European Space Agency Dataset and Benchmark for Real-World Anomaly Detection in Spacecraft Time Series

Krzysztof Kotowski[1,*] KKOTOWSKI@KPLABS.PL
Christoph Haskamp[2,*] CHRISTOPH.HASKAMP@AIRBUS.COM
Jacek Andrzejewski[1]
Bogdan Ruszczak[3,1]
Jakub Nalepa[4,1]
Daniel Lakey[5]
Peter Collins[6]
Jose Martínez-Heras[7]
Gabriele De Canio[6]

[1] *KP Labs, Bojkowska 37J, 44-100 Gliwice, Poland*
[2] *Airbus Defence and Space GmbH, Airbus-Allee 1, 28199 Bremen, Germany*
[3] *Opole University of Technology, Proszkowska 76, 45-758 Opole, Poland*
[4] *Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland*
[5] *CGI Deutschland B.V. & Co. KG, Mornewegstr. 30, 64295 Darmstadt, Germany*
[6] *European Space Operations Centre, European Space Agency, Robert-Bosch-Str. 5, 64293 Darmstadt, Germany*
[7] *Solenix GmbH, Spreestr. 3, 64295 Darmstadt, Germany*

[*] *Corresponding authors*

Reviewed on OpenReview: *https://openreview.net/forum?id=bbpRMoatVO*

Editor: Katharina Eggensperger

Abstract

Time series from spacecraft sensors are high-dimensional, nonstationary, nonlinear, irregularly sampled, and exhibit both spatial and temporal dependencies. Detecting anomalies in such signals is critical for both on-ground and in-orbit space operations. The potential of machine learning in this task is currently hampered by a lack of comprehensive datasets and benchmarks that capture its real-world complexity. The European Space Agency Anomaly Detection Benchmark (ESA-ADB) addresses this issue and establishes a new standard in the domain. It is a result of close cooperation between engineers from the European Space Operations Center and machine learning experts from industry and academia. Our newly introduced dataset (doi.org/10.5281/zenodo.12528695) contains several years of real-life raw data from 3 large spacecraft, including 224 channels, 821 control signals, and 1430 annotated events, which makes it the biggest dataset of its kind in the literature. The associated benchmark defines 9 specific requirements and 5 evaluation metrics for assessing anomaly detection algorithms in operational practice. The results indicate that widely used anomaly detection algorithms, even with our proposed adaptations, are not yet suitable

for effective deployment. Thus, ESA-ADB remains an open challenge and continues to be explored through dedicated competitions and related projects listed on the official project webpage (ai4gs.space-codev.org/esa-adb).

Keywords: aerospace engineering, anomaly detection, benchmark, dataset, satellite telemetry, time series

1 Introduction

Monitoring anomalies in time series data from spacecraft sensors (spacecraft telemetry) is a daily practice of thousands of spacecraft operations engineers (SOEs) in mission control centers worldwide. It ensures safe and uninterrupted operations of multiple scientific, communication, observation, and navigation satellites. SOEs are typically supported by simple automatic anomaly detection systems that alarm when a measurement falls outside its predefined nominal limits or when a measurement correlates with a known anomalous pattern (Martinez and Donati, 2018). However, more sophisticated anomalies are usually detected manually, which is a very expensive and error-prone task (Lutz and Mikulski, 2004). For this reason, all major space-related entities have been actively researching, developing, and testing advanced automatic anomaly detection systems in recent years, including space agencies from Europe (O'Meara et al., 2016; Evans et al., 2017; Fuertes et al., 2018; Ruszczak et al., 2025), USA (Hundman et al., 2018), Canada (Qian et al., 2024), Korea (Tariq et al., 2019), and Japan (Yairi et al., 2017), and multiple private companies (Pilastre et al., 2020; Zhang et al., 2019; Cuéllar et al., 2024). It is also a prioritized domain of the Artificial Intelligence for Automation Roadmap of the European Space Agency (ESA) (De Canio et al., 2023), and there is a growing trend in applying such systems directly onboard spacecraft for faster alarming and autonomous operations (Ruszczak et al., 2023b). However, spacecraft telemetry is an especially complex example of multivariate time series data of high dimensionality and volume (years of recordings from up to thousands of channels per spacecraft (Lakey and Schlippe, 2024)), complex characteristics (i.e., nonstationarity, nonlinearity, spatiotemporal dependencies, varying sampling frequencies, and data gaps), diverse data types (i.e., large variety and ranges of physical measures, categorical status flags, counters, and binary telecommands), and inherent noise related to the space environment.

Related work. There are hundreds of algorithms for time series anomaly detection (TSAD) proposed in the literature (158 according to Schmidl et al. (2022)) that could be viable solutions for spacecraft telemetry, but currently, the main challenge is the evaluation of different approaches. This occurs because there are relatively few anomalies in flying spacecraft and no comprehensive data collection from multiple sources. Thus, it is difficult to objectively conclude that one approach works better than the other. Moreover, multiple recent papers show that publicly available datasets, benchmarks, and metrics for spacecraft anomaly detection are flawed and cannot be used for an unbiased evaluation of emerging machine learning (ML) techniques, especially in complex real-world settings (Sehili and Zhang, 2023; Wagner et al., 2023; Wu and Keogh, 2023; Herrmann et al., 2024). Specifically, the most popular NASA SMAP and MSL datasets of spacecraft telemetry (Hundman et al., 2018) are too trivial and have unrealistic anomaly density, inconsistent ground truth, and run-to-failure bias (Wu and Keogh, 2023). There are a few generic TSAD benchmarks that avoid these flaws, but they are either univariate (Wu and Keogh, 2023), synthetic (Fleith,

2023), or do not represent complexities of real systems (varying sampling rates, different channel types, or real-time processing) (Liu and Paparrizos, 2024). See Appendix B.6 for detailed analysis of related datasets and benchmarks.

Contributions. The proposed **European Space Agency Anomaly Detection Benchmark (ESA-ADB)** directly addresses all the mentioned issues and establishes a new standard of validating algorithms for anomaly detection in real-world time series from spacecraft. It is a result of close cooperation between SOEs from the European Space Operations Center (ESOC) and ML experts from industry and academia. ESA-ADB consists of three main components (Figure 1):

1. **ESA Anomalies Dataset (ESA-AD)** – a large-scale, curated, and structured collection of real-world spacecraft telemetry data, collected from **3 ESA missions** and annotated by SOEs and ML experts.

2. **Evaluation pipeline** designed by ML experts for the practical needs of SOEs. It introduces a list of **9 requirements** and **5 metrics** designed for real-world spacecraft telemetry anomaly detection according to the latest advancements in TSAD. It simulates real operational scenarios, i.e., **5 different mission phases** and real-time monitoring.

3. **Baseline results for 8 TSAD algorithms**, filtered from the 71 available in the TimeEval framework (Wenig et al., 2022) and adapted to the complexities of real-world spacecraft telemetry.

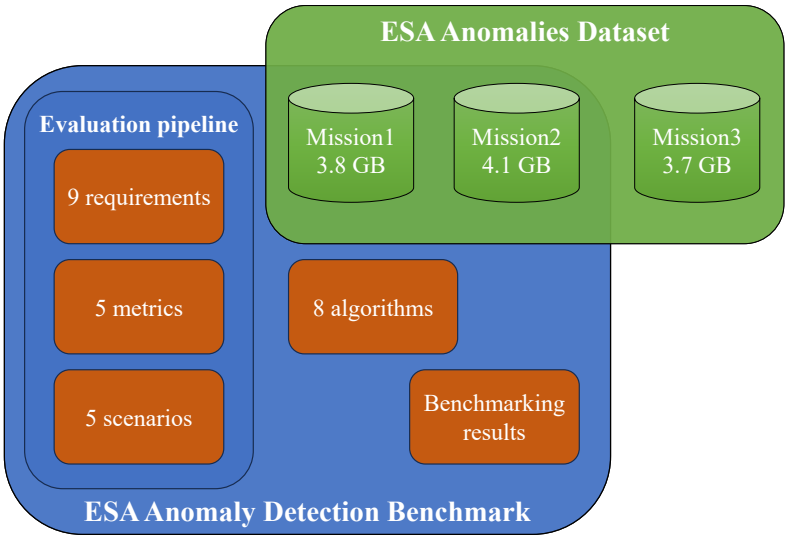

Figure 1: Main elements of the proposed dataset and benchmark.

The main goal of ESA-ADB is to allow researchers and practitioners to design and thoroughly assess if an algorithm could be applied as a support for SOEs in real-world

operational environments, taking into account all challenges of this complex time series data. To support that, we also launched a Kaggle competition based on a separate private test set (kaggle.com/competitions/esa-adb-challenge). ESA-ADB has been already downloaded more than 2,000 times and is actively used in several projects by ESA and its partners.

2 ESA Anomalies Dataset

The dataset is publicly available at doi.org/10.5281/zenodo.12528695. The nomenclature (e.g., channel types, telecommands, and event categories) is explained in the first section of the Appendix A.

2.1 Dataset collection and curation

Three missions (spacecraft) of different types (purposes, orbits, and launch dates) were selected by SOEs from the ESA portfolio based the presence of historical anomalies that are problematic to detect using existing out-of-limit approaches. The data selection was focused on collecting a large dataset with a possibly diverse spectrum of signals and anomalies as reported in the Anomaly Report Tracking System (artsops.esa.int) used at ESOC. Although each spacecraft collects thousands of telemetry signals (Appendix B), our dataset includes only limited subsets of channels and telecommands, identified by SOEs as essential for anomaly investigation. This selection was necessary to keep the annotation effort and overall dataset size at a manageable level. The data was initially annotated using the OXI annotation tool (oxi.kplabs.pl) (Ruszczak et al., 2023a) and the annotations were iteratively refined with assistance of unsupervised and semi-supervised algorithms, as a part of a structured process defined in Appendix B.3 and our previous related works (Kotowski et al., 2023, 2025).

Design choices. Our dataset has several features distinguishing it from other related datasets (Appendix B.6). It is intended to reflect the raw telemetry data accessible for SOEs, with all its pros and cons, volume, varying sampling rates, data gaps, and an overabundance of telecommands. It distinguishes events of different types, not only anomalies, i.e., rare nominal events, communication gaps, and invalid segments. Each channel is annotated independently, following the approach used in recent datasets such as SMD (Su et al., 2019), CATS (Fleith, 2023), and TELCO (González et al., 2023). This allows for evaluating not only whether an anomaly is detected, but also which specific channels are correctly identified as affected. Furthermore, a single annotated anomaly can consist of multiple non-contiguous segments, separated by periods of nominal behavior. For example, a series of short attitude disturbances caused by the same underlying issue is treated as one event (see Figure 2). This design choice avoids unfairly penalizing models for detecting each segment as a separate anomaly in the benchmark. The dataset was consistently structured to facilitate its usage in ML-based pipelines (Appendix B.5) and can be used with both supervised and unsupervised regimes.

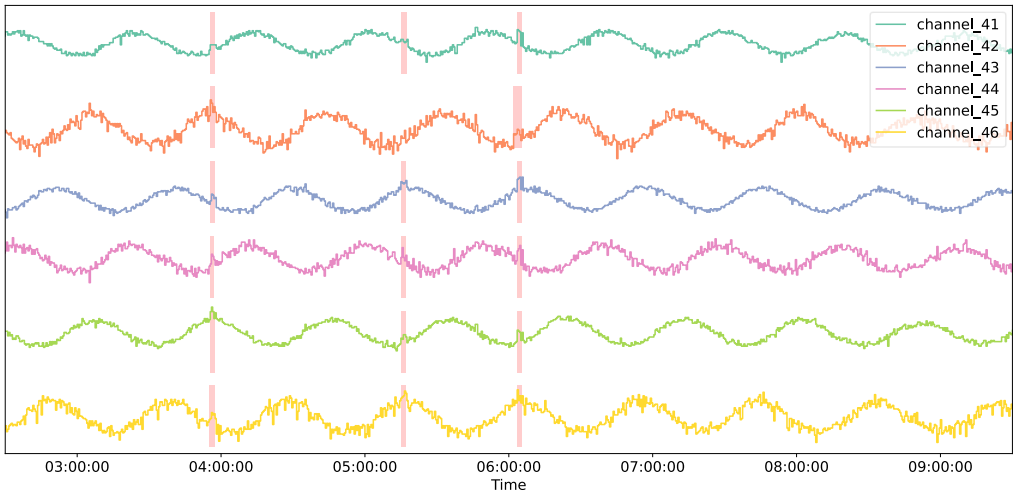

Figure 2: Example fragment with annotated event id_155 for a subset of 6 channels from Mission1 (highlighted with light red boxes).

