
Zhansaya Myrzayeva
Master of Technical Sciences, Lecture
Pedagogical Institute
Astana International University
Qabanbay Batyr ave., 8, Astana, Republic of Kazakhstan, 020000
ORCID: https://orcid.org/0009-0000-7375-7985
Gulnar Imasheva
Doctor of Technical Sciences, Professor
Department of Logistics
School of Transport Engineering and Logistics***
ORCID: https://orcid.org/0000-0003-3604-3320
Zhanat Kaskatayev
Candidate of Technical Sciences, Associate Professor
Department of Logistics
School of Transport Engineering and Logistics***
ORCID: https://orcid.org/0009-0005-9519-8885
Yersain Chinibayev
PhD, Associate Professor
Department of Software Engineering***
ORCID: https://orcid.org/0009-0009-8985-5892
Nurzhamal Ospanova
PhD, Associate Professor
Department of Radio Engineering, Electronics and Telecommunications
International IT University
Manasa str., 34/1, Almaty, Republic of Kazakhstan, 050040
ORCID: https://orcid.org/0000-0003-1170-4794
Mukhit Abdullayev
Candidate of Technical Sciences**
ORCID: https://orcid.org/0009-0002-3349-1881
Kalmukhamed Tazhen
Master Student**
ORCID: https://orcid.org/0009-0004-5890-5328
*Department of Mechanics and Machine Science
Institute of Mechanics and Machine Science named after Academician U. A. Dzholdasbekov
Kyrmangazy str., 29, Almaty, Republic of Kazakhstan, 050010
Department of Electronics, Telecommunications and Space Technologies*
***Satbayev University
Satbayev str., 22, Almaty, Republic of Kazakhstan, 050013

The object of the study is the onboard diagnostic process in CubeSat nanosatellite systems, with particular emphasis on fault detection based on real-time analysis of multivariate telemetry data under real-time embedded operating conditions. The problem statement is the gap between the relatively low computational expenses and predictability of classical FDIR approaches and their limited ability to detect complex anomalies, and, conversely, the greater anomaly sensitivity and lower embedded performance of machine-learning models. A hybrid onboard diagnostic architecture that integrates telemetry acquisition, telemetry preprocessing, statistical feature extraction, a deterministic FDIR branch, a lightweight machine-learning branch, and decision fusion into one onboard diagnostics session has been elaborated. The proposed hybrid onboard diagnostic architecture has been evaluated based on hardware-in-the-loop testing using representative telemetry and fault cases for CubeSat systems. The evaluation metrics were the fault-detection accuracy, the detection latency, the CPU consumption, the memory consumption, and the resistance to telemetry noise. The hybrid onboard diagnostic architecture was compared to the conventional threshold-based FDIR system and standalone machine learning algorithms under the same HIL conditions. In the reported experiment setup, the proposed hybrid onboard diagnostic architecture demonstrated the fault-detection accuracy of 94%, while the classical FDIR method provided 71%, and the standalone machine-learning approach provided 88%. The average detection latency was decreased to 83 ms, in contrast to 120 ms and 95 ms, respectively. The embedded solution requires only 17.6% CPU and 58 KB of memory. Under the highest telemetry-noise level, the fault-detection accuracy of the proposed hybrid onboard diagnostic architecture decreases to 80%, whereas the standalone machine-learning and the classical FDIR baselines provide only 65% and 43% fault-detection accuracy, respectively

Keywords: CubeSat onboard diagnostics, fault detection, hybrid FDIR, lightweight machine learning

Received 02.02.2026
Received in revised form 10.04.2026
Accepted 20.04.2026
Published 30.04.2026

How to Cite: *Kuttybayeva, A., Zhamalova, S., Boranbayeva, A., Myrzayeva, Z., Imasheva, G., Kaskatayev, Z., Chinibayev, Y., Ospanova, N., Abdullayev, M., Tazhen, K. (2026). Development of a hybrid onboard diagnostic architecture with lightweight machine learning for resource-constrained CubeSat systems. Eastern-European Journal of Enterprise Technologies, 2 (9 (140)), 72–85. https://doi.org/10.15587/1729-4061.2026.358310*

1. Introduction

CubeSat nanosatellites have become a dominating platform in modern space missions because of the low costs, rapid development cycle, and accessibility for use in academic and commercial applications. Over the last decade, their deployment has been greatly enhanced and are supporting the Earth observation, communication experiments and in orbit technology demonstrations. Despite these advantages, CubeSat systems have tight constraints in onboard resources such as power supply, computation capability, memory capability, and communication bandwidth.

These limitations directly have an impact on system reliability and fault tolerance. Unlike big space probes, CubeSats cannot afford to have extensive redundancy in their hardware and cannot have an operator on the ground 24 hours a day, so autonomous on-board diagnostics is a mission-critical requirement for mission success. In such conditions, the diagnosis system must not only detect faults accurately, it must also work on constraints of embeddedness and time and support timely recovery actions.

Classical fault detection, isolation, and recovery (FDIR) approaches are extensively used in aerospace systems because they are based on deterministic behavior while having a low computational overhead. These methods are usually built on thresholds, rules & predefined fault trees. However, they are limited in their effectiveness in handling complex and nonlinear or previously unseen anomalies particularly in dynamic orbital environments. As CubeSat missions become more complex, deterministic approaches to diagnosing a mission become less important.

Recent innovations in data-driven techniques have introduced machine learning techniques in spacecraft diagnostics, to better understand and provide higher sensitivity to slight deviations in data obtained by telemetry systems and to detect anomalies at early stages. These approaches offer adaptive capabilities and are able to catch multivariate and temporal dependencies that are not available to traditional monitoring methods. However, their direct use in CubeSat systems is very difficult because of high computational needs, limited interpretability and the requirement for deterministic and predictable behavior in safety-critical aerospace systems.

Therefore, current diagnosis methods have a basic trade-off relationship among the computational efficiency, detection accuracy and operational reliability. Classical FDIR systems are efficient and predictable but not flexible and machine learning-based techniques are sensitive and flexible but are also incompatible with embedded constraints and real-time requirements.

From a practical point of view, the improvement of onboard diagnostic systems can have a significant impact in terms of CubeSat mission reliability and decreasing the dependence on ground control; furthermore, it will allow autonomous fault management in case of lack of communication availability. This is especially important for deep space missions, for distributed satellite constellations and long-duration operations, for which immediate ground intervention is not possible.

Thus, there is a clear need for study focusing on the development of diagnostic approaches that can at the same time ensure detection accuracy, real-time capability, robustness to noise and compatibility with embedded CubeSats platforms. Consequently, study dedicated to hybrid onboard diagnostic architectures combining a deterministic and a data-driven approach are relevant and necessary in modern aerospace engineering. This study addresses this need by developing and experimentally validating a hybrid onboard diagnostic architecture for CubeSat systems under embedded resource constraints.

2. Literature review and problem statement

Fault detection, localization, and recovery in aerospace systems are traditionally based on deterministic control logic, including threshold control, rule-based diagnostics, and predefined recovery procedures. In paper [1], it is summarized the basic principles of fault diagnosis systems and it was shown that deterministic approaches provide interpretation, low computational complexity, and compatibility with critical applications. However, these methods are most effective when the fault conditions are known in advance and can be clearly documented. This significantly limits their ability to detect weak, nonlinear, or previously non-existent anomalies.

In paper [2], fault diagnostics for dynamic systems has been considered from the point of view of model-based reasoning. Analytical redundancy and residue-based control have been shown to be able to improve the location of defects compared to simple limit control. However, due to the dependence of such methods on sufficiently accurate mathematical models, unresolved problems remain. In CubeSat on-board systems, this requirement is difficult to meet, as the behavior of subsystems varies with time-changing environmental conditions, parameter uncertainty, external disturbances, and communication between subsystems. As a result, model-based diagnostics are difficult to maintain in real working conditions.

Another direction was proposed in paper [3], in which methods for detecting defects based on the history of the process were considered. Data-based diagnostics have been shown to be able to detect complex dependencies in multidimensional observations even in the absence of a clear physical model. This makes such approaches attractive for telemetry analysis, where system degradation can be seen as a gradual change in correlation parameters rather than direct threshold violations. However, the methods considered in [3] are designed primarily for industrial process applications and were not originally intended for built-in aerospace platforms with strict time limits and computing resources.

The use of artificial intelligence in space systems was analyzed in paper [4]. AI-based methods can improve onboard autonomy, increase sensitivity to detect anomalies, and reduce dependence on ground control. In addition, unresolved problems associated with the limited resources of on-board computing, reduced interpretation, complexity of verification, and the need for predictable behavior in critical aerospace systems were noted. Thus, although artificial intelligence has significant diagnostic potential, its direct use in CubeSat on-board systems is still limited by operational reliability requirements.

The current level of CubeSat on-board computers was analyzed in paper [5] important practical limitations were noted. CubeSat computing platforms remain limited in terms of processor performance, memory size, energy budget, and fault tolerance. The reason for this is the strict physical and energy constraints of nanosatellite platforms. This makes direct deployment of resource-intensive diagnostic algorithms impractical, even if such algorithms demonstrate high detection efficiency in laboratory conditions. Therefore, any on-board diagnostic architecture for CubeSat must be designed with clear consideration of the limitations of the built-in resources.