Anonymization. Some information, such as mission and channel names, timelines, or units of measured values, are anonymized to avoid disclosing sensitive information. The anonymization does not affect the data integrity and it was verified that algorithms produce the same results as for the original data (Appendix B.4). It does prevent using physics-informed approaches or domain-specific knowledge to design algorithms (for example, to match telecommands and channels by names or to expect anomalies in specific times, e.g., during increased solar activity). However, it enforces the usage of universal data-driven approaches, instead of focusing on mission-specific intricacies.

2.2 Dataset content

The summary statistics of the dataset are presented in Table 1. The dataset contains 224 channels, 821 telecommands, and 1430 annotated events (including 157 anomalies) across 3 missions. Channels are categorized into target (monitored for anomalies) or non-target (auxiliary); and numerical (e.g., sensor measurements) or categorical (e.g., status flags and operating modes) ones. Channels originate from 6 common spacecraft subsystems and are clustered into groups of related channels (e.g., coming from similar sensors or showing similar characteristics). There are hundreds of different telecommands with millions of executions and they are grouped by SOEs according to the impact on the mission data (Appendix B.3) – from 0 (low impact) to 3 (high impact).

Missions differ significantly in aspects such as the proportion of categorical channels, the number of telecommands, and the distribution of event categories. They also vary in terms of signal characteristics and specific challenges posed for TSAD algorithms (Appendix B). However, they are all equally big (around 4GBs and 750M data points each) and the hundreds of annotated events constitute just a small fraction of the dataset ($< 2\%$), addressing the flaw of unrealistic anomaly density (Wu and Keogh, 2023).

Table 1: **Statistics of the ESA Anomalies Dataset.**

	Mission1	Mission2	Mission3	All
Channels	**76**	**100**	**48**	**224**
Target / Non-target	58 / 18	47 / 53	24 / 24	129 / 95
Numerical / Categorical	76 / 0	90 / 10	4 / 44	170 / 54
Channel groups	18	29	12	59
Subsystems	4	5	3	6*
Telecommands	**698**	**123**	**0**	**821**
Priority 0/1/2/3	345 / 323 / 19 / 11	0 / 0 / 119 / 4	0 / 0 / 0 / 0	345 / 323 / 138 / 15
Total executions	1,594,722	1,918,002	0	3,512,724
Data points	**774,856,895**	**776,734,364**	**744,530,898**	**2,296,122,157**
Duration (anonymized)	14 years	3.5 years	8 years	25.5 years
Compressed size [GB]	3.8	4.1	3.7	11.6
Annotated points [%]	1.80	0.58	1.03	1.14
Annotated events	**200**	**644**	**586**	**1,430**
Anomalies	118	31	8	157
Rare nominal events	78	613	25	716
Communication gaps	4	0	397	401
Invalid segments	0	0	156	156
Univariate / Multivariate	32 / 164	9 / 635	8 / 25	49 / 824
Global / Local	113 / 83	585 / 59	28 / 5	726 / 147
Point / Subsequence	12 / 184	0 / 644	3 / 30	15 / 858
Distinct event classes	**22**	**32**	**6**	**60**

*There are 3 matching subsystems between all missions.

Each anomaly and rare nominal event is described by three attributes corresponding to its dimensionality (uni-/multivariate), locality (local/global), and length (point/subsequence) according to the adjusted nomenclature of anomaly types by Blázquez-García et al. (2021). Most annotations are categorized as multivariate global subsequences, but there is also a diverse set of other types of anomalies (Appendix C.1), including some especially challenging ones (Appendix B.2). Additionally, events of similar characteristics are grouped into classes by SOEs, so it is easier to analyze results and design anomaly classifiers. The distributions of classes of events across missions' timelines are presented in Figure 3.

After initial experiments, we concluded that Mission3 is not suitable as a benchmarking dataset and should not be used for algorithm comparison (see Appendix B.1 for details). Nevertheless, we retain it as part of ESA-AD, as it is fully annotated and may still serve as a valuable resource for practitioners. It also provides an illustrative example demonstrating that not all real-world datasets are appropriate for benchmarking purposes.

2.3 Dataset split for the benchmark

Data for Missions 1 and 2 are split in half: the first half is used for training, and the second half for testing. This results in 84 months of training data for Mission1 and 21 months for Mission2. Within each training set, the last 3 months are reserved for validation. This validation window was chosen in agreement between ML experts and SOEs, as sufficiently long to assess algorithm performance under recent environmental conditions. Crucially, this temporal split ensures that no future information leaks into the training process, preserving the integrity of the evaluation. Anomalies appear in all sets, including training and validation ones. The default 50/50 split reflects mature phases of missions, where a substantial amount

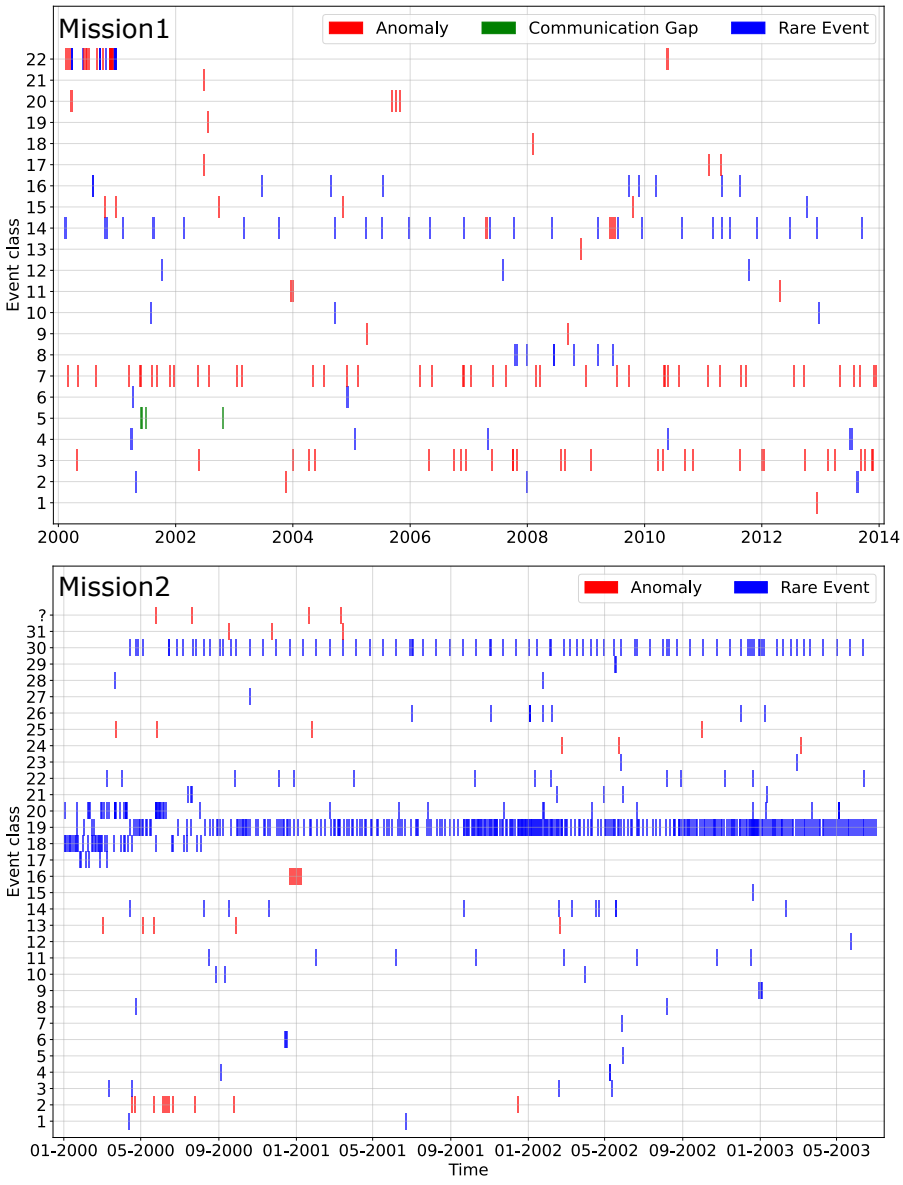

Figure 3: **Distributions of different classes (Y-axis) and categories (colors) of events across timelines of Mission1 (top) and Mission2 (bottom).** The bar width corresponds to the event length, but for better visualization, the minimum width was limited to 10 and 2.5 days for Mission1 and Mission2, respectively. The question mark represents anomalies of unknown class.

of telemetry data is already available for training. However, it is also important to deploy anomaly detection systems early in the mission life cycle. To address this, additional scenarios with shorter training periods are explored in Appendix D.6. These scenarios aim

to evaluate how well the algorithms perform under limited data conditions, assess their robustness to evolving mission environments, and determine the earliest point in a mission when reliable detectors can be trained.

Lightweight subsets of channels. In the default setting of ESA-ADB, all channels and the highest priority telecommands are used as input, and all target channels are used as output from algorithms. However, anomaly detection in tens of channels simultaneously is a challenging task, so for initial experiments, simpler models, and potential on-board applications, there are lightweight subsets of channels proposed in ESA-ADB. These are channels 41-46 for Mission1 and channels 18-28 for Mission2. These channels are commonly monitored by operators, are usually affected by anomalies (in 59% cases for Mission1 and 97% cases for Mission2), contain the most challenging anomalies to detect, and can be effectively analyzed in isolation. Channels from these subsets are presented in Figure 2 and Appendix D.4.

3 ESA Anomaly Detection Benchmark

The objective of the benchmark is to validate the performance of widely used TSAD algorithms on the ESA-AD dataset using the proposed evaluation procedures. The code is publicly available at github.com/kplabs-pl/ESA-ADB to ensure full reproducibility of the benchmark.

3.1 Algorithms' selection

There are several recent comprehensive reviews of TSAD approaches that list hundreds of ML algorithms (Schmidl et al., 2022; Wagner et al., 2023; Garg et al., 2022; Boniol et al., 2022; Li and Jung, 2023). Algorithms for our benchmark have been preselected based on the work by Schmidl et al. (2022) and its corresponding TimeEval framework (Wenig et al., 2022). We adapted it to our task due to the largest number of implemented algorithms (more than 70) among all considered frameworks (i.e., PyOD (Zhao et al., 2019), Luminaire (Chakraborty et al., 2020), TODS (Lai et al., 2022), Merlion (Bhatnagar et al., 2021), Sintel Orion (Alnegheimish et al., 2022), and TimeSeAD (Wagner et al., 2023)). This framework also includes the most widely used deep learning algorithm for anomaly detection in spacecraft telemetry – Telemanom by NASA (Hundman et al., 2018) – which we use as the primary baseline in our benchmark.

Functional requirements. To support the selection of algorithms, nine functional requirements (R1-R9) for anomaly detection algorithms in real-world space operations have been formulated by our team (Table 2) and evaluated against the capabilities of multivariate algorithms within the TimeEval framework (Appendix C.4.5). It turned out that no algorithm meets all the requirements. The widely used algorithms have problems especially with learning from known anomalies (R4), including auxiliary variables (R6), test-time learning (R7), irregular time series data (R8), and most importantly, handling large volumes of data without memory errors and time-outs (R9). Thus, additional work is necessary to adapt algorithms to our real-world use case.