In paper [6] an approach based on deep clustering has been proposed to detect anomalies in satellite telemetry. Data-based methods have been shown to be able to capture hidden structures in the multidimensional streams of telemetry and detect anomalies that are difficult to detect using only classical control rules. However, unresolved issues related to the on-board implementation remain. The method described in [6] is mainly concerned with anomaly recognition performance and cannot fully solve the problems of deterministic execution, limited memory use, and real-time deployment on CubeSat-class hardware.

A method to reduce the gap between diagnostic efficiency and embedding capability was proposed in paper [7], in which a simplified and annotated scheme for detecting defects in data – based aerospace sensors was developed. It has been shown that computational complexity can be reduced while maintaining acceptable detection performance and improving interpretation. This approach is especially relevant for aerospace systems with limited resources. However, the [7] method focused on sensor-level fault detection and did not provide a full on-board architecture that combined detection, isolation, and recovery under deterministic security constraints.

A broader review of this area was made in the study [8], which methods for detecting anomalies in the telemetry data of spacecraft were systematically considered. The field of the study has been shown to have evolved from classical statistical and marginal methods to machine learning, deep learning, and hybrid methods. However, [8] it has also shown that many existing studies are limited to autonomous analysis, are aimed only at detecting anomalies, and not at completely eliminating onboard defects, and often do not take into account diagnostic accuracy, computational efficiency, and the possibility of using built-in spacecraft platforms.

The work [9] proposed a hybrid approach to detecting telemetry anomalies. It has been shown that combining several diagnostic principles increases sensitivity to minor telemetry fluctuations and increases the overall efficiency of detecting abnormalities. This approach was used to overcome some of the limitations inherent in purely deterministic methods and methods based only on data. However, the problems of real-time on-board control, deterministic priority in making decisions that are important for security, and compatibility with limited memory and computing resources inherent in CubeSat systems remain unresolved.

Another important issue was discussed in paper [10], where anomaly detection algorithms for satellite telemetry were evaluated in terms of comparative efficiency estimates. It has been shown that the obvious advantage of a particular algorithm may depend on the selected data set, verification strategy, and evaluation metrics. This means that many of the results presented in the literature should be interpreted with caution, especially if they were obtained offline or only in simulation conditions. Therefore, the point is not only in the development of specific algorithms, but also in the need for sufficiently realistic validation in conditions close to on-board use.

Thus, an analysis of the literature shows that existing diagnostic approaches can be divided into three main groups. Deterministic FDIR methods are effective in computation, can be explained, and can be promptly predicted, but they are limited in determining complex non-finite deviations. Model-based methods increase the consistency of diagnostics, but they are strongly dependent on the presence of accurate and supported system models, which is difficult in dynamically changing CubeSat environments. Methods based on data and artificial intelligence provide greater flexibility and sensitivity to multidimensional fluctuations in telemetry, but they often require large computational costs, are not sufficiently explained, have a deterministic logic of weakly integrated security, or are tested mainly in autonomous conditions.

Thus, an unresolved scientific and applied problem is the lack of sufficiently proven on-board diagnostic architecture for CubeSat nanosatellites, which simultaneously provides deterministic reliability, adaptive recognition of complex telemetry anomalies, compatibility with limited inline resources, the ability to perform in real time, and support for autonomous fault response in the event of intermittent ground communication.

All this suggests that it is reasonable to conduct a study to develop and experimentally verify a hybrid onboard fault detection, isolation and recovery architecture that combines the deterministic logic of FDIR with a light machine learning module for CubeSat nanosatellite systems.

3. The aim and objectives of the study

The aim of this study is to develop a hybrid onboard diagnostic architecture for CubeSat systems that combines deterministic FDIR logic with lightweight machine learning in order to improve diagnostic performance while preserving embedded feasibility.

To achieve this aim, the following objectives were accomplished:

– to propose a conceptual architectural solution for hybrid onboard diagnostics in CubeSat systems that integrates telemetry acquisition, telemetry preprocessing, statistical feature extraction, deterministic FDIR logic, lightweight machine-learning-based anomaly detection, and decision fusion under embedded resource constraints;

– to evaluate the fault-detection accuracy of the proposed hybrid onboard diagnostic architecture in comparison with classical threshold-based FDIR and standalone machine-learning diagnostics;

– to evaluate the detection latency of the compared diagnostic approaches under the same hardware-in-the-loop conditions;

– to assess the embedded feasibility of the proposed architecture in terms of CPU usage and memory consumption on a representative CubeSat-class processor platform;

– to assess the robustness of the proposed architecture to telemetry noise under progressively disturbed measurement conditions.

4. Materials and methods

4. 1. The object, hypothesis, assumptions, and scope of the study

The object of the study is the onboard diagnostic process in CubeSat nanosatellite systems with a particular emphasis on the detection and isolation of faults based on real-time analysis of multivariate telemetry data under real-time embedded operating conditions.

The hypothesis of the study is that the combination of lightweight machine-learning-based anomaly detection with deterministic FDIR logic allows earlier and more accurate fault detection in CubeSat onboard systems than classical threshold-based approaches, while preserving real-time execution and embedded feasibility. In this study, the improvement is evaluated using measurable criteria, namely fault-detection accuracy, detection latency, processor load, memory consumption, and robustness to telemetry noise [11].

In order to derive a univariate formulation of the methodology, it is assumed to be subject to the following:

1) the telemetry streams have a high enough temporal resolution to be diagnostic in real-time;

2) the measured telemetry variables have observable trends of both normal and malfunctioning operating conditions;

3) the chosen fault cases are reflective of common diagnostic situations in CubeSat onboard subsystems;

4) the onboard computing system will facilitate periodic running of the diagnostic algorithm in the permitted telemetry update route;

5) the training and validation data is representative enough of the operational modes of the system under consideration.

This study is carried out under the following scope and simplifications:

1) additive noise in telemetry channels characterizes environmental disturbances and measurement problems as bounded noise;

2) the test is not conducted on all possible subsystem faults, but only on a representative set of those that are of interest to onboard monitoring, but not on the entire range of infrequent system-specific effects;

3) the interactions between cross-subsystems are represented in an aggregated way by a set of telemetry dependencies and obtained statistical characteristics, without developing a complete model of all the subsystem interactions;

4) the machine-learning model is trained offline and deployed only onboard to make inference, which aligns with the real-world limitations of implemented embedded CubeSat;

5) validation is done in a hardware-in-the-loop setting of realistic onboard envious conditions, but not yet on in-orbit telemetry.

In the present work, the diagnostic process is considered as a sequence of telemetry acquisition, preprocessing, feature formation, rule-based monitoring, machine-learning-based anomaly recognition, and diagnostic decision generation within the onboard computing loop. The study focuses on diagnostic operation under strict limitations in computational resources, memory, and execution time, which are characteristic of CubeSat-class platforms.

The proposed study considers the interaction between deterministic fault supervision and data-driven anomaly recognition in a unified onboard diagnostic architecture. The diagnostic task is formulated for telemetry streams reflecting the operating state of the electrical power subsystem, onboard computer, communication subsystem, and sensor subsystem. The considered architecture is intended for situations in which ground contact may be intermittent and diagnostic decisions therefore have to be generated autonomously on board. A functional representation of the considered onboard diagnostic process in Fig. 1.

In this way, the defined object, hypothesis, assumptions, and scope provide a methodological framework of the further elaboration of the proposed hybrid onboard diagnostic architecture and of the experiment that is going to be made in the following sections.

The synthesis of the proposed hybrid onboard diagnostic architecture was carried out as a constrained multi-criteria design problem. The target was to maximize diagnostic performance while preserving embedded feasibility [12]. The primary performance criteria were fault-detection accuracy and detection latency. The embedded feasibility criteria were processor load, memory consumption, and deterministic execution within the telemetry update cycle. Robustness to telemetry noise was considered as an additional criterion of practical reliability.

The adjustable parameters used were the window length for telemetry features, composition of the statistical feature vector, the classifier itself and its complexity, the weight parameter α, and the threshold value τ. The architecture chosen was that which presented the best compromise in terms of diagnostics and embedded implementation in relation to the restrictions on CPU, memory consumption, determinism, safety prioritization, and real-time operation aboard the platform.

4. 2. Telemetry variables, monitored channels and representative fault scenarios

The proposed hybrid onboard diagnostic architecture works on multivariate telemetry on the main CubeSat subsystems in each diagnostic check. In the current work, telemetry is regarded as the main source of data used to diagnose deviations of nominal operation and identify patterns associated with faults and create the input to the deterministic and machine-learning-based branches of diagnosis. The monitored telemetry channels had been chosen in a manner that would capture the most significant operating conditions of the onboard electrical, computational, communication and sensing subsystems in the presence of embedded real time conditions.

The variables under monitoring are electrical, thermal, computational and communication indicators. In the electrical power subsystem, the telemetry channels would be battery voltage, battery current, power bus voltage and battery temperature. In the case of the onboard computer, the variables to monitor are processor load, task-execution delay and watchdog status indicators. In the case of the communication subsystem, the telemetry channels are received signal quality, ratio of packet-loss and transmission-state indicators. In the case of the sensor subsystem, the variables monitored can be considered measured temperature, sensing outputs (voltage) and signal-consistency indicators. This list of choices of telemetry variables is the only reason that the diagnostic system can address direct threshold violation as well as less obvious multivariate deviations that may be related to degradation processes or non-standard interactions between the subsystems.

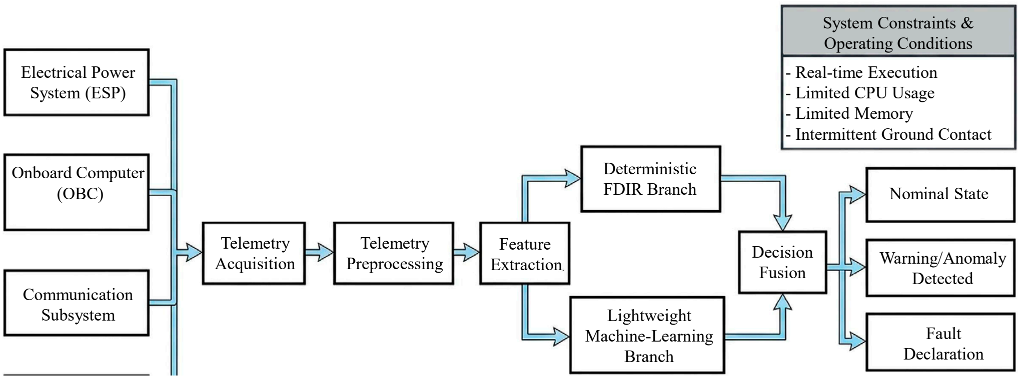

Fig. 1. Functional representation of the onboard diagnostic process in a CubeSat system

The proposed study does not view the telemetry channels as independent or independent variables, but the combined activity of these variables describes the health of the CubeSat system. This is significant because not all abnormal operating conditions are expressed as the overt threshold exceedances in a single channel, but as the coordinated variations in multiple variables with time. This is why the selected telemetry set is supposed to benefit the rule-based threshold monitoring and the feature-based anomaly detection.

In order to test the architecture during the representative onboard conditions, a list of fault scenarios was defined to test the key monitored subsystems. These scenarios were chosen to represent some of the common diagnostic events which can occur when operating CubeSat and which can be reasonably modeled in a hardware-in-the-loop validation setting. The fault cases that the study has looked into are the presence of undervoltage and over temperature in the electrical power subsystem, excessive load and execution delay growth in the onboard computer, communication quality degradation in the form of increased packet loss of reduced link stability and sensor problems of bias, drift or as sudden abnormal values. The fault scenarios of interest include sudden fault and slow onset of any anomaly.

All flawed situations were correlated with referent diagnostic labels to be utilized in appraisals. Under the current formulation the labels were classified into two major active categories which are the nominal condition and faulty condition. Simultaneously, the physical cause of every fault occurrence was maintained in the process of scenario generation so as to determine how the proposed architecture reacts to various subsystem-specific perturbations. This form enables the possibility to compare the reaction of the hybrid method to the baseline diagnostic methods under similar telemetry conditions. Table 1 the representative telemetry channels and the fault scenarios of the cases of interest used in the study.

Table 1

Representative telemetry channels and fault scenarios used in the study

Subsystem	Monitored variable	Symbol	Unit	Representative nominal condition	Representative fault scenario
Electrical power system	Battery voltage	V_{bat}	V	Stable regulated range	Undervoltage
Electrical power system	Battery temperature	T_{bat}	°C	Nominal thermal range	Overtemperature
Electrical power system	Bus current	I_{bus}	A	Nominal load profile	Current anomaly / overload
Onboard computer	CPU load	L_{cpu}	%	Stable execution load	Processor overload
Onboard computer	Task execution delay	D_{task}	Ms	Nominal scheduling delay	Timing degradation
Communication subsystem	Packet-loss ratio	P_{loss}	%	Stable low-loss link	Communication degradation
Communication subsystem	Link-quality indicator	Q_{link}	dB or normalized value	Nominal communication quality	Link-quality drop
Sensor subsystem	Sensor output bias	B_{sens}	normalized value	Stable calibrated signal	Sensor bias
Sensor subsystem	Sensor drift/abnormal reading	S_{sens}	normalized value	Consistent signal behavior	Drift or abrupt anomaly

The set of defined telemetry and typical failure conditions provide the framework of the experiment which is the foundation of the suggested diagnostic approach. They define what states are observed in the system, what is introduced in the deviation to the system to be abnormal and what diagnostic data is exposed to the architecture. It is based on this that, the next subsection outlines the telemetry preprocessing and feature-extraction process, which is applied to convert the signals being monitored into diagnostic inputs to be used in real time decision making.

4. 3. Telemetry preprocessing and feature-extraction procedure

The implementation of the proposed integrated diagnostic architecture makes it possible to obtain the initial telemetry of the subsystem using an algorithm for preprocessing/assigning functions performed in real time on a small integrated platform. The purpose of this process is to transform heterogeneous telemetry streams into a single diagnostic representation that can be ported to deterministic FDIR logic and anomaly detection based on machine learning. The cycle of this study is implemented in the form of a working pipeline, which runs in parallel with the telemetry collection cycle.

Assume that the definition of a multidimensional telemetry vector at a discrete time is k

$$x(k)=\left[x_1(k),x_2(k),\ldots,x_m(k)\right]^T, \tag{1}$$

where $x_i(k)$ – the i-th channel of the observed telemetry, and m – the number of telemetry variables that must be taken into account in the diagnostic cycle. Synchronized measurements of the power subsystem, on-board computer, communication subsystem and sensor subsystems are synchronized in accordance with the vector $x(k)$.

Since different telemetry channels have different physical properties, digital range, and sensitivity to interference, preprocessing of the raw measurements is performed first. Pre-treatment is the synchronization, normalization, elimination of disorders and treatment of abnormal single outbreaks. Synchronization ensures that all telemetry channels operate at the same diagnostic time stage. This is necessary because the hybrid architecture reacts to the behavior of the Combined subsystems, rather than to individual signals. When this is synchronized, normalization is performed, so that all telemetry variables change at the same scale, and the dominant signal in the diagnostic model is suppressed. This assignment uses minimum-maximum normalization

$$\tilde{x}_i(k)=\frac{x_i(k)-x_i^{\min}}{x_i^{\max}-x_i^{\min}+\varepsilon}, \tag{2}$$

where $x_i^{\min}$ and $x_i^{\max}$ – the minimum and maximum reference values of the i-th telemetry channel estimated from the normal operating range and ε – a small positive constant introduced to avoid division by zero.

After the telemetry channels are normalized, they are filtered in such a way as to minimize high-frequency measurement noise and short-term transient fluctuations that are not related to a real subsystem malfunction. In addition, individual outbreaks are considered as potential emissions and eliminated if they do not match the channel's behavior at local time. This is especially necessary in the integrated diagnostic environment under consideration, since abnormal jumps can affect threshold-based management, the quality of functions and processes used by the machine learning unit.

The second is the extraction of characteristics. The diagnostic system not only receives immediate standardized telemetry values, but also uses the obtained statistical and temporal characteristics, which improve the description of changes in subsystem behavior over time. For each of the telemetry channels, there is a sliding W-width analysis window from which representative descriptors are calculated. The highlighted characteristics are the moving average, the rate of change, the local variance, and the selected relationships between the parameters. The moving average for the *i*-th channel is defined as

$$\bar{x}_i(k)=\frac{1}{W}\sum_{j=k-W+1}^{k}\tilde{x}_i(j), \tag{3}$$

this averages out short-term fluctuations and highlights the local trend of the signal. The rate of change over time is calculated as

$$\Delta x_i(k)=\tilde{x}_i(k)-\tilde{x}_i(k-1), \tag{4}$$

this takes into account sudden changes and emerging disadvantages. The local variance of the sliding window is calculated as follows

$$\sigma_i^2(k)=\frac{1}{W}\sum_{j=k-W+1}^{k}\left(\tilde{x}_i(j)-\bar{x}_i(k)\right)^2, \tag{5}$$

this describes the short-term stability of the telemetry signal. There is also an assessment of the correlation between the parameters (in a small set of pairs of telemetry channels) to assess the consistent deviations of the subsystem, which otherwise could not be observed on an individual basis.

This will give a complete set of diagnostic characteristics, consisting of

$$\phi(k)=\left[\tilde{x}(k),\bar{x}(k),\Delta x(k),\sigma^2(k),r(k)\right]^T, \tag{6}$$

where $\tilde{x}(k)$ – the normalized telemetry vector; $\bar{x}(k)$ – the moving averages; $\Delta x(k)$ – the vector of time gradients; $\sigma^2(k)$ – the local deviations; $r(k)$ – the set of descriptors selected for cross-channel correlations. This representation supports the properties of the instantaneous and temporary state of the embedded system and provides unified input data for the following diagnostic levels.