Algorithms adaptation and final selection. To establish a simple baseline, five unsupervised algorithms based on traditional ML have been used without any special adaptation: Histogram-based Outlier Score (HBOS) (Goldstein and Dengel, 2012), k-nearest

Table 2: **Requirements for anomaly detection algorithms in real-world spacecraft telemetry.**

ID	Priority	Requirement	Description
R1	must	provide binary responses	Algorithms must define a clear threshold for triggering alarms. SOEs cannot rely solely on abstract anomaly scores. **This aspect is absolutely crucial in practical applications.**
R2	must	be able to model dependencies between channels	Algorithms are particularly needed to detect complex anomalies that only become apparent when analyzing multiple channels simultaneously.
R3	must	allow for online streaming anomaly detection	Algorithms in real-world settings must detect anomalies continuously without using future samples to generate predictions.
R4	should	learn from known anomalies in the training set	Algorithms should be able to leverage information about historical anomalies to effectively detect similar ones during inference.
R5	should	provide a list of affected channels	With thousands of channels in real missions, identifying affected channels would save SOEs significant effort and improve trust in the algorithm.
R6	should	support auxiliary variables in the input	Real-world data exists within a broader system of external variables (i.e., non-target channels) and control signals (i.e., telecommands). Algorithms should leverage this context to make better-informed decisions.
R7	should	allow for test-time learning from human feedback	In real-world settings, algorithms should be able to adapt based on feedback from domain experts – for example, to stop raising alarms for rare but known nominal events.
R8	should	natively handle irregular time series data	Varying sampling frequencies and data gaps are common in real-world time series. Standard resampling and interpolation methods make algorithms unaware of this fact and may lead to many incorrect detections.
R9	should	be able to run on a single high-end PC with a modern GPU (Appendix D.7)	ML algorithms are often run on a dedicated PC within the mission control room to minimize reliance on external systems and ensure data privacy, security, and integrity.

neighbors (KNN) (Ramaswamy et al., 2000), principal components classifiers (PCC) (Shyu et al., 2003), and isolation forest (iForest) (Liu et al., 2008) with its windowed version. They are not strictly time series algorithms, but they are close to what is currently used in real operations (i.e., semi-automatic out-of-limits and rule-based approaches). The more advanced semi-supervised LSTM-based Telemanom algorithm (Hundman et al., 2018) has been significantly adapted (Appendix C.4.1) to meet requirements R4, R6, and R9. The adapted version, called Telemanom-ESA, comes with pruned and non-pruned modes. Additionally, two methods have been added to the framework: a simple algorithm that mimics the classic out-of-limits approach by detecting values being more than N standard deviations away from the mean (Global STDN, described in Appendix C.4.2) and the Dilated Convolutional Variational AutoEncoder (DC-VAE) (González et al., 2023) with a series of adaptations (DC-VAE-ESA in Appendix C.4.3), including a thresholding based on N confidence intervals generated from VAE.

3.2 Real-world evaluation of unsupervised algorithms

Default evaluation procedures for unsupervised outlier detection algorithms in TimeEval (and many other frameworks, e.g., the popular PyOD (Zhao et al., 2019)) do not include any separate training step on a training set. The algorithms are run on all available data at once to look for outliers. This setup is unrealistic for real-world anomaly monitoring, where future data is not available at training time, and it introduces information leakage that gives unsupervised algorithms an unfair advantage in benchmarks. To address this, we modified the TimeEval framework so that each unsupervised algorithm is first initialized only on the training set – this includes calculating contamination levels, setting thresholds, and computing standardization parameters – and then applied to the test set.

3.3 Preprocessing

Our dataset contains raw telemetry in which channels have different, irregular sampling rates. There are no algorithms in the TimeEval framework that can handle such data without any preprocessing. Additionally, there are many different types of channels, so a consistent preprocessing is needed to run and compare all algorithms.

Resampling. Propagating the last known value (forward fill) is a widely recommended interpolation method for real-world sensor data (Ruszczak et al., 2023a; Liu et al., 2022). It is especially well suited for binary or quantized signals – such as status flags or measurements from analog-to-digital converters – because, unlike linear or Fourier-based interpolation, it avoids generating artificial or invalid intermediate values. Crucially, this method only uses past data, making it appropriate for real-time streaming applications where future samples are not yet available. Hence, this method was used for resampling (Appendix C.3).

Encoding telecommands. Telecommands in the original data are represented as lists of timestamps indicating when they were executed on board the spacecraft. For use in ESA-ADB as input channels to algorithms, telecommands are encoded as binary impulse signals – single-sample spikes aligned with the target resampling resolution.

Standardization. This step is essential for certain algorithms, such as KNN, and can also improve the performance of neural networks (Shanker et al., 1996). In our preprocessing, each channel is standardized independently to have zero mean and unit standard deviation,

based on the nominal (non-anomalous) points in the training set after resampling. Exceptions are constant and binary channels (e.g., encoded telecommands) that are just normalized to <0, 1> range. Monotonic channels (e.g., counters and cumulative readings) are differentiated and categorical channels (e.g., status flags) are enumerated before standardization (Appendix C.3).

3.4 Metrics and hierarchical evaluation

The selection of metrics and evaluation pipeline is a crucial step in establishing a reliable benchmark. Despite many years of research in the domain, there is no consensus on a reliable and unified set of TSAD metrics. Many recent advances criticize popular sample-wise and point-adjust protocols for being overoptimistic, and propose better alternatives (Sehili and Zhang, 2023; Wagner et al., 2023; Herrmann et al., 2024; Nalepa et al., 2022b; Huet et al., 2022; Sørbø and Ruocco, 2023; Garg et al., 2022; Hwang et al., 2022; Kim et al., 2022b,a; Paparrizos et al., 2022; Boniol et al., 2022; Abdulaal et al., 2021). Besides, there are several constraints on the selection of metrics arising directly from the functional requirements in Table 2. Metrics should operate on binary detections (R1), handle ground truth with irregular sampling (R8), and have reasonable computational complexity to handle large datasets (R9). Detailed analysis of all recent metrics in the context of our benchmark is presented in Appendix C.2.1.

First, SOEs identified and prioritized five most important aspects of anomaly detection in real-world mission operations. They are listed in Table 3 together with the proposed metrics to assess them. Importantly, each metric is designed to focus solely on a single specific aspect, in the maximum isolation from the other factors. There are several reasons for this: 1) to improve the interpretability of results by avoiding complex metrics combining multiple aspects at once, 2) to allow researchers from different domains to easily reorder or discard priorities, and 3) to enable the hierarchical evaluation approach in which algorithms are compared using one aspect at a time, from the highest to the lowest priority. This kind of evaluation has three important practical advantages: 1) it puts a strong emphasis on the priorities suggested by SOEs, 2) there is no need to select relative weights of specific aspects, and 3) it saves computational time by calculating only the necessary metrics.

The highest priority aspect relates to the proper identification of anomalous events with a strong emphasis on avoiding false alarms. This is because false positives are costly to resolve and deter operators from using the system. A high false positive rate is one of the main obstacles to the wider adoption of anomaly detection algorithms in space operations (Hundman et al., 2018). In the main text, we focus only on this most important aspect and the corresponding corrected event-wise metric, but other aspects are thoroughly described and assessed in Appendix C.2.3.

Corrected event-wise $F_{0.5}$-score. The event-wise scoring used for spacecraft telemetry by Hundman et al. (2018) is better than sample-wise approach in real-world scenarios as it 1) weighs all anomalies equally (not by their length) and 2) does not focus on the level of overlap between detections and the ground truth (in practice, it is enough to give an approximate location of the anomaly to human operators). However, the simple event-wise precision has one serious flaw – an algorithm that detects anomalies in every sample would have a perfect score (example in Appendix C.2). To mitigate this, we use the correction proposed by Sehili

Table 3: **Priority aspects and proposed metrics for assessing algorithms in ESA-ADB.**

Group	Aspect with priority level and brief description	Proposed metric
Primary	**1a.** *No false alarms* – minimize the number of false detections	Corrected event-wise $F_{0.5}$-score
	1b. *Anomaly existence* – maximize the number of correctly detected anomalies	
	2a. *Subsystems identification* – find a list of affected subsystems	Subsystem-aware $F_{0.5}$-score
	2b. *Channels identification* – find a list of affected channels	Channel-aware $F_{0.5}$-score
Secondary	**3.** *Exactly one detection per anomaly* – avoid multiple detections for the same annotated segment	Event-wise alarming precision
	4. *Detection timing* – determine the anomaly start time as precisely as possible	Anomaly detection timing quality curve (ADTQC)
	5. *Anomaly range and proximity* – find the exact duration of the anomaly and promote detections in close proximity to the ground truth	Modified affiliation-based $F_{0.5}$-score

and Zhang (Sehili and Zhang, 2023) that penalizes sample-wise (time-wise) false positives in the computation of the event-wise precision Pr_e (Equation 1):

$$\mathrm{Pr}_e = \frac{\mathrm{TP}_e}{\mathrm{TP}_e + \mathrm{FP}_e} \cdot (1 - \frac{\mathrm{FP}_t}{\mathrm{N}_t}) \quad , \tag{1}$$

where TP_e is the number of event-wise true positives, FP_e is the number of event-wise false positives, FP_t is the total duration of false positives, and N_t is the total duration of nominal signal. Based on that, the corrected event-wise F_β-score is defined by Equation 2:

$$\mathrm{F}_{\beta_e} = \frac{(1+\beta^2) \cdot \mathrm{Pr}_e \cdot \mathrm{Rec}_e}{\beta^2 \cdot \mathrm{Pr}_e + \mathrm{Rec}_e} \quad , \quad \mathrm{Rec}_e = \frac{\mathrm{TP}_e}{\mathrm{TP}_e + \mathrm{FN}_e} \quad , \tag{2}$$

where Rec_e is the event-wise recall and FN_e is the number of event-wise false negatives. We use β of 0.5 following Hundman et al. (2018) to additionally penalize false detections. The effect of β on the benchmark results is additionally analyzed in Appendix D.1.

3.5 Results and discussion

The goal of this benchmark is to provide a solid foundation for future research, rather than to identify the single best algorithm for real-world time series. Therefore, the experiments do not involve extensive hyperparameter tuning, which would be computationally prohibitive given the dataset size. Instead, we use default settings recommended by the original authors of each algorithm, with minor adjustments to match the specific characteristics of our dataset (Appendix C.4.6). This approach is intentional – it reflects typical TSAD practices and is meant to encourage the research community to build upon these results.

There are no algorithms in the TimeEval framework that can explicitly distinguish between anomalies and rare nominal events, so the results are presented for both types combined. However, separate results considering only anomalies are available in Appendix D.4. The corrected event-wise scores for Missions 1 and 2 are presented in Table 4. Results for

lower priority metrics are available in Appendix D. Scores are rounded to 3 significant digits to account for the inherent uncertainty of annotations in real-world data. The processing times of the algorithms are given in Appendix D.8.

Table 4: **Corrected event-wise scores for detection of anomalies and rare nominal events in lightweight and full sets of channels for Mission1 and Mission2.** Boldfaced results indicate the best values among all algorithms. OOM – out-of-memory.

Model	Mission1			Mission2		
	$\mathbf{Pr}_e$	$\mathbf{Rec}_e$	$\mathbf{F}_{0.5_e}$	$\mathbf{Pr}_e$	$\mathbf{Rec}_e$	$\mathbf{F}_{0.5_e}$
	*Trained and tested on **lightweight subsets of channels***					
PCC	<0.001	0.554	<0.001	0.029	**1.000**	0.036
HBOS	<0.001	0.585	<0.001	0.055	0.911	0.068
iForest	<0.001	0.585	<0.001	0.557	0.974	0.609
Windowed iForest	<0.001	0.738	<0.001	0.951	0.940	**0.949**
KNN	<0.001	0.754	<0.001	0.000	**1.000**	0.001
Global STD3	0.001	0.431	0.001	0.006	**1.000**	0.007
Global STD5	0.288	0.169	0.253	0.061	**1.000**	0.075
DC-VAE-ESA STD3	0.002	0.554	0.003	0.003	**1.000**	0.003
DC-VAE-ESA STD5	0.063	0.338	0.075	0.064	**1.000**	0.079
Telemanom-ESA	0.148	**0.894**	0.178	0.188	0.986	0.224
Telemanom-ESA Pruned	**0.999**	0.424	**0.786**	**0.978**	0.540	0.842
	*Trained and tested on **full set of channels***					
PCC	<0.001	0.870	<0.001	**0.064**	**0.997**	**0.079**
HBOS	<0.001	0.957	<0.001	0.016	0.820	0.020
iForest	<0.001	**0.967**	<0.001	0.022	0.903	0.027
Windowed iForest	OOM	OOM	OOM	0.034	0.746	0.042
KNN	OOM	OOM	OOM	OOM	OOM	OOM
Global STD3	<0.001	0.848	<0.001	0.006	**0.997**	0.008
Global STD5	0.002	0.761	0.003	0.052	0.992	0.064
DC-VAE-ESA STD3	<0.001	0.924	<0.001	0.002	**0.997**	0.002
DC-VAE-ESA STD5	0.005	0.804	0.007	0.008	0.904	0.011
Telemanom-ESA	0.007	0.946	0.008	0.052	0.992	0.064
Telemanom-ESA Pruned	**0.050**	0.870	**0.061**	0.058	0.964	0.071

Mission1. The pruned Telemanom-ESA has achieved the highest corrected event-wise $F_{0.5_e}$ and the lowest number of redundant alarms in both channel sets and all mission phases (Appendix D.6). The huge advantage of Telemanom in terms of these metrics is its dynamic thresholding scheme and additional pruning. This highlights the importance of proper thresholding and postprocessing methods in real-world settings. On the other hand, pruning significantly decreases subsystem-/channel-aware, detection timing (ADTQC), and affiliation-based scores, so the anomalies may be harder to identify. Unsupervised algorithms perform very poorly for Mission1 in terms of event-wise scores. DC-VAE-ESA and GlobalSTD are just slightly better which is especially disappointing for the former deep learning method. The main problem of these algorithms is a massive number of false detections caused by the noise and varying sampling rates in the data, as visible in the examples in Appendix D.3. However, DC-VAE-ESA has the best timing and affiliation-based scores – higher than Telemanom-ESA. This suggests that more advanced thresholding or postprocessing would significantly improve the event-wise scores of DC-VAE.

Mission2. Surprisingly, the simple windowed iForest and PCC algorithms turned out to be the best algorithms for the lightweight and full sets, respectively. It supports the claim to always consider simple algorithms as a baseline (Wu and Keogh, 2023; Rewicki et al., 2023). Windowed iForest achieved a very high corrected event-wise $F_{0.5_e}$ (0.949), ADTQC (0.985), and affiliation-based $F_{0.5_e}$ (0.959) scores. The main reason is the relative triviality of the lightweight subset of Mission2 which contains mainly rare nominal events characterized by significant sudden changes in the signal (Appendix B). However, the full set is much more challenging as reflected by much lower corrected event-wise scores. Moreover, metrics for anomalies alone (Appendix D.4) show that no algorithm was able to accurately identify all 9 actual anomalies in the overabundance of rare nominal events. This is one of the main practical challenges in many missions. Mission2 is also particularly problematic for Telemanom-ESA because of a lack of clear periodicity and many commanded events that are impossible to forecast.

Full sets vs. lightweight subsets. In most cases, the results for full sets of channels are much worse than for lightweight subsets. While the two are not directly comparable (since the lightweight test sets contain fewer annotated events), Appendix D.5 includes a direct comparison that supports this observation. This is one of the main challenges of high-dimensional real-world data – the more target channels there are, the higher the chance of false detections is. Additionally, due to the strong interconnections between channels, false detections frequently seep into many irrelevant channels.

4 Conclusions

ESA-ADB is a departure point for further development of better algorithms for anomaly detection in real-world time series (e.g., spacecraft telemetry). It was designed in close collaboration between ML and domain experts to fulfill the need for a reliable benchmark for both communities. Our goal was to ensure that improving the results of ESA-ADB does not just create an *illusion of progress* but solves real-world challenges in the TSAD domain – Appendix B.6 gives a summary of how ESA-ADB addresses common flaws listed by Wu and Keogh (Wu and Keogh, 2023). The requirements analysis and results show that our dataset poses a significant challenge for popular TSAD algorithms, and many changes had to be applied in the TimeEval framework (Wenig et al., 2022), training procedures, and algorithms to make them applicable to real-world data. While the results of Telemanom-ESA on subsets of channels may appear promising, the approach is highly parameterized, and the chosen thresholds may not generalize well to other missions. More importantly, the main challenge lies in scaling these algorithms to the full set of channels in our dataset – and to thousands of channels in real-world operations.

Limitations and future work. ESA-ADB has several limitations that we were not able to address in the scope of this study. Despite our best efforts, labeling inaccuracies are inevitable in such volumes of real-world data, so we are open to requests for corrections and plan to release updated versions of the dataset. Additionally, anonymization of physical units and timelines may constrain certain use cases. The dataset is still just a small fragment of real-world telemetry, but its complexity already poses a high entry barrier and requires some effort to fully understand. Due to the computational, functional (Table 2), and framework-related constraints, the current benchmark includes a limited range of algorithms and does

not involve extensive hyperparameter tuning. As such, ESA-ADB should be viewed not as a comprehensive benchmark of TSAD methods, but rather as a solid baseline for future research on real-world time series. Promising directions for extending this work include adapting Matrix Profile methods (Lu et al., 2023b) and transformer-based models with positional time encoding (Tuli et al., 2022; Xu et al., 2021; Drouin et al., 2022). Beyond TSAD, the dataset also holds potential for research in time series forecasting, telemetry data compression, continual learning, and foundation models.

Broader Impact Statement

We present a new large dataset for time series anomaly detection which is widely used in projects led by ESA and will inevitably have a broad impact on the domain. This should allow for training more complex models with greater relevance and more direct applicability to real-world time series anomaly detection problems. On the other hand, learning from larger quantities of data requires proportionally more computational time and resources. As such, it is important to always keep in mind the balance between computational expense and real-world relevance. There are also potential risks associated with the misuse of the data and improved methods in spacecraft cybersecurity, although we do not feel that there is a significant direct risk associated with this work.

Acknowledgments and Disclosure of Funding

This work was financially supported by the European Space Agency under the contract number 4000137682/22/D/SR. Authors thank all employees of ESOC involved in the project. Authors are grateful to Alicja Musiał, Szymon Rogoziński, and Dawid Lazaj from KP Labs for their valuable insights into the evaluation process.

References

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical Approach to Asynchronous Multivariate Time Series Anomaly Detection and Localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 2485–2494, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 978-1-4503-8332-5. doi: 10.1145/3447548.3467174.

Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, CySWATER '17, pages 25–28, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 978-1-4503-4975-8. doi: 10.1145/3055366.3055375. URL `https://doi.org/10.1145/3055366.3055375`.

Sarah Alnegheimish, Dongyu Liu, Carles Sala, Laure Berti-Equille, and Kalyan Veeramachaneni. Sintel: A Machine Learning Framework to Extract Insights from Signals. In *Proceedings of the 2022 International Conference on Management of Data*,

SIGMOD '22, pages 1855–1865, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 978-1-4503-9249-5. doi: 10.1145/3514221.3517910. URL https://doi.org/10.1145/3514221.3517910.

Mohammad Amin Maleki Sadr, Yeying Zhu, and Peng Hu. An Anomaly Detection Method for Satellites Using Monte Carlo Dropout. *IEEE Transactions on Aerospace and Electronic Systems*, 59(2):2044–2052, April 2023. ISSN 1557-9603. doi: 10.1109/TAES.2022.3206257. Conference Name: IEEE Transactions on Aerospace and Electronic Systems.

Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. Detecting Anomalies in Spacecraft Telemetry Using Evolutionary Thresholding and LSTMs. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 143–144, New York, NY, USA, 2021. ACM. ISBN 978-1-4503-8351-6.

Aadyot Bhatnagar, Paul Kassianik, Chenghao Liu, Tian Lan, Wenzhuo Yang, Rowan Cassius, Doyen Sahoo, Devansh Arpit, Sri Subramanian, Gerald Woo, Amrita Saha, Arun Kumar Jagota, Gokulakrishnan Gopalakrishnan, Manpreet Singh, K. C. Krithika, Sukumar Maddineni, Daeki Cho, Bo Zong, Yingbo Zhou, Caiming Xiong, Silvio Savarese, Steven Hoi, and Huan Wang. Merlion: A Machine Learning Library for Time Series, September 2021. URL http://arxiv.org/abs/2109.09265. arXiv:2109.09265 [cs, stat].

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys*, 54(3):56:1–56:33, April 2021. ISSN 0360-0300. doi: 10.1145/3444690.

Paul Boniol, John Paparrizos, Yuhao Kang, Themis Palpanas, Ruey S. Tsay, Aaron J. Elmore, and Michael J. Franklin. Theseus: navigating the labyrinth of time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(12):3702–3705, August 2022. ISSN 2150-8097. doi: 10.14778/3554821.3554879.

Paul Boniol, Ashwin K. Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, Michael J. Franklin, and John Paparrizos. VUS: effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal*, 34(3):32, May 2025. ISSN 1066-8888, 0949-877X. doi: 10.1007/s00778-025-00907-x. URL https://link.springer.com/10.1007/s00778-025-00907-x.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 978-1-58113-217-5. doi: 10.1145/342009.335388. URL https://doi.org/10.1145/342009.335388.

Teodora Sandra Buda, Haytham Assem, and Lei Xu. ADE: An ensemble approach for early Anomaly Detection. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 442–448, 2017.

Sayan Chakraborty, Smit Shah, Kiumars Soltani, Anna Swigart, Luyao Yang, and Kyle Buckingham. Building an Automated and Self-Aware Anomaly Detection System. In

2020 IEEE International Conference on Big Data (Big Data), pages 1465–1475, December 2020. doi: 10.1109/BigData50022.2020.9378177. URL https://ieeexplore.ieee.org/document/9378177.

Sara Cuéllar, Matilde Santos, Fernando Alonso, Ernesto Fabregas, and Gonzalo Farias. Explainable anomaly detection in spacecraft telemetry. *Engineering Applications of Artificial Intelligence*, 133:108083, July 2024. ISSN 0952-1976. doi: 10.1016/j.engappai.2024.108083.

Gabriele De Canio, James Eggleston, Jorge Fauste, Artur M. Palowski, and Mariella Spada. Development of an actionable AI roadmap for automating mission operations. In *2023 SpaceOps Conference*, Dubai, United Arab Emirates, March 2023. American Institute of Aeronautics and Astronautics. URL https://star.spaceops.org/user_manudownload.php?doc=303__bm05ydei.pdf.

Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. TACTiS: Transformer-Attentional Copulas for Time Series. In *Proceedings of the 39th International Conference on Machine Learning*, pages 5447–5493. PMLR, June 2022. URL https://proceedings.mlr.press/v162/drouin22a.html. ISSN: 2640-3498.

Mohamed El Amine Sehili and Zonghua Zhang. Multivariate Time Series Anomaly Detection: Fancy Algorithms and Flawed Evaluation Methodology. In Raghunath Nambiar and Meikel Poess, editors, *Performance Evaluation and Benchmarking*, pages 1–17, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-68031-1. doi: 10.1007/978-3-031-68031-1_1.

David Evans, José Martinez, Moritz Korte-Stapff, Attilio Brighenti, Chiara Brighenti, and Jacopo Biancat. Data Mining to Drastically Improve Spacecraft Telemetry Checking. In Craig Cruzen, Michael Schmidhuber, Young H. Lee, and Bangyeop Kim, editors, *Space Operations: Contributions from the Global Community*, pages 87–113. Springer International Publishing, Cham, 2017. ISBN 978-3-319-51941-8. doi: 10.1007/978-3-319-51941-8_4.

Bob Ferrell and Steven Santuro. NASA Shuttle Valve Data, 2005. URL https://cs.fit.edu/~pkc/nasa/data/.

Patrick Fleith. Controlled Anomalies Time Series (CATS) Dataset, February 2023. URL https://doi.org/10.5281/zenodo.7646897.

Sylvain Fuertes, Barbara Pilastre, and Stéphane D'Escrivan. Performance assessment of NOSTRADAMUS & other machine learning-based telemetry monitoring systems on a spacecraft anomalies database. In *2018 SpaceOps Conference*, SpaceOps Conferences. American Institute of Aeronautics and Astronautics, May 2018. doi: 10.2514/6.2018-2559. URL https://arc.aiaa.org/doi/10.2514/6.2018-2559.

Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2508–2517, June 2022. ISSN 2162-2388. doi: 10.1109/TNNLS.2021.3105827.

Markus Goldstein and Andreas Dengel. Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm. In *35th German Conference on Artificial Intelligence*, Saarbrucken, 2012.

G. García González, S. Martinez Tagliafico, A. Fernández, G. Gómez, J. Acuna, and P. Casas. One Model to Find Them All Deep Learning for Multivariate Time-Series Anomaly Detection in Mobile Network Data. *IEEE Transactions on Network and Service Management*, 2023. ISSN 1932-4537. doi: 10.1109/TNSM.2023.3340146. URL https://ieeexplore.ieee.org/abstract/document/10345720.