The chosen set of features was designed to detect both transient and gradual instances of abnormality in a subsystem, without imposing high computational costs [13]. Normalized telemetry represents the present state of the system; moving average shows short-term trends; temporal derivative detects sudden changes; local variance shows temporary instability; and correlation between channels indicates simultaneous deviations in subsystems. This particular combination of features is chosen as it increases detection capabilities and still allows real-time implementation on embedded systems.

The extracted functions are introduced along with the deterministic part of the architecture and the machine learning part of the architecture. The deterministic branch is the verification of thresholds and rules based on standardized telemetry and a selected indicator, while the machine learning branch is the detection of anomalies based on a complete set of characteristics. Therefore, preprocessing and feature extraction provide a common information base for both diagnostic units and help them make coordinated decisions based on built-in constraints in real time. The telemetry preprocessing and feature-extraction pipeline is shown in Fig. 2.

The following figure shows the diagnostic information. Fig. 2 shows the sequence of transformations of the subsystem's source telemetry into diagnostic inputs with functions and filters. This process increases the stability of the hybrid architecture, sensitivity to slow multidimensional anomalies, as well as the consistency and responsiveness of the resulting diagnostic solution based on a consistent and inefficient representation of integrated status monitoring. The decision logic and simplified machine learning implementation presented in the next subsection are based on this preprocessing platform.

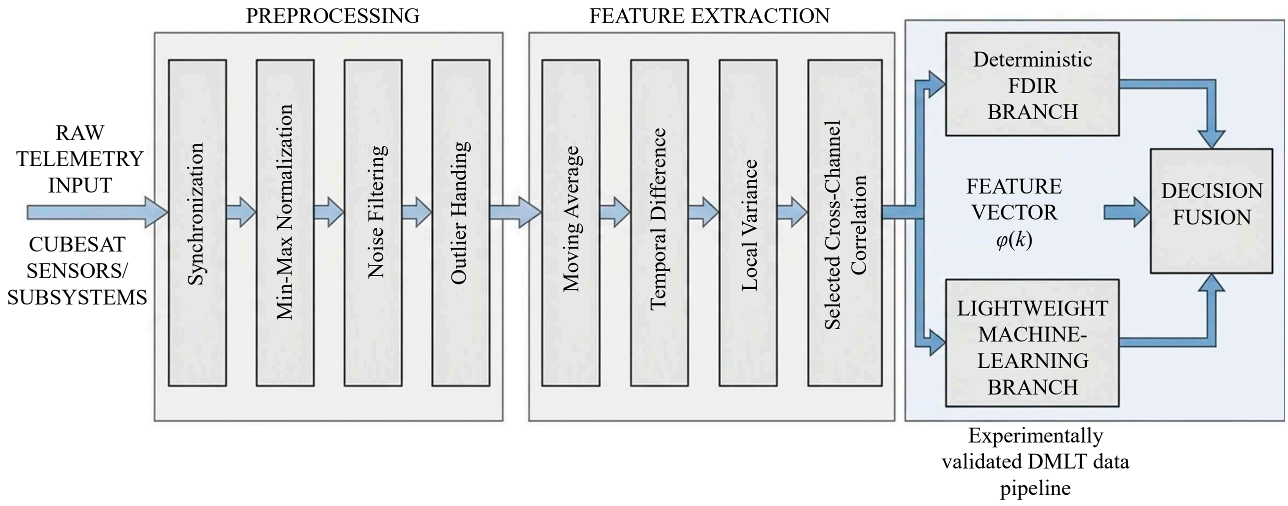

Fig. 2. Telemetry preprocessing and feature-extraction pipeline for the proposed onboard diagnostic architecture

4. 4. Deterministic decision logic, lightweight machine-learning model, and embedded implementation

The proposed hybrid onboard diagnostic architecture combines two additional diagnostic branches: the rule-based deterministic branch of FDIR and the branch of simplified machine learning. The purpose of this combination is to maintain the predictability and safety of threshold-based monitoring while increasing the sensitivity of the diagnostic system to weak, multidimensional, and nonlinear anomalies that can only be reliably detected using explicit rules. In this study, both branches work on the basis of common diagnostic data obtained using the telemetry preprocessing and characterization procedure described in the previous subsection.

The deterministic branch is responsible for monitoring known critical operating conditions using predefined thresholds, logical rules, and subsystem consistency checks. This branch applies to telemetry parameters, where error-related deviations can be directly related to the security status. Let be the result of a deterministic branch in the diagnostic cycle k, denoted as

$$d_{FDIR}(k)\in\{0,1\}, \tag{7}$$

where $d_{FDIR}(k) = 1$ corresponds to a rule-based error declaration, and $d_{FDIR}(k) = 0$ corresponds to the absence of a deterministic error condition. In practice, this device evaluates the thresholds of certain electrical, computational, thermal, and communication parameters and prioritizes safety-critical events that require a predictable on-board response.

The machine learning industry was developed to detect complex telemetry anomalies that are not explicitly described by deterministic rules [14]. The present study uses a simplified random structure classifier because it provides a favorable compromise between the ability to make nonlinear decisions, resistance to interference, relatively low memory footprint, and suitability for integrated implementation. The classifier works with the feature vector $\phi(k)$ defined in (6) and outputs a probabilistic result or a result based on the anomaly estimate

$$p_{ML}(k)\in[0,1], \tag{8}$$

where higher values of $p_{ML}(k)$ indicate a higher probability of abnormal behavior of the system. The machine learning model is trained offline using labeled telemetry segments corresponding to nominal and defective operating conditions and is used on board only for data output. Such a system meets the practical limitations of CubeSats, where on-board training is usually impossible due to limited computing and energy resources.

The structure of the random forest model is chosen so that it remains compatible with the built-in execution. The number of trees, tree depth, and functional usage were limited during the development of the model in order to reduce output time and memory footprint while maintaining sufficient anomaly detection capability. Therefore, the classifier is not designed to maximize the complexity of autonomous operations, but rather to provide a computationally efficient anomaly detection mechanism that can be deployed in real time on an on-board computer.

Random Forest has been chosen due to its practicality, having properties of nonlinearity and ability to cope with noise in the telemetry signal while maintaining low complexity in inference. However, the choice of the machine learning model in this study has not been made on the principle of maximizing its diagnostic capabilities autonomously, but rather based on the compromise with embedded implementation possibilities.

The final diagnostic solution is formed by combining the results of the deterministic branch and the machine learning branch at the decision-making stage. In the present work, the fusion score is defined as

$$s(k)=\alpha d_{FDIR}(k)+(1-\alpha)p_{ML}(k), \tag{9}$$

where $\alpha\in[0,1]$ – the weighting coefficient controlling the relative contribution of deterministic supervision and machine-learning-based anomaly recognition.

The parameters of the deterministic decision-making branch, namely, the fusion coefficient α and decision threshold τ, were empirically chosen to guarantee the preservation of the priority principle of the deterministic decision under all critical safety conditions, while maximizing the fault-detection accuracy of the hybrid decision subject to the imposed embedded restrictions. Hence, the configuration of the decision-fusion stage took into account not only diagnostic performance, but also predictable embedded behavior. The resulting diagnostic output is then determined by the decision rule:

$$d(k)=\begin{cases}1, & d_{crit}(k)=1,\\ 1, & s(k)\geq\tau,\\ 0, & s(k)<\tau,\end{cases} \tag{10}$$

where $d(k)$ – the final binary diagnostic solution, τ – the trigger threshold, and $d_{crit}(k)$ – the signal with strict safety priority generated by the deterministic branch in the presence of a critical failure condition. This formulation ensures that machine learning results cannot replace a deterministic, security-critical statement. Thus, the fusion mechanism increases diagnostic sensitivity while maintaining predictable behavior on board.

A schematic representation of the hybrid decision logic and integrated diagnostic implementation is shown in Fig. 3.

The integrated implementation of the proposed architecture is carried out on a typical arm Cortex-M-based platform, which meets the limitations of CubeSat embedded computing. The diagnostic algorithm is performed periodically synchronously with the telemetry update cycle and consists of collecting telemetry, preprocessing, calculating characteristics, deterministic evaluation of rules, machine learning output, and a combination of solutions. The implementation was designed with three practical limitations in mind: limited CPU usage, limited RAM and flash memory usage, and deterministic execution time compatible with built-in real-time monitoring.

From an implementation point of view, the deterministic branch has little impact on computational costs, while the machine learning branch incurs additional but controlled inference costs. Thus, the goal of the hybrid design is not only to increase the efficiency of fault detection, but also to ensure that this improvement is compatible with the limited integrated resources of the CubeSat platforms. Using a lightweight classifier and a compact set of functions helps maintain this balance.

In such a way, the final decision was obtained based on the synthesis of deterministic supervision, feature-based detection, and decision fusion subject to imposed constraints. Such an approach allowed improving diagnostic performance without violating embedded constraints of low CPU consumption, limited RAM capacity and deterministic operation.