Erik W. Grafarend. *Linear and nonlinear models : fixed effects, random effects, and mixed models*. Berlin ; New York : Walter de Gruyter, 2006. ISBN 978-3-11-016216-5. URL http://archive.org/details/linearnonlinearm00wgra.

Niklas Heim and James E Avery. Adaptive anomaly detection in chaotic time series with a spatially aware echo state network. *arXiv preprint arXiv:1909.01709*, 2019.

Lars Herrmann, Marie Bieber, Wim J. C. Verhagen, Fabrice Cosson, and Bruno F. Santos. Unmasking overestimation: a re-evaluation of deep anomaly detection in spacecraft telemetry. *CEAS Space Journal*, February 2024. ISSN 1868-2510. doi: 10.1007/s12567-023-00529-5. URL https://doi.org/10.1007/s12567-023-00529-5.

Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local Evaluation of Time Series Anomaly Detection Algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 635–645, Washington DC USA, August 2022. ACM. ISBN 978-1-4503-9385-0. doi: 10.1145/3534678.3539339. URL https://dl.acm.org/doi/10.1145/3534678.3539339.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 387–395, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5552-0. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

Won-Seok Hwang, Jeong-Han Yun, Jonguk Kim, and Byung Gil Min. "Do you know existing accuracy metrics overrate time-series anomaly detections?". In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 403–412, Virtual Event, April 2022. ACM. ISBN 978-1-4503-8713-2. doi: 10.1145/3477314.3507024. URL https://dl.acm.org/doi/10.1145/3477314.3507024.

Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. Exathlon: a benchmark for explainable anomaly detection over time series. *Proceedings of the VLDB Endowment*, 14(11):2613–2626, October 2021. ISSN 2150-8097. doi: 10.14778/3476249.3476307. URL https://doi.org/10.14778/3476249.3476307.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

Ga-Yeong Kim, Su-Min Lim, and Ieck-Chae Euom. A Study on Performance Metrics for Anomaly Detection Based on Industrial Control System Operation Data. *Electronics*, 11(8):1213, January 2022a. ISSN 2079-9292. doi: 10.3390/electronics11081213. URL https://www.mdpi.com/2079-9292/11/8/1213.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a Rigorous Evaluation of Time-Series Anomaly Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7194–7201, June 2022b. ISSN 2374-3468. doi: 10.1609/aaai.v36i7.20680. URL https://ojs.aaai.org/index.php/AAAI/article/view/20680. Number: 7.

Krzysztof Kotowski, Christoph Haskamp, Bogdan Ruszczak, Jacek Andrzejewski, and Jakub Nalepa. Annotating Large Satellite Telemetry Dataset For ESA International AI Anomaly Detection Benchmark. In *Proceedings of the 2023 conference on Big Data from Space*, pages 341–344, Vienna, November 2023. Publications Office of the European Union. doi: 10.2760/46796. URL https://publications.jrc.ec.europa.eu/repository/handle/JRC135493.

Krzysztof Kotowski, Christoph Haskamp, Jacek Andrzejewski, Bogdan Ruszczak, Jakub Nalepa, Daniel Lakey, Peter Collins, Mauro Bartesaghi, Jose Martinez-Heras, and Gabriele De Canio. The Making of the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry. In *2025 SpaceOps Conference*, Montreal, Canada, 2025. Canadian Space Agency. URL https://star.spaceops.org/2025/user_manudownload.php?doc=152__57v8n7en.pdf.

Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, Virtual Event, January 2022. URL https://openreview.net/forum?id=r8IvOsnHchr.

Daniel Lakey and Tim Schlippe. A Comparison of Deep Learning Architectures for Spacecraft Anomaly Detection. In *2024 IEEE Aerospace Conference*, pages 1–11, March 2024. doi: 10.1109/AERO58975.2024.10521015. URL https://ieeexplore.ieee.org/document/10521015. ISSN: 1095-323X.

Alexander Lavin and Subutai Ahmad. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44, December 2015. doi: 10.1109/ICMLA.2015.141.

Gen Li and Jason J. Jung. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, 91:93–102, March 2023. ISSN 1566-2535. doi: 10.1016/j.inffus.2022.10.008. URL https://www.sciencedirect.com/science/article/pii/S1566253522001774.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. COPOD: Copula-Based Outlier Detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123, November 2020. doi: 10.1109/ICDM50108.2020.00135. ISSN: 2374-8486.

Dongyu Liu, Sarah Alnegheimish, Alexandra Zytek, and Kalyan Veeramachaneni. MTV: Visual Analytics for Detecting, Investigating, and Annotating Anomalies in Multivariate Time Series. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):103:1–103:30, April 2022. doi: 10.1145/3512950. URL https://doi.org/10.1145/3512950.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation Forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 413–422, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3502-9. doi: 10.1109/ICDM.2008.17. URL https://doi.org/10.1109/ICDM.2008.17.

Qinghua Liu and John Paparrizos. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c3f3c690b7a99fba16d0efd35cb83b2c-Abstract-Datasets_and_Benchmarks_Track.html.

Yue Lu, Makoto Imamura, Thirumalai Vinjamoor Akhil Srinivas, Eamonn Keogh, and Takaaki Nakamura. Matrix Profile XXX: MADRID: A Hyper-Anytime and Parameter-Free Algorithm to Find Time Series Anomalies of all Lengths. In *IEEE International Conference on Data Mining*, pages 1199–1204, Shanghai, December 2023a. doi: 10.1109/ICDM58522.2023.00148.

Yue Lu, Renjie Wu, Abdullah Mueen, Maria A. Zuluaga, and Eamonn Keogh. DAMP: accurate time series anomaly detection on trillions of datapoints and ultra-fast arriving data streams. *Data Mining and Knowledge Discovery*, 37(2):627–669, March 2023b. ISSN 1384-5810, 1573-756X. doi: 10.1007/s10618-022-00911-7. URL https://link.springer.com/10.1007/s10618-022-00911-7.

R.R. Lutz and I.C. Mikulski. Empirical analysis of safety-critical anomalies during operations. *IEEE Transactions on Software Engineering*, 30(3):172–180, March 2004. ISSN 1939-3520. doi: 10.1109/TSE.2004.1271171. URL https://ieeexplore.ieee.org/document/1271171. Conference Name: IEEE Transactions on Software Engineering.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 89–94, Bruges, 2015.

Jose Martinez and Alessandro Donati. Novelty Detection with Deep Learning. In *2018 SpaceOps Conference*, Marseille, May 2018. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-562-3. doi: 10.2514/6.2018-2560. URL https://arc.aiaa.org/doi/10.2514/6.2018-2560.

Aditya P. Mathur and Nils Ole Tippenhauer. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, April 2016. doi: 10.1109/CySWater.2016.7469060.

Jakub Nalepa, Jacek Andrzejewski, Bogdan Ruszczak, Krzysztof Kotowski, Alicja Musiał, Vladimir Zelenevskiy, Sam Bammens, and Rodrigo Laurinovics. Toward On-Board Detection Of Anomalous Events From Ops-Sat Telemetry Using Deep Learning. In *8th International Workshop On On-Board Payload Data Compression*, Athens, 2022a. Zenodo. doi: 10.5281/zenodo.724499. URL https://zenodo.org/record/7244991.

Jakub Nalepa, Michal Myller, Jacek Andrzejewski, Pawel Benecki, Szymon Piechaczek, and Daniel Kostrzewa. Evaluating algorithms for anomaly detection in satellite telemetry data. *Acta Astronautica*, June 2022b. ISSN 0094-5765. doi: 10.1016/j.actaastro.2022.06.026. URL https://www.sciencedirect.com/science/article/pii/S0094576522003162.

Corey O'Meara, Leonard Schlag, Luisa Faltenbacher, and Martin Wickler. ATHMoS: Automated Telemetry Health Monitoring System at GSOC using Outlier Detection and Supervised Machine Learning. In *SpaceOps 2016 Conference*, Daejeon, Korea, May 2016. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-426-8. doi: 10.2514/6.2016-2347. URL https://arc.aiaa.org/doi/10.2514/6.2016-2347.

Randy Paffenroth, Kathleen Kay, and Les Servi. Robust PCA for anomaly detection in cyber networks. *arXiv preprint arXiv:1801.01571*, 2018.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, July 2022. ISSN 2150-8097. doi: 10.14778/3551793.3551830. URL https://dl.acm.org/doi/10.14778/3551793.3551830.

Matej Petković, Luke Lucas, Jurica Levatić, Martin Breskvar, Tomaž Stepišnik, Ana Kostovska, Panče Panov, Aljaž Osojnik, Redouane Boumghar, José A. Martínez-Heras, James Godfrey, Alessandro Donati, Sašo Džeroski, Nikola Simidjievski, Bernard Ženko, and Dragi Kocev. Machine-learning ready data on the thermal power consumption of the Mars Express Spacecraft. *Scientific Data*, 9(1):229, May 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01336-z. URL https://www.nature.com/articles/s41597-022-01336-z. Publisher: Nature Publishing Group.

Barbara Pilastre, Loïc Boussouf, Stéphane D'Escrivan, and Jean-Yves Tourneret. Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning. *Signal Processing*, 168:107320, March 2020. ISSN 0165-1684. doi: 10.1016/j.sigpro.2019.107320. URL https://www.sciencedirect.com/science/article/pii/S0165168419303731.

Weicheng Qian, Joshua DeJong, and Bryan Cooke. Prompt Anomaly Detection for Small Satellites in Low-Earth Orbit Constellations: A Machine Learning Approach. *Small Satellite Conference*, August 2024. URL https://digitalcommons.usu.edu/smallsat/2024/all2024/219.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, 29(2):427–438, 2000. ISSN 0163-5808. doi: 10.1145/335191.335437. URL https://dl.acm.org/doi/10.1145/335191.335437.

Ferdinand Rewicki, Joachim Denzler, and Julia Niebling. Is It Worth It? Comparing Six Deep and Classical Methods for Unsupervised Anomaly Detection in Time Series. *Applied Sciences*, 13(3):1778, January 2023. ISSN 2076-3417. doi: 10.3390/app13031778. URL https://www.mdpi.com/2076-3417/13/3/1778. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

Bogdan Ruszczak, Krzysztof Kotowski, Jacek Andrzejewski, Christoph Haskamp, and Jakub Nalepa. OXI: An online tool for visualization and annotation of satellite time series data. *SoftwareX*, 23, July 2023a. ISSN 2352-7110. doi: 10.1016/j.softx.2023.101476. URL https://www.softxjournal.com/article/S2352-7110(23)00172-3/fulltext. Publisher: Elsevier.

Bogdan Ruszczak, Krzysztof Kotowski, Jacek Andrzejewski, Alicja Musiał, David Evans, Vladimir Zelenevskiy, Sam Bammens, Rodrigo Laurinovics, and Jakub Nalepa. Machine Learning Detects Anomalies in OPS-SAT Telemetry. In Jiří Mikyška, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M.A. Sloot, editors, *Computational Science – ICCS 2023*, Lecture Notes in Computer Science, pages 295–306, Cham, 2023b. Springer Nature Switzerland. ISBN 978-3-031-35995-8. doi: 10.1007/978-3-031-35995-8_21.

Bogdan Ruszczak, Krzysztof Kotowski, David Evans, and Jakub Nalepa. The OPS-SAT benchmark for detecting anomalies in satellite telemetry. *Scientific Data*, 12(1):710, April 2025. ISSN 2052-4463. doi: 10.1038/s41597-025-05035-3. URL https://www.nature.com/articles/s41597-025-05035-3. Publisher: Nature Publishing Group.

Mayu Sakurada and Takehisa Yairi. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, MLSDA'14, pages 4–11, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 978-1-4503-3159-3. doi: 10.1145/2689746.2689747. URL https://doi.org/10.1145/2689746.2689747.

Fernando Sanchez, Christopher Pankratz, Douglas M. Lindholm, Ransom Christofferson, Darren Osborne, and Thomas Baltzer. WebTCAD: A Tool for Ad-hoc Visualization and Analysis of Telemetry Data for Multiple Missions. In *2018 SpaceOps Conference*, Marseille, France, May 2018. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-562-3. doi: 10.2514/6.2018-2616. URL https://arc.aiaa.org/doi/10.2514/6.2018-2616.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, July 2022. ISSN 2150-8097. doi: 10.14778/3538598.3538602. URL https://doi.org/10.14778/3538598.3538602.