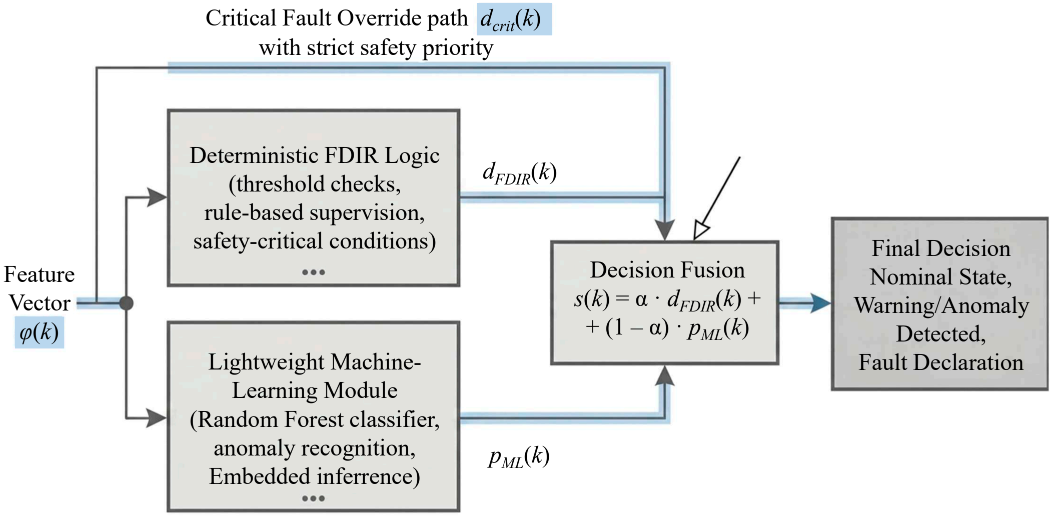

Fig. 3. Hybrid decision logic and decision-fusion mechanism of the proposed onboard diagnostic architecture

The proposed hybrid onboard diagnostic architecture was validated via an experimental test bed that replicated representative CubeSat operating conditions using embedded computational constraints [15]. The goal of the test bed was not limited to validation of the diagnostic functionality itself, but also of its implementation under realistic computational resource limitations.

The experimental test bed included the following four key modules: a representative embedded onboard processor, telemetry simulation, fault injection, and host-side logging environment. Onboard the embedded system, the whole process of telemetry acquisition, preprocessing, feature extraction, deterministic rule evaluation, machine learning inference, and decision fusion was performed. Telemetry simulation module was used to generate simulated telemetry vectors that replicated normal operational telemetry for the set of representative CubeSat subsystems that are being monitored (electrical power subsystem, onboard computer, communication subsystem, sensor subsystem). Fault injection was used to introduce disturbances to some or all channels of telemetry vector in order to simulate fault conditions corresponding to representative fault scenarios that are described in Section 4. 2. Host-side logging environment included experiment setup and telemetry acquisition, time stamping, logging of reference states, and post-processing analysis of results. The dataset consisted of two representative types of operating conditions: nominal and faulty. Nominal operating conditions included stable operation of the monitored subsystems in expected operational ranges of monitored telemetry values. Faulty operating conditions included deviations from the expected behavior as specified by the set of representative fault scenarios, including: undervoltage; over temperature; overload in the computer; timing degradation; communication degradation; sensor bias; sensor drift. Both abrupt and gradually developing faults were considered in order to test sensitivity to different fault evolution patterns.

In each experimental run, the simulated telemetry stream was continuously injected into the embedded system. During each cycle, the telemetry vector $x(k)$ was obtained, synchronized, normalized, filtered, and transformed into the feature vector $\phi(k)$ in accordance with the described pre-processing procedure. Then, the deterministic FDIR and machine learning branches were applied to the same input in parallel. Their outputs were combined into the final decision according to the described decision fusion rules. This was repeated until the end of the experiment to obtain sufficient statistics. For fair comparison of the proposed hybrid architecture with other methods, two baseline algorithms were also evaluated under the same telemetry and hardware conditions. These included classical threshold-based FDIR method and machine learning-based anomaly detection method. The first baseline relied on deterministic rules and thresholding only. The second baseline included machine learning component but did not involve deterministic decision fusion rules. Such an approach enables one to evaluate the benefit of hybridizing deterministic and machine learning approaches. Accuracy of fault detection was taken as a primary measure of success of any FDIR algorithm. It was defined as follows

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{11}$$

where TP – true positives (correctly detected faulty states), TN – true negatives (correctly recognized nominal states), FP – false positives, and FN – false negatives. This metric evaluates overall ability of FDIR systems to detect faulty states. Latency of fault detection is another important criterion of diagnostic effectiveness. The design and analysis of the diagnostic algorithm were done according to the adopted synthesis criteria of the proposed architecture. Namely, diagnostic capabilities were measured by fault-detection accuracy and latency, while feasibility was characterized by CPU utilization and RAM capacity. Additionally, robustness of the algorithm was assessed by introducing noise in telemetry.

This measure defines the rapidity of the detection process and is especially relevant to onboard autonomy applications. It was evaluated as follows

$$\bar{L} = \frac{1}{N_f} \sum_{j=1}^{N_f} \left(t_{det}^{j} - t_{fault}^{j} \right), \tag{12}$$

where t_{fault}^{j} – the moment when j-th fault was introduced and t_{det}^{j} – the corresponding fault-detection time and N_f – the number of evaluated fault events. This metric evaluates timeliness of responses to abnormal conditions development.

Besides fault detection accuracy and latency, resource usage of onboard algorithms was also analyzed in order to evaluate their feasibility. The measures considered here included the following: processor load and memory requirements. The first metric was evaluated as the average CPU utilization caused by periodic execution of diagnostic procedure on embedded hardware. The second was defined as the amount of memory occupied by diagnostic modules, which include feature buffer, machine learning models, and decision variables. These two metrics allow to analyze feasibility of diagnostic procedures in terms of available computational resources. Finally, robustness to telemetry disturbance was evaluated through introducing random noise into telemetry stream and analyzing results under increasingly disturbed conditions.

The aim of such evaluation was to analyze the impact of imperfect sensor measurements on diagnostic decisions. For each noisy case, the same diagnostic experiment was conducted and diagnostic results obtained. Then, the fault detection accuracies for the compared methods were evaluated. This allows comparing degradation of hybrid and baseline methods when telemetry becomes more distorted. As can be seen from above, all comparative performance results were obtained using the same experimental test bed configuration and protocol. All methods were tested under identical fault scenarios, telemetry conditions, hardware configuration, and metric definitions.

Therefore, the comparative results that are presented in the results section were gathered in the same test bed environment rather than using separately configured experiments. The reason why this issue needs to be discussed is that it enables reproducibility and clarity of reported results.

5. Results of the development of the hybrid onboard diagnostic architecture for CubeSat systems

5. 1. Proposed conceptual architectural solution

The first result of the study is the proposed conceptual architectural solution for hybrid onboard diagnostics in resource-constrained CubeSat systems. The solution combines deterministic FDIR supervision with lightweight machine-learning-based anomaly recognition within one embedded diagnostic loop and establishes the structural basis for the subsequent quantitative evaluation.

The architecture design relied on the fusion of two mutually compatible diagnostic approaches, subject to CubeSat embedding restrictions. On the one hand, the deterministic approach was kept intact for maintaining consistent failure declaration and criticality prioritization, while the second part of the architecture design involved the development of the lightweight machine learning model for improving anomaly detection sensitivity with respect to multivariate data, which cannot be adequately identified using threshold logic alone.

The described embedded diagnostics architecture is implemented as a software-implemented centralized onboard diagnostic architecture as part of the embedded computing cycle of the CubeSat platform with limited resources. It is designed to perform real-time telemetry measurements of subsystems, determine the non-standard state of the operating environment, and develop diagnostic solutions limited by high CPU usage, memory consumption, and runtime. The architecture is a combination of deterministic control and simplified data-driven analysis in a software-implemented onboard diagnostic architecture.

The CubeSat platform in question consists of four main subsystems that generate telemetry, namely: power systems; on-board computer (OBC), communication subsystem, and sensor subsystem. EPS provides telemetry related to power generation and battery status, currents, and controlled voltage levels. OBC reports the status of calculations, including CPU usage, execution status, and scheduler flags. The communication subsystem provides telemetry related to connection quality, packet suppression, and transmission. The sensor subsystem provides values for operating variables such as temperature, voltage, current, and other reference values needed to assess conditions on board.

All telemetry coming from the subsystem is transmitted via the built-in data bus and stored in the OBC, while the proposed diagnostic module is implemented as a real-time software module. The diagnostic module is regularly used in conjunction with the telemetry collection and represents a centralized technological chain. This central structure reduces the cost of communication between diagnostic components, simplifies data synchronization, and helps timely decision-making for integration diagnostics.

Structurally, the proposed architecture is represented by the following functional blocks: obtaining telemetry, preprocessing telemetry, feature extraction, deterministic FDIR logic, simplified anomaly detection based on machine learning and combining solutions. The measurement stream in the telemetry unit collects data on synchronized subsystems with each diagnostic cycle. Normalization, filtering, and processing of measurement interference and deviations from the norm are performed in the preprocessing unit. The feature extraction unit is designed to convert raw telemetry into statistical and temporal characteristics that can be used to analyze data and perform diagnostics. This encoded data is then transmitted simultaneously to two decision-making branches.

The first entry is a deterministic FDIR layer that verifies the threshold value, monitors compliance with regulations, and monitors compliance with predefined critical security conditions. This unit deals with the rapid and predictable detection of known violations and takes priority in critical situations on board. The second layer is the lightweight machine-learning layer which is a processor that is configured to process the extracted telemetry features, to find complex, weak, nonlinear and multivariate deviations that cannot be notified easily with explicit threshold logic. This branch in the current architecture can be considered as a means of adaptive diagnostic that helps to widen the range of sensory of the system towards anomalies that occur at an earlier stage.

The results of the branches are fed together at the decision-fusion block that forms the final state of diagnostic of the system. The fusion mechanism is customized to achieve deterministic reliability and is more sensitive in diagnosis via data-based recognition. In the case of a safety-critical situation, the deterministic FDIR branch is prioritized above the machine-learning branch to ensure onboard deterministic behavior and prevent the engineer of the diagnostic outcome overpowering non-deterministic control decisions. It means one diagnostic decision of nominal operation, warning-level anomaly or known fault condition.

Under this architecture, thus, a structural base of consistency onboard diagnostics of real-time of the CubeSat systems is established. It permits the interoperability between computationally efficient rule-based monitoring and lightweight anomaly detection, as well as compatibility with the embedded resource constraints. A structural representation of the proposed hybrid onboard diagnostic architecture is shown in Fig. 4.

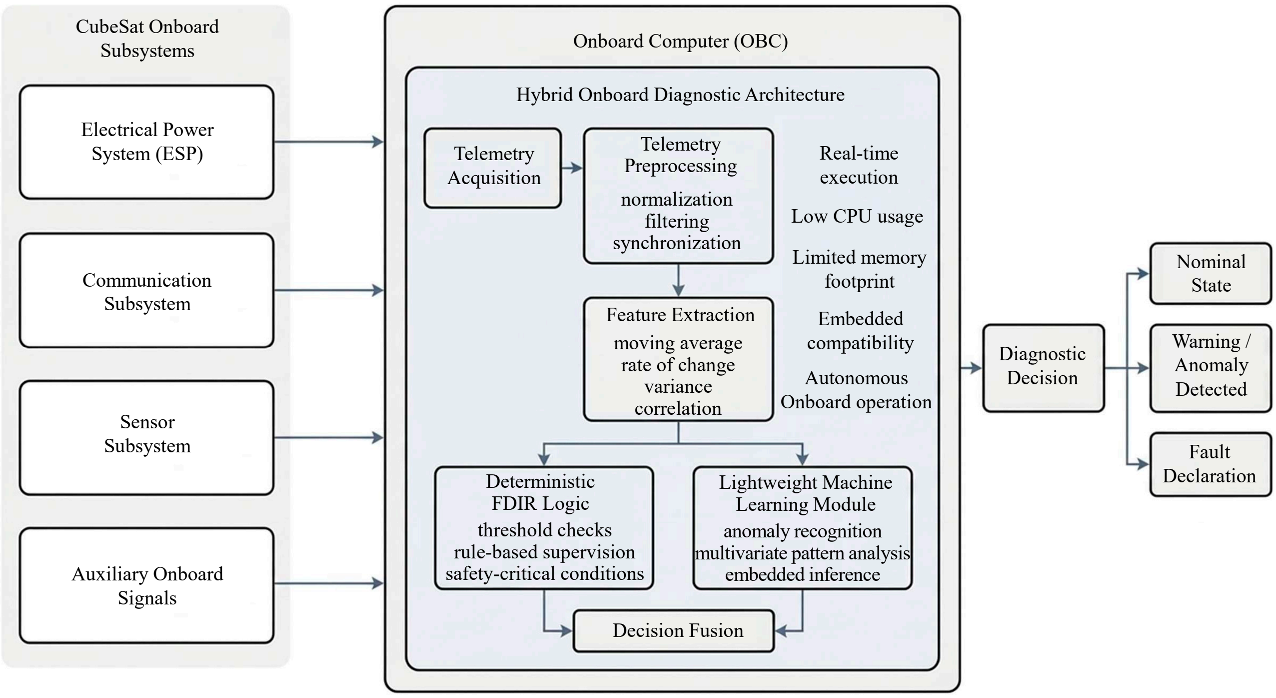

Fig. 4. Structural representation of the proposed hybrid onboard diagnostic architecture integrated into the CubeSat onboard system

As shown in Fig. 4, the proposed diagnostic architecture is organized as a diagnostic software module within the onboard computer in which subsystem telemetry is collected, preprocessed, analyzed in parallel by deterministic and machine-learning branches, and then merged into a unified diagnostic decision. Such an organization ensures compatibility with real-time CubeSat operation while preserving deterministic supervision of safety-critical states. This structural representation provides the basis for the subsequent description of telemetry variables, monitored parameters, and representative fault scenarios used in the experimental evaluation.

5. 2. Fault detection accuracy

The fault-detection performance of the proposed hybrid onboard diagnostic architecture was tested by a data set that includes the nominal operating conditions as well as representative fault case corresponding to CubeSat subsystems.

The evaluation was done by comparing three approaches to the diagnosis:

1) classical threshold based FDIR;
2) anomaly detection using a machine learning model;
3) the proposed hybrid onboard diagnostic architecture.

Detection accuracy is defined as the ratio between the correct identification of the states in the system (normal and faulty) divided by the total number of instances evaluated. The comparison of the detection accuracy of the evaluated methods is shown in Table 2.

Table 2

Fault detection accuracy comparison

Method	Accuracy (%)
Classical FDIR	71
ML-based	88
Hybrid (proposed)	94

The hybrid onboard diagnostic architecture has the highest accuracy of the evaluated methods. The following performance observed shows that the combination of the deterministic and data-driven diagnostic mechanisms allows for better identification of the threshold-exceeding faults as well as subtle anomalies in the telemetry data and allows for a more complete ability to diagnose the issue than did the individual approaches.

5. 3. Detection latency

Detection latency is described as the time between occurrence of the fault and the identification of the fault by the diagnostic system. This metric is very important for CubeSat missions where the late detection of delay can cause fault propagation and system degradation.

The evaluations of the investigated diagnostic approaches were performed for the same hardware-in-the-loop (HIL) conditions and their latency performance was measured. The comparison of the detection latency of various diagnostic methods is shown in Fig. 5.

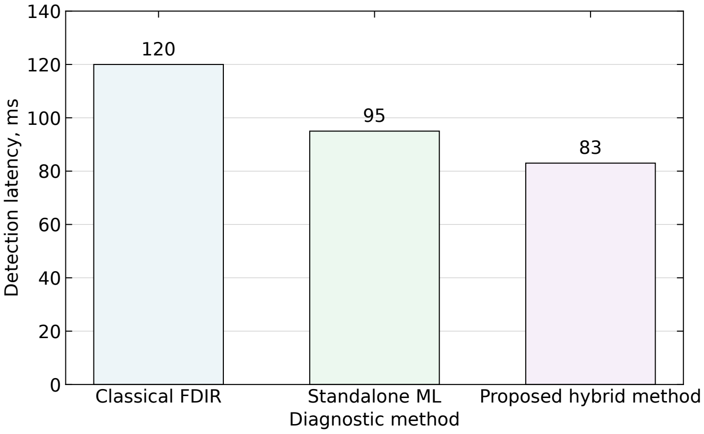

Fig. 5. Comparison of fault detection latency for different diagnostic approaches

The hybrid diagnostic approach shows the shortest latency out of the methods being evaluated. The resulting reduced detection latency means integration of the machine learning-based recognition of anomalies with deterministic monitoring allows for faster recognition of deviations in system behavior ensuring real-time execution capability.

5. 4. Computational resource usage

The computational performance of the proposed hybrid diagnostic system was tested on an embedded platform representative of CubeSat onboard computers. The evaluation involves utilization of the processor and memory in actual operation. The computational resource use of the hybrid diagnostic system, which was measured, is summarized in Table 3

Table 3

Embedded system resource consumption

Metric	Value
CPU usage	17.6%
Memory usage	58 KB

The results show that the diagnostic system works within the constraints of typically available CubeSat onboard hardware. The division of computational load between diagnostic components can be seen in Fig. 6.

The machine learning part is responsible for a slight increase in overall computational load as compared to the deterministic FDIR logic. Despite the extra computational overhead added to the system by the machine learning module, the system, as a whole, is still within acceptable limits for real time embedded operation on CubeSat platforms.