Mohamed El Amine Sehili and Zonghua Zhang. Multivariate Time Series Anomaly Detection: Fancy Algorithms and Flawed Evaluation Methodology. In *TPC Technology Conference on Performance Evaluation & Benchmarking*, Vancouver, November 2023. arXiv. doi: 10.48550/arXiv.2308.13068. URL http://arxiv.org/abs/2308.13068.

M. Shanker, M. Y. Hu, and M. S. Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, August 1996. ISSN 0305-0483. doi: 10.1016/0305-0483(96)00010-2. URL https://www.sciencedirect.com/science/article/pii/0305048396000102.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering, 2003.

Nidhi Singh and Craig Olinsky. Demystifying Numenta anomaly benchmark. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1570–1577, May 2017. doi: 10.1109/IJCNN.2017.7966038. ISSN: 2161-4407.

Hongchao Song, Zhuqing Jiang, Aidong Men, and Bo Yang. A Hybrid Semi-Supervised Anomaly Detection Model for High-Dimensional Data. *Computational Intelligence and Neuroscience*, 2017(1):8501683, 2017. ISSN 1687-5273. doi: 10.1155/2017/8501683. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2017/8501683. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2017/8501683.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2828–2837, New York, NY, USA, July 2019. Association for Computing Machinery. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Sondre Sørbø and Massimiliano Ruocco. Navigating the metric maze: a taxonomy of evaluation metrics for anomaly detection in time series. *Data Mining and Knowledge Discovery*, November 2023. ISSN 1573-756X. doi: 10.1007/s10618-023-00988-8. URL https://doi.org/10.1007/s10618-023-00988-8.

Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S. Woo. Detecting Anomalies in Space using Multivariate Convolutional LSTM with Mixtures of Probabilistic PCA. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2123–2133, New York, NY, USA, July 2019. Association for Computing Machinery. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330776. URL https://doi.org/10.1145/3292500.3330776.

Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. Precision and Recall for Time Series. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/2018/hash/8f468c873a32bb0619eaeb2050ba45d1-Abstract.html.

Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, February 2022. doi: 10.14778/3514061.3514067. URL https://doi.org/10.14778/3514061.3514067.

Dennis Wagner, Tobias Michels, Florian C. F. Schulz, Arjun Nair, Maja Rudolph, and Marius Kloft. TimeSeAD: Benchmarking Deep Multivariate Time-Series Anomaly Detection. *Transactions on Machine Learning Research*, April 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iMmsCI0JsS.

Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. TimeEval: a benchmarking toolkit for time series anomaly detection algorithms. *Proceedings of the VLDB Endowment*, 15(12):3678–3681, September 2022. ISSN 2150-8097. doi: 10.14778/3554821.3554873. URL https://doi.org/10.14778/3554821.3554873.

Renjie Wu and Eamonn Keogh. Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2421–2429, March 2023. ISSN 1041-4347, 1558-2191, 2326-3865. doi: 10.1109/TKDE.2021.3112126. URL https://ieeexplore.ieee.org/document/9537291/.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and Honglin Qiao. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 187–196, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5639-8. doi: 10.1145/3178876.3185996. URL https://doi.org/10.1145/3178876.3185996.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. October 2021. doi: 10.48550/arXiv.2110.02642. URL https://arxiv.org/abs/2110.02642v5.

Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. Fault detection by mining association rules from house-keeping data. volume 18, page 21, Quebec, 2001. URL https://robotics.estec.esa.int/i-SAIRAS/isairas2001/papers/Paper_AS012.pdf.

Takehisa Yairi, Naoya Takeishi, Tetsuo Oda, Yuta Nakajima, Naoki Nishimura, and Noboru Takata. A Data-Driven Health Monitoring Method for Satellite Housekeeping Data Based on Probabilistic Clustering and Dimensionality Reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 53(3):1384–1401, June 2017. ISSN 1557-9603. doi: 10.1109/TAES.2017.2671247. Conference Name: IEEE Transactions on Aerospace and Electronic Systems.

Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 565–574, New Orleans, LA, November 2017. IEEE. ISBN 978-1-5386-3835-4. doi: 10.1109/ICDM.2017.66. URL http://ieeexplore.ieee.org/document/8215529/.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In

Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19, pages 1409–1416, Honolulu, Hawaii, USA, January 2019. AAAI Press. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33011409. URL https://doi.org/10.1609/aaai.v33i01.33011409.

Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate Time-series Anomaly Detection via Graph Attention Network, September 2020. URL http://arxiv.org/abs/2009.02040. Number: arXiv:2009.02040 arXiv:2009.02040 [cs, stat].

Yue Zhao, Zain Nasrullah, and Zheng Li. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. URL http://jmlr.org/papers/v20/19-011.html.

Appendix A. Definitions

A.1 Channel vs parameter

Satellite telemetry consists of multiple time series that are called *parameters* by SOEs. This name is very problematic from the ML point of view because it collides with its fundamental nomenclature in which the *parameter* already has a couple of different meanings:

- a *parameter* of the model that is updated during the training, i.e., a single weight of the neural network;
- a *parameter* (or hyperparameter) of the algorithm which controls its behavior;
- a *parameter* of a statistical test (e.g., mean or variance of the estimated Gaussian distribution).

Hence, the *parameter* was replaced with the *channel* for purposes of ESA-ADB to avoid potential nomenclature collisions. *Channels* represent measurements from different sensors, status flags, and payload-related information. Each channel contains a list of samples defined by pairs of timestamps and signal values.

A.2 Subsystems

Satellites are typically composed of multiple specialized parts (subsystems) including propulsion, electrical power, thermal control, attitude and orbit control, communication, and data handling subsystems. There are also unique satellite-specific subsystems in some missions. A subsystem gathers all components (and channels) responsible for a specific function.

A.3 Telecommands

Telecommands (TCs) are sent from the Earth to the satellite in order to control different aspects of its operation. There are hundreds or thousands of different TCs for each mission

with millions of total executions, affecting different subsystems and specific components. Many different TCs are frequently executed simultaneously or in series to perform specific instructions. They may affect the observed telemetry in various ways, from no visible changes to strong disruptions. In our dataset, each TC is a binary signal with values of 1 in the exact timestamps of TC's executions on-board the satellite. TCs are not expected to contain any anomalies and even if they were, anomalies (e.g., missing TCs) would be impossible to identify automatically without additional expert knowledge and information about mission plans. Thus, they are not monitored nor annotated for anomalies.

A.4 Target and non-target channels

Not every channel can be a target for anomaly detection benchmarking. Like telecommands, some channels are not expected to contain any anomalies, and it would be impossible to annotate them without additional external data anyway. Examples include status flags, counters, and metadata, such as location coordinates. They often contain important information in the context of anomaly detection but are not monitored nor annotated for anomalies. They may contain outliers that are, however, irrelevant (or nominal) for SOEs. They are called *non-target* channels in ESA-ADB. This aspect is usually not considered in existing multivariate anomaly detection datasets and benchmarks. The selection of *target* and *non-target* channels is somewhat subjective and it may turn out that some algorithms would be able to properly handle some *non-target* channels by discovering some unknown relationships in the data. However, the metrics in ESA-ADB are calculated only for *target* channels. *Non-target* channels may and should be used as input features for algorithms.

A.5 Event class vs category vs type

Each annotated event can be assigned to a different class, category, and type:

- *event classes* relate to main causes of events and their specific variations (subclasses) as identified by SOEs. For example, attitude disturbances (with subclasses depending on the specific cause), resets, power drops, latch-ups, solar flares, etc.;
- *event categories* relate to the categorization of events from the operational point of view, i.e., anomalies, rare nominal events, communication gaps, and invalid segments, as described in the next section;
- *event types* relate to the signal characteristics of events according to the taxonomy introduced in Appendix A.7.

Note that each feature is independent of others, that is, events of the same class can have different categories and types, e.g., resets caused by telecommands are categorized as rare nominal events, but unexpected non-commanded resets are categorized as anomalies.

A.6 Event categories

For the purposes of our project, 4 categories of events are introduced: anomalies, rare nominal events, communication gaps, and invalid segments. They are defined in Table 5. The main reason was to distinguish atypical changes in the telemetry that should not

be alarmed to operators (rare nominal events, communication gaps, and invalid segments) from unexpected ones that should be alarmed (anomalies). Rare nominal events are not anomalies from the operators' point of view and they are usually not reported in anomaly tracking systems. Eventually, they are recorded in the mission log as special operations. For some missions, i.e., Mission2, there is a significant number of such operations causing (not so) rare events. Hence, the ideal algorithm should not alarm for rare nominal events, but it is usually impossible to distinguish between novel rare nominal events and anomalies without additional a priori expert knowledge. As agreed with SOEs, it would be acceptable if an anomaly detection system shows a false alarm for the first occurrence of the specific rare event, but it should not alarm for any subsequent occurrences of similar rare events. In machine learning, we can define that problem as active one-shot learning. To enable evaluation in such a scenario using ESA-AD, it is necessary to distinguish rare events from anomalies in ground truth annotations. Besides, such a division allows us to calculate separate performance metrics for rare events and "real" anomalies. It also helps to interpret the results in case of false negative or false positive detections for rare events.

A.7 Event types

To the best of our knowledge, the taxonomy by Blázquez-García et al. (2021) is the only one in the literature that comprehensively defines multivariate anomaly types, and our definitions are built based on this foundation. It divides event types into point and subsequence ones, where point events are defined as single outlying data points. However, this definition does not take into account varying sampling rates for which the length of "a single data point" may differ in time. Thus, for our purposes, multi-instance point anomalies are allowed if they are relatively short fragments of the signal that resemble points or peaks (i.e., up to 3 samples) when inspected using a typical sampling frequency for the channel. Both point and subsequence anomalies may be univariate or multivariate depending on whether they affect one or more channels. Anomalies can additionally be divided into global and local (contextual) ones, similarly as proposed in behavior-driven taxonomy by Lai et al. (2022). To make the original definitions more specific in our taxonomy, the global subsequence anomaly is defined as a subsequence of anomalous values in which at least one instance can be treated as a global point anomaly.

In the proposed taxonomy, each anomaly type can be described by three attributes: ***dimensionality*** (uni-/multi-variate), ***locality*** (local/global), and ***length*** (point/subsequence), as presented in Fig. 11. These attributes can be automatically inferred from per-channel annotations:

1. **Dimensionality** can be inferred by counting the number of channels affected by an anomaly. One affected channel makes it *univariate* and more affected channels make it *multivariate*.

2. To infer **locality**, we calculate the minimum and maximum values of all nominal samples in the dataset for each channel. If any sample of an annotated event lays out of <min, max> range for any channel, we mark it as *global*, otherwise it is *local*. This approach is a bit simplistic taking into account severe distribution shifts and different nominal levels of the signal in some missions, but it should be enough to identify *global*

anomalies which could be detected with an out-of-distribution approach from more challenging *local* anomalies.

3. In terms of **length**, considering non-uniform sampling rates and the differences between mission and channels, it is hard to give a strict definition of a *point* anomaly. Our proposition is to make it dependent on the dominant sampling frequency for each mission (0.033 Hz for Mission1, 0.056 Hz for Mission2 and 0.065 Hz for Mission3). A point anomaly is defined as a sequence of up to 3 samples after signal resampling to the dominant sampling frequency. Importantly, some anomalies are fragmented into several non-overlapping annotated regions. In this case, we treat each region separately, so even if an anomaly contains several regions it can be a point anomaly if all of these regions are categorized as point anomalies.

Such automatically inferred attributes for every anomaly and rare event are given in anomaly_types.csv for each mission, taking into account annotations for all channels. However, when working with subsets of channels, only the specific subset of channels should be considered to infer anomaly types. For this purpose, the script infer_anomaly_types.py is available in the code repository. The attributes are not inferred for communication gaps and invalid fragments.

Table 5: Definitions of event categories.