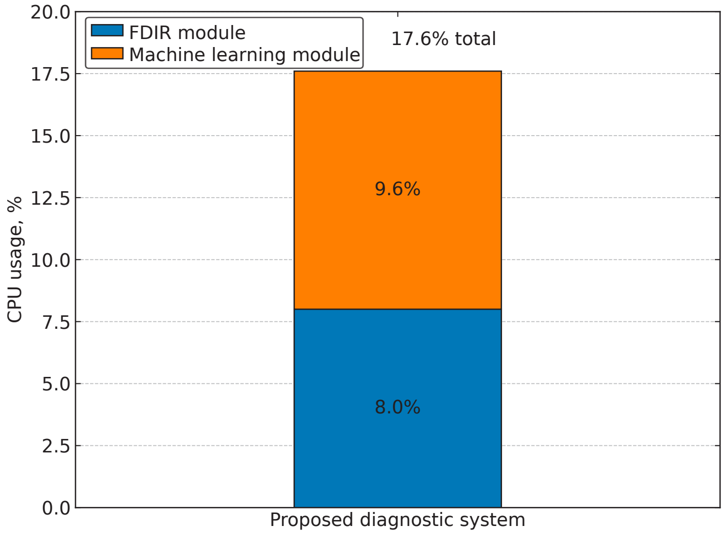

Fig. 6. Computational load distribution of the onboard diagnostic system

5. 5. Robustness to telemetry noise

The robustness of the diagnostic system was tested with different amounts of noise added to the telemetry data. Noise was simulated as bounded stochastic disturbances on all the measured parameters. Detection accuracy was measured with respect to increasing noise levels to evaluate the stability of each of the diagnostic approaches. The effects of noise on the accuracy of fault detection for various diagnostic techniques are shown in Fig. 7.

The hybrid diagnostic approach is consistent in the highest detection accuracy over all noise levels. The results show consistent degradation trends in performances of all the evaluated methods while retaining the relative performances under increasing noise conditions.

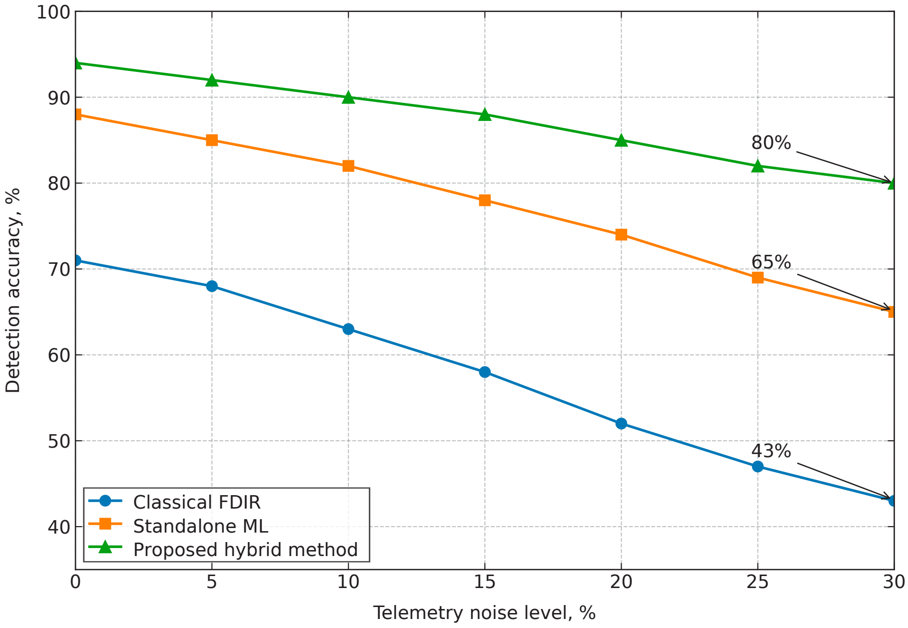

Fig. 7. Robustness of diagnostic methods under increasing telemetry noise levels

6. Discussion and interpretation of results

The obtained results show that the proposed hybrid diagnostic architecture provides consistent improvements in fault-detection accuracy, detection latency, and robustness under noisy telemetry conditions when compared with classical FDIR and standalone machine-learning approaches. As shown in Table 2 and Fig. 5, the hybrid method achieved 94% accuracy and 83 ms latency, whereas the classical FDIR and standalone machine-learning baselines achieved 71%/120 ms and 88%/95 ms, respectively. These results indicate that parallel operation of deterministic supervision and data-driven anomaly recognition improves both diagnostic sensitivity and response speed.

These experimental results prove the effectiveness of the chosen synthesis approach of the proposed architecture. This improvement was not due to increased complexity, but due to the combination of deterministic supervision, computational efficiency of statistical features computation, lightweight classification, and constrained decision fusion.

The reason for the increased detection accuracy can be found in the telemetry representation used in the machine learning branch as well. According to the feature vector defined in (6), the diagnostic model utilizes not only raw telemetry values but also derived statistical features and temporal features such as moving averages, gradients, variance, and relationships between parameters. This enables the system to detect gradual degradation, and correlated anomalies that cannot be reliably detected by static threshold monitoring alone. Therefore, the higher accuracy given in Table 2 is attributed to not only the fusion mechanism in (9), but also to the richer description of the telemetry used in the feature extraction stage.

The reduction in the detection latency in Fig. 5 is explained by the parallel diagnostic structure of the proposed architecture. In the deterministic branch, fault declaration is triggered when threshold or rule violations are detected, whereas the machine learning branch responds earlier to developing abnormal telemetry patterns represented by the feature vector in (6). The final diagnostic decision is then made with the fusion mechanism in (9). As a result, the system is not obliged to wait for an explicit threshold exceedance before indicating the abnormal behavior. This explains the improvement in latency from 120 ms for the classical FDIR approach and 95 ms for standalone machine learning method to 83 ms for the proposed hybrid architecture as shown in Fig. 5.

The reported computational efficiency of the proposed architecture can be explained by is explained by the lightweight structure of the model of the machine learning and the distribution of the processing tasks of the deterministic and data-driven branches. According to Table 3, the total CPU load of the diagnostic algorithm is 17.6% and the memory consumption is 58 KB, so it is confirmed to be compatible with the embedded CubeSat constraints. Fig. 6 further shows that the processor load is shared between the FDIR module and the machine learning module, and that the additional cost of the data-driven branch remains moderate. Thus, the results presented in Table 3 and Fig. 6 demonstrate that this improvement in the diagnostic performance is accomplished without breaking the computational limits of the CubeSat onboard hardware.

The robustness results presented in Fig. 7 are explained by the combination of statistical feature extraction with ensemble-based classification which helps to reduce sensitivity to random fluctuations in the telemetry measurements. Since the feature representation in (6) contains instantaneous as well as aggregated characteristics of system behavior the final decision is less dependent on isolated disturbances of signals. In addition, the deterministic branch also provides a stabilizing affect by preserving rule-based supervision of critical system states. For this reason, the proposed range of telemetry noise levels and keeps up to 80% accuracy under severe disturbance conditions as shown in Fig. 7. Similar tendencies have been observed in recent spacecraft anomaly detection studies using spatio-temporal learning and multidimensional telemetry processing [16, 17], although such approaches often need more complex models than the one used in the present study.

The obtained results are also consistent with the recent results about progressive anomaly event alerting in spacecraft systems [18]. In the present work this tendency is manifested both in the improvement in the latent period demonstrated in Fig. 6, and in the greater sensitivity to weak deviations which do not immediately lead to explicit violations of the thresholds. Therefore, the proposed architecture improves not only the classification accuracy, but also the capability of the onboard system to react at earlier stages of the developing of abnormal behavior.

When compared with classical FDIR systems, the proposed architecture has a better adaptability to nonlinear and non-threshold-based anomalies because the final diagnostic decision is not decided based solely on predefined rules. At the same time, in contrast to purely machine learning-based solutions, the hybrid method maintains the deterministic prioritization for safety critical operation by the rule-based branch and the fusion logic, which is defined in (9). This trade-off between predictability and adaptability is the reason why the method is better than either individual paradigm given the same test conditions.

Another important result is that the proposed approach does not require highly accurate models of the physical subsystems. This separates it from model-based diagnostic techniques, performance of which is strongly dependent on the model fidelity under changing operational conditions. In the current architecture, the diagnostic capability is accomplished by telemetry preprocessing, statistical features extraction, anomaly recognition using machine learning, and deterministic decision supervision. This makes the approach structurally simpler, and still allows for acceptable diagnostic effectiveness for CubeSat onboard applications.

From a practical point of view, the obtained results suggest that the proposed hybrid architecture can be used for CubeSat missions that require an increased autonomy capability on board, in particular when the continuous contact with the ground control cannot be guaranteed. The combination of better accuracy (Table 2), lower detection latency (Fig. 5), acceptable resource usage (Table 3 and Fig. 6) and robustness in the presence of telemetry noise (Fig. 7) make the method of this study practical for autonomous on-board diagnostics in resource-constrained nanosatellite systems.