Event category	**Definition**	**Typical examples**	**Alarming**
Anomaly	Atypical, rare, unplanned, and unwanted change in the telemetry.	Micrometeorite impacts, solar flares, hardware or software failures, latch-ups, unexpected responses to telecommands	Every occurrence should be alarmed.
Rare nominal event	Atypical and rare but expected or planned change in the telemetry. It can be triggered by known telecommands (commanded rare event) or by any other non-commanded special event in the mission timeline.	*Commanded*: maneuvers, resets, calibrations, switching devices on/off *Non-commanded*: planned autonomous operations, eclipses, lunar transitions	Only the first occurrence of a rare nominal event from each class may be alarmed. Subsequent occurrences should not be alarmed.
Communication gap	Unusually long gap in the telemetry (missing data in some or all channels) not directly related to known anomalies.	Problems with the ground infrastructure, effects of resets	It should not be alarmed unless explicitly stated to do so.
Invalid segment	Fragment of telemetry data containing invalid or forbidden values not directly related to known anomalies. It is neither nominal nor anomalous.	Telemetry does not meet clearly defined validity rules of the mission.	It should not be alarmed unless explicitly stated to do so.

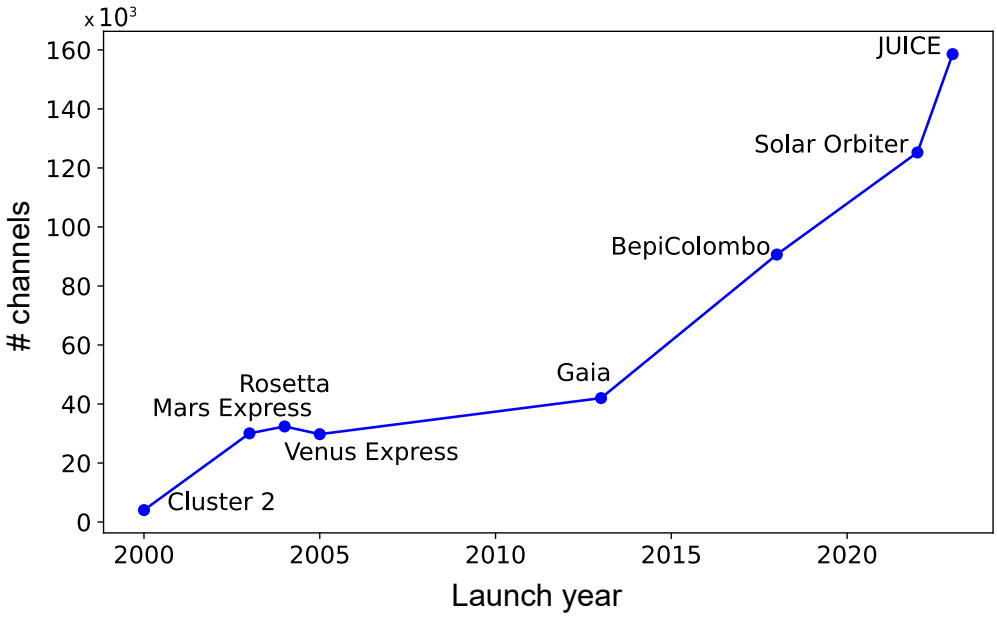

Figure 4: Increasing complexity of selected ESA spacecrafts over time (Lakey and Schlippe, 2024)

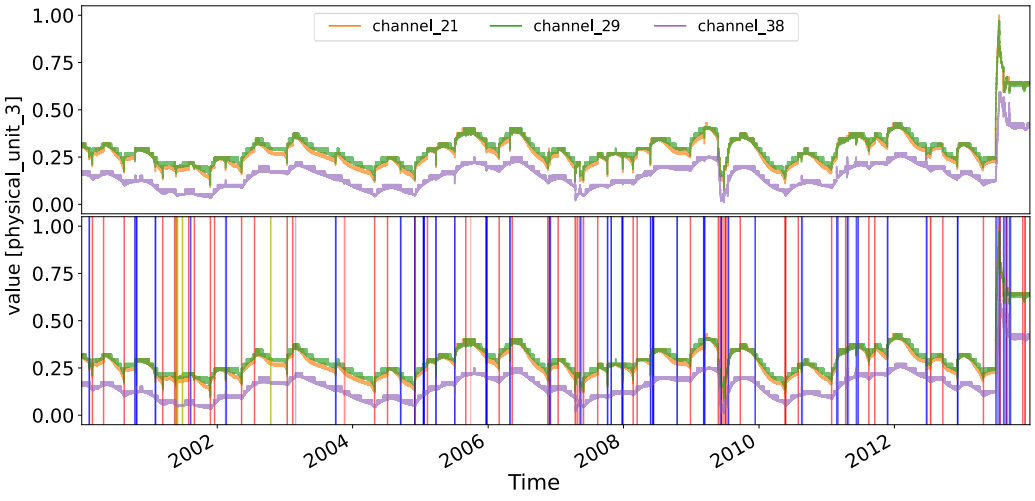

Figure 5: Overview of 3 channels from group 4 in Mission1 without annotations (top panel) and annotated (bottom panel). Blue, yellow, and red vertical bars are rare nominal events, communication gaps, and anomalies, respectively.

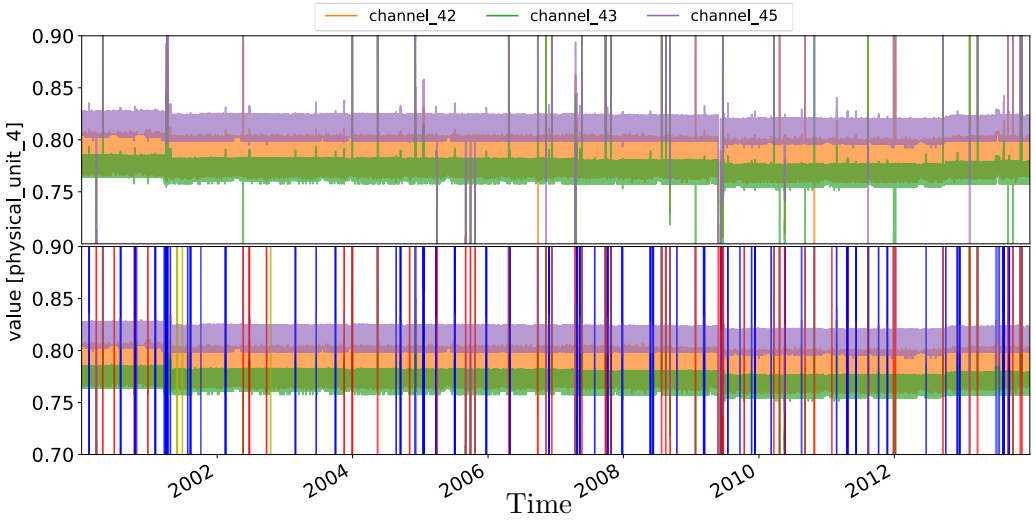

Figure 6: Overview of 3 channels from group 8 in Mission1 without annotations (top panel) and annotated (bottom panel).. Blue, yellow, and red vertical bars are rare nominal events, communication gaps, and anomalies, respectively. The close-up of these channels is presented in in the main text.

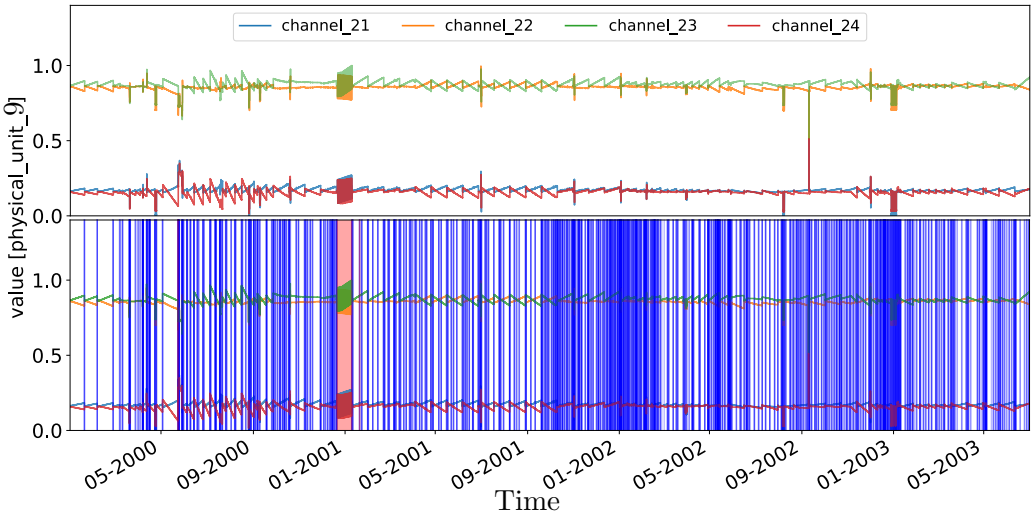

Figure 7: Overview of 4 channels from group 2 in Mission2 without annotations (top panel) and annotated (bottom panel). Blue and red vertical bars are rare nominal events and anomalies, respectively.

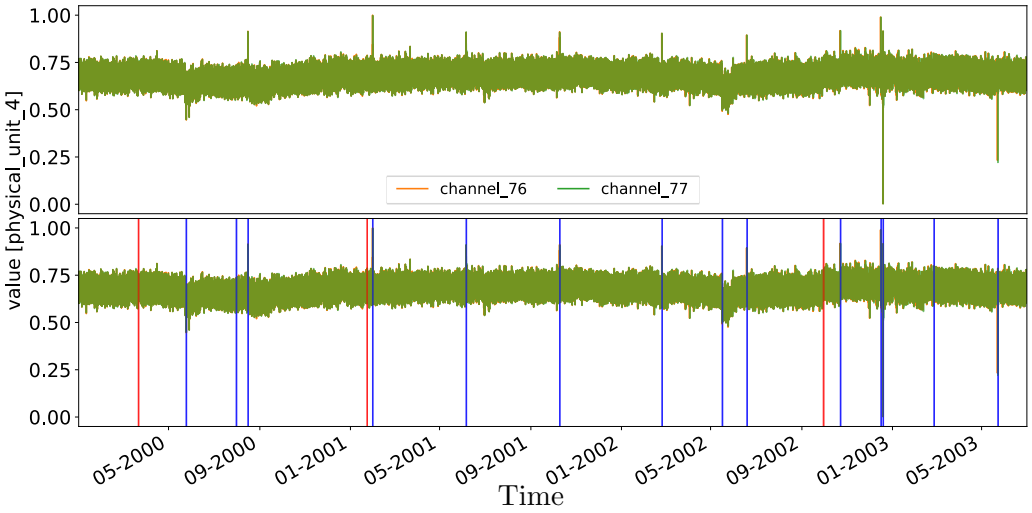

Figure 8: Overview of 2 channels from group 31 in Mission2 without annotations (top panel) and annotated (bottom panel). Blue and red vertical bars are rare nominal events and anomalies, respectively. Note that channels have very similar values, so it is hard to distinguish them.

Table 6: Main challenges posed for algorithms by missions in ESA-ADB

Mission	Main challenges for algorithms
1	• Some anomalies are hard to spot (see Table 7) • Some huge outliers (usually related to rare nominal events) • Low signal-to-noise ratio in channels from group 8 • Monotonically non-decreasing signals in channels from group 2 • Last 18 months include a severe concept drift in channels from groups 4, 7, and 13 • There is a visible seasonality with a very long period length • Overabundance of telecommands
2	• Some anomalies are hard to spot when looking at individual channels only (see Table 7) • Overabundance of rare nominal events and a very small number of anomalies • No obvious periodicity of the signal • Monotonically non-decreasing signals in channels from group 20 • Many categorical and non-target channels

Appendix B. ESA Anomalies Dataset

B.1 Mission3

Mission3 is a part of ESA-AD but had to be omitted in the benchmarking part after initial experiments. It has many communication gaps (see Figure 9), invalid segments (corrupted data), long periods of constant signals, lack of telecommands, and a small number of anomalies that are trivial to detect according to Definition 1 by Wu and Keogh (2023). We decided that it is not a good benchmarking dataset but it should still be made available to the community as it is fully annotated and may still be an interesting resource for practitioners in the domain. It is also retained as an illustrative example demonstrating that not all real-world datasets are suitable for use as benchmarks.