Despite these advantages, a number of limitations of the present study should be acknowledged. First, for the machine learning part, generalization to fault conditions not seen in the training data depends on the representativeness of that data. Second, the present implementation takes into account a small set of the possible subsystem interactions and fails to fully account for strongly coupled cross-subsystem dynamics. Third, while results of the CPU and memory presented in Table 3 are acceptable for embedded feasibility, further optimization may be required for ultra-low-power CubeSat configurations. Finally, the validation was done in a hardware in the loop environment, not in real in-orbit telemetry, which means that some further verification of the flight level is still needed.

Future study should work on expanding the diagnostic framework to more complex subsystem interactions, increasing generalization to telemetry patterns not seen previously and of validating the approach with actual mission telemetry. Additional work is also needed to pursue adaptive onboard update strategies, as well as more advanced decision fusion mechanisms as well, while not compromising the deterministic operational safety.

7. Conclusion

1. A conceptual architecture of an on-board hybrid fault-diagnostic system for CubeSat platforms was developed. Such an architecture involves synchronizing telemetry acqui-

sition, telemetry preprocessing, statistical feature extraction, deterministic FDIR branches, lightweight machine-learning branch, and decision fusion into one diagnostic cycle of an embedded system.

2. The experimental analysis revealed that the suggested hybrid fault-diagnosis architecture provided the most accurate fault detection performance (94% accuracy), outperforming classical FDIR (71%) and standalone machine learning approaches (88%). The discovery clearly proves that utilizing determinism and data-driven approach concurrently leads to better fault-detection capabilities.

3. The proposed hybrid fault-diagnostic architecture demonstrated the lowest latency in detecting faults – only 83 ms compared with 120 ms in case of classical FDIR and 95 ms in machine learning alone. That means that deterministic and ML-based monitoring concurrently helps recognize faults much faster than other methods do.

4. The hybrid fault-diagnostic architecture was proven to be compatible with onboard computation, as its implementation entailed minimal CPU consumption (only 17.6%), while memory usage did not exceed 58 KB. Thus, the suggested architecture can provide the desired diagnostic performance within onboard computing resources.

5. The architecture suggested by the researcher turned out to have the best resilience towards telemetry measurement noise, with the highest level providing just 80% accuracy. In contrast, standalone machine learning and classical FDIR had 65% and 43%, respectively, meaning that hybridization improves the robustness.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing

This study has been/was/is funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP26197553 “Optimization of sensitivity of fiber optic sensors for monitoring deformations in building structures”.

Data availability

The data will be made available on reasonable request.

Use of artificial intelligence

ChatGPT (GPT-5.2 Thinking) was used to a limited extent for proofreading the English text, including grammar, spelling, and punctuation checks. The AI tool was used only in preparing the English-language text (e.g., in the Abstract, Introduction/Literature Review, Aim and Objectives, and Data Availability sections), specifically for proofreading and correcting grammar, spelling, punctuation, and improving overall clarity and logical consistency. The AI tool was used only for language-related tasks:

1) proofreading the English text;

2) correcting grammar, spelling, punctuation, and minor sentence-level errors;

3) improving readability and logical consistency without changing the technical meaning;

4) checking consistency of terminology and notation across the English sections. The AI tool was not used to develop scientific ideas, perform theoretical analysis, design algorithms, run simulations, interpret results, or write the study conclusions.

The authors manually reviewed the AI-suggested edits and verified that the scientific meaning, terminology, formulas, notation, and references remained unchanged. They made corrections where necessary and finalized the manuscript through full author review.

The AI tool did not influence the scientific conclusions of the study. It was used only for English-language proofreading (grammar, spelling, and punctuation), while all analyses, results, and conclusions were produced and validated by the authors.

Authors’ contributions

Ainur Kuttybayeva: Writing – original draft, Methodology, Investigation, Conceptualization; **Samal Zhamalova**: Writing – original draft, Formal analysis, Writing – review & editing; **Anagul Boranbayeva**: Funding acquisition, Writing-original draft, Data curation. **Zhansaya Myrzayeva**: Supervision, Project administration, Methodology; **Gulnar Imasheva**: Methodology, Validation, Resources, Visualization; **Kaskatayev Zhanat**: Formal analysis, Resources, Funding acquisition; **Yersain Chinibayev**: Visualization, Visualization, Validation; **Nurzhamal Ospanova**: Funding acquisition, Writing – review & editing; **Mukhit Abdullayev**: Conceptualization, Mathematical modeling, Software, Simulation, Visualization, Writing – review & editing; **Kalmukhamed Tazhen**: Conceptualization, Methodology, Software, Visualization, Data curation, Writing – review & editing.

References

1. Isermann, R. (2006). Fault-Diagnosis Systems. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-30368-5
2. Patton, R. J., Frank, P. M., Clark, R. N. (Eds.) (2000). Issues of Fault Diagnosis for Dynamic Systems. Springer London. https://doi.org/10.1007/978-1-4471-3644-6
3. Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., Yin, K. (2003). A review of process fault detection and diagnosis. Computers & Chemical Engineering, 27 (3), 327–346. https://doi.org/10.1016/s0098-1354(02)00162-x
4. Oche, P. A., Ewa, G. A., Ibekwe, N. (2024). Applications and Challenges of Artificial Intelligence in Space Missions. IEEE Access, 12, 44481–44509. https://doi.org/10.1109/access.2021.3132500
5. Cratere, A., Gagliardi, L., Sanca, G. A., Golmar, F., Dell’Olio, F. (2024). On-Board Computer for CubeSats: State-of-the-Art and Future Trends. IEEE Access, 12, 99537–99569. https://doi.org/10.1109/access.2024.3428388

6. Obied, M. A., Ghaleb, F. F. M., Hassanien, A. E., Abdelfattah, A. M. H., Zakaria, W. (2023). Deep Clustering-Based Anomaly Detection and Health Monitoring for Satellite Telemetry. Big Data and Cognitive Computing, 7 (1), 39. https://doi.org/10.3390/bdcc7010039
7. Li, Z., Zhang, Y., Ai, J., Zhao, Y., Yu, Y., Dong, Y. (2023). A Lightweight and Explainable Data-Driven Scheme for Fault Detection of Aerospace Sensors. IEEE Transactions on Aerospace and Electronic Systems, 59 (6), 8392–8410. https://doi.org/10.1109/taes.2023.3303855
8. Fejjari, A., Delavault, A., Camilleri, R., Valentino, G. (2025). A Review of Anomaly Detection in Spacecraft Telemetry Data. Applied Sciences, 15 (10), 5653. https://doi.org/10.3390/app15105653
9. Xu, Z., Cheng, Z., Guo, B. (2023). A hybrid data-driven framework for satellite telemetry data anomaly detection. Acta Astronautica, 205, 281–294. https://doi.org/10.1016/j.actaastro.2023.02.009
10. Nalepa, J., Myller, M., Andrzejewski, J., Benecki, P., Piechaczek, S., Kostrzewa, D. (2022). Evaluating algorithms for anomaly detection in satellite telemetry data. Acta Astronautica, 198, 689–701. https://doi.org/10.1016/j.actaastro.2022.06.026
11. Shon, T., Moon, J. (2007). A hybrid machine learning approach to network anomaly detection. Information Sciences, 177 (18), 3799–3821. https://doi.org/10.1016/j.ins.2007.03.025
12. Hodge, V., Austin, J. (2004). A Survey of Outlier Detection Methodologies. Artificial Intelligence Review, 22 (2), 85–126. https://doi.org/10.1023/b:aire.0000045502.10941.a9
13. He, J., Cheng, Z., Guo, B. (2024). Anomaly detection in telemetry data using a jointly optimal one-class support vector machine with dictionary learning. Reliability Engineering & System Safety, 242, 109717. https://doi.org/10.1016/j.ress.2023.109717
14. Hedayati, M., Rahimi, A. (2025). A hybrid framework for real-time satellite fault diagnosis using Markov jump-adjusted models and 1D sliding window Residual Networks. Acta Astronautica, 228, 1066–1087. https://doi.org/10.1016/j.actaastro.2024.12.057
15. Crotti, E., Colagrossi, A. (2025). Machine Learning Approaches for Data-Driven Self-Diagnosis and Fault Detection in Spacecraft Systems. Applied Sciences, 15 (14), 7761. https://doi.org/10.3390/app15147761
16. Lai, Y., Zhu, Y., Li, L., Lan, Q., Zuo, Y. (2025). STGLR: A Spacecraft Anomaly Detection Method Based on Spatio-Temporal Graph Learning. Sensors, 25 (2), 310. https://doi.org/10.3390/s25020310
17. Liang, H., Liu, C., Liu, W., Li, W., Zhang, Y. (2025). Spacecraft Health Status Monitoring Method Based on Multidimensional Data Fusion. Machines, 13 (12), 1136. https://doi.org/10.3390/machines13121136
18. Liu, M., Xia, Q., Qiu, S. (2024). A new data-driven framework for progressive anomaly event alerts in spacecraft based on reconstruction discrepancy. Advances in Space Research, 74 (11), 5890–5905. https://doi.org/10.1016/j.asr.2024.08.054