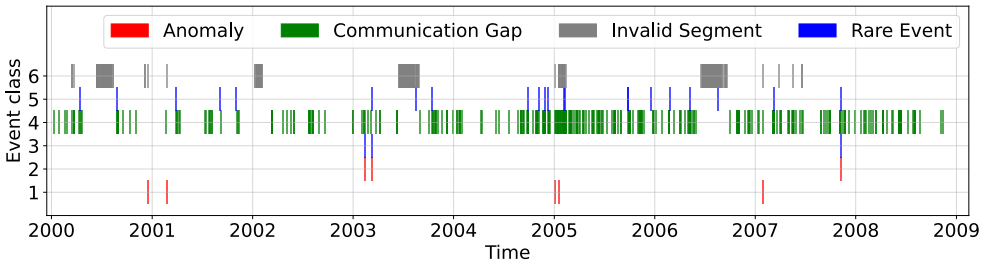

Figure 9: Distributions of classes of annotated events across the timeline of Mission3.

B.2 Examples of challenging events to detect

As mentioned in the main text, the initial selection of missions was based on the presence of challenging anomalies according to SOEs. To support the analysis of results, a list of selected events of this type in test sets of ESA-ADB is provided in Table 7. It is not a complete list. It is limited to test sets and includes only subjectively selected examples among many others. Example detections by semi-supervised algorithms trained on full (suffix "-*Full*") and lightweight (suffix "-*Light*") subsets for selected events are presented in Appendix D.3 as a series of figures referenced in Table 7.

Other interesting examples include events from classes 2, 14, 15, and 22 in Mission1 where similar changes in the same channel are sometimes categorized as anomalies and sometimes as rare nominal events, depending on the presence of TCs. A similar case for Mission2 is visualized in Fig. 10 for the non-commanded anomaly id_618 and the commanded rare event id_609. One of the important future works is to design algorithms that would be able to distinguish between such cases.

There are also some interesting nominal fragments related to atypical changes in sampling frequency in Mission1. There are 3 main examples of such behavior in the training set on days 2001-05-28, 2001-05-31, and 2001-06-27 where rapidly changing sampling rate causes small atypical "gaps" in data for channels 41-46. In the refinement process, it was observed that those gaps are detected as anomalies by many algorithms. However, we decided that they should not be annotated, because varying sampling rates are expected in satellite

telemetry and these false detections are mainly since the selected algorithms are not aware of frequency changes.

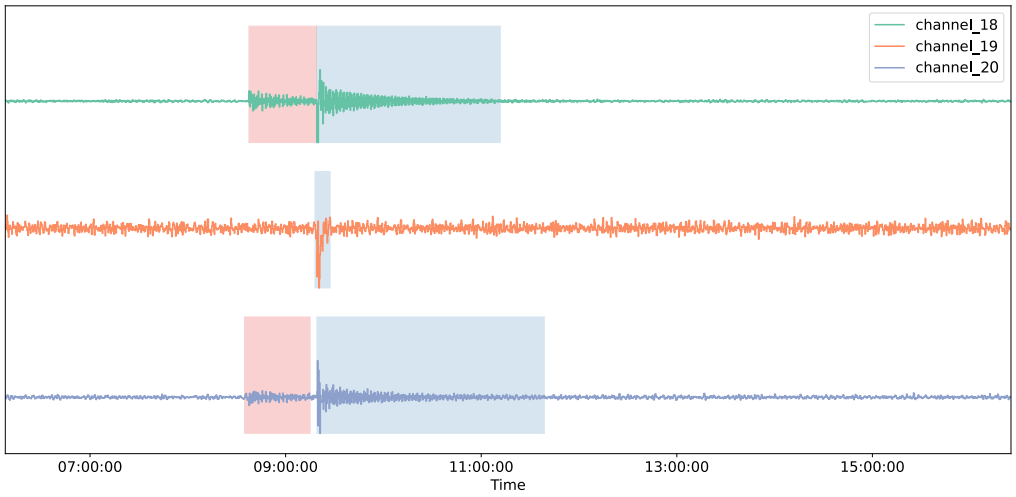

Figure 10: Annotated anomaly id_618 (marked in red) directly preceding the commanded rare nominal event id_609 (marked in blue) in Mission2. Importantly, the annotation at channel_20 ends just before the rare event, because, unlike for channel_18, the signal goes back to the previous level for a moment. The Y-axis is omitted, as channels are normalized and vertically shifted for improved visualization.

B.3 Annotation details

While annotating, a special focus was put on the precise identification of anomaly starting points for all channels. On the other hand, anomaly end times may be less accurate, because they are much harder to identify objectively, especially for long anomalies. Importantly, ARTS reports are intended for human use and are not well-suited for ML purposes. They usually include only approximate time ranges and a small fraction of affected channels. Moreover, well-known anomalies and rare nominal events are often not reported. Thus, the whole signal was carefully revisited by the ML team in search of any suspicious events. An initial list of subsystems, channels, and telecommands relevant for anomaly detection was proposed by SOEs, but was gradually extended during several iterations of the annotation refinement process in which overlooked anomalies were discovered by the ML team using different TSAD algorithms. During this process, channels were divided into target and non-target for anomaly detection. Non-target channels should only be used as additional information for the algorithms. They are not annotated and are not assessed in the benchmark. Examples include status flags, counters, and metadata such as location coordinates, where anomalies are not expected or it is not possible to check for anomalies without external data. Related channels measuring the same physical values and showing similar characteristics are organized into numbered groups, so it is easier to manage the dataset for ML purposes, e.g., to train group-specific models or to visualize results.

Table 7: List of selected challenging events annotated in test sets of ESA-ADB.

Mission	Event category	Event ID	Start time (YYYY-MM-DD hh:mm:ss)	Duration	Reason for selection
Mission1	Rare Event	id_24	2012-12-18 06:32:09	24h 15m	Hard to spot and not commanded. Not found by SOEs initially and added during the refinement process. It is related to a temporary change of nominal operational conditions.
	Rare Event	id_49	2011-10-08 07:08:39	10h 25m	Hard to spot, especially when looking at too narrow context. Caused by a rare TC. Fig. 17
	Rare Event	id_51	2011-08-14 19:12:39	1h 19m	Hard to spot, especially when looking at too narrow context. Caused by a rare TC. Fig. 18
	Rare Event	id_55	2011-04-23 08:19:39	0s (point)	Hard to spot in both lightweight and full sets. Caused by a unique execution of TC of priority 1. Overlaps with the rare event id_155.
	Anomaly	id_138	2009-10-13 06:39:17	1d 20h	Hard to spot using the lightweight subset of channels 41-46 only. Much easier to spot in channels 58-60. Fig. 19
	Anomaly	id_153	2011-01-28 22:29:18	15h 14m	Hard to spot using the lightweight subset of channels 41-46 only. Much easier to spot in channels 64-66. Fig. 20
	Rare Event	id_155	2011-04-21 22:15:52	11d	Hard to spot using the lightweight subset of channels 41-46 only. Easier to spot in multiple other channels, but hard to accurately identify the start time due to very slow changes. Main text Fig 2. and Fig. 21
	Anomaly	id_157	2011-04-19 07:09:39	14h 35m	Hard to spot using the lightweight subset of channels 41-46 only. Much easier to spot in channels 64-66.
	Rare Event	id_159	2011-06-09 02:57:09	10d 21h	Hard to spot using the lightweight subset of channels 41-46 only. Very long annotations in other affected channels. Fig. 22
Mission2	Rare Event	id_466	2003-02-08 16:25:19	1h 10m	Small disturbance in 7 channels which may be easily overlooked, especially when using only the lightweight subset.
	Rare Event	id_591	2002-04-16 16:30:53	35m	Small disturbance in 7 channels which may be easily overlooked.
	Anomaly	id_631	2001-12-14 19:16:29	1h 18m	Small disturbance of unknown source in 7 channels which may be easily overlooked by operators. Fig. 23
	Anomaly	id_644	2002-02-18 05:42:45	9h 48m	Divergence of channel 81 from channel 73 which can only be detected when looking at both channels in the proper context window.

There are hundreds of different telecommands (TCs) in each mission. Some of them are critical for detecting annotated anomalies (i.e., when there is no reaction to the TC or the reaction is different than usual) or distinguishing anomalies from rare nominal events. However, it may be impractical to use them all in anomaly detection algorithms. Thus, 4 different priority levels for TCs were introduced as a suggestion about their potential usefulness for anomaly detection algorithms. The priorities from the least important to the most important are:

0. TCs not directly related to any subsystem included in the dataset.

1. TCs related to subsystems included in the dataset but not marked as potentially valuable for anomaly detection by SOEs.

2. TCs selected as potentially valuable for anomaly detection by SOEs.

3. A subset of TCs of priority 2 for which there are more than 2 occurrences in the training data and at least one occurrence in the test data.

TCs of priority 3 are used as input for DC-VAE-ESA and Telemanom-ESA algorithms trained on full sets of channels. These priorities are only suggestions and ESA-ADB users are welcome to experiment with any combination of TCs.

B.4 Anonymization details

The anonymization had to be applied to conform with the ESA privacy policy and to avoid any accidental disclosure of sensitive mission-specific information or metadata. The anonymization process was carefully designed to maintain data integrity, so the results are independent of the anonymization. The following modifications were applied as a part of the anonymization process for each mission:

- *Renaming* of missions, subsystems, channels, telecommands, physical units, anomaly classes, and event types. They were consistently numbered according to their order of occurrence in files. Subsystems and physical units have consistent naming across missions, so it is possible to train cross-mission models.

- *Time scaling* and shifting of each mission. The timeline of every mission was scaled by a non-disclosed factor larger than 1 and shifted to start on 1st January 2000.

- *Normalizing* values within channel groups to <0, 1> range. Normalization per group was applied to preserve the same dependencies between similar channels before and after anonymization.

It was verified that the anonymization is fully reversible and there are no numerical errors related to the limited floating point resolution of values or timestamps. Additionally, it was verified that all deterministic algorithms in the benchmark produce the same results before anonymization.

B.5 Dataset structure

ESA-AD consists of three folders, one per each mission. Each folder has the same structure presented in Table 8. There is a subfolder named channels and an optional subfolder named telecommands. Both subfolders include serialised and compressed Pickle files (docs.python.org/3/library/pickle.html, protocol version 4.0, ZIP compression), one for each channel and telecommand. Each file contains a single pandas DataFrame (pandas version 1.5.3) including an index with consecutive timestamps and a single column with the corresponding raw telemetry values. Annotations of all events are in a separate file called labels.csv placed directly in the mission folder. It contains rows that describe anomalous fragments using 4 columns: the anomaly identifier (ID), the name of the channel affected by the anomaly, the start time, and the end time of the anomalous segment. Start and end times are defined as closed ranges and they usually represent timestamps of actual points in the dataset. There may be multiple segments with the same ID and channel name, but their time ranges cannot overlap. Additional information on anomalies can be found in the anomaly_types.csv file. It describes each anomaly ID with its class, subclass, category, and type. The channels are described in the channels.csv file using the channel name, the associated subsystem, and the physical unit. The channel description also includes group numbers that indicate similar channels and the information if the channel is a target channel. If telecommands are included in the dataset their priority is described in the telecommands.csv file.

Table 8: Folder structure of ESA-AD

Folder/File	Description
ESA-Mission/	Main folder for the mission
– channels/	Folder including all channels of the mission
—- *.zip	Compressed *Pickle* files for each channel
– telecommands/ (optional)	Folder including all telecommands of the mission
—- *.zip	Compressed *Pickle* files for each telecommand
labels.csv	Annotations
anomaly_types.csv	Description of anomalies and rare nominal events
channels.csv	Description of channels
events.csv (optional)	List of special operations and mission events
telecommands.csv (optional)	Description of telecommands

Some files are marked as optional, these files are not mandatory for the dataset or there might be missions not including these files. It should be possible to apply anomaly detection algorithms to the datasets not using the optional data, but it is expected that the optional data enhances the performance of algorithms when used. Mission2 includes an optional file events.csv which lists special operations and events with their start and end times according to the mission plan provided by SOEs. It was used to identify rare nominal events annotated in labels.csv, usually with slightly different start and end times due to different propagation times between channels.

B.6 Comparison to related public datasets

Space-related datasets. A quantitative comparison of the missions included in ESA-ADB and other public spacecraft-related telemetry datasets from the literature is presented in Table 9. There are 5 other public datasets of real-life spacecraft telemetry – 3 by NASA and

2 by ESA – and a single simulated one (CATS). The most popular ones are Soil Moisture Active Passive (SMAP) and Mars Science Laboratory (MSL) released by NASA (Hundman et al., 2018). According to the search for "SMAP" and "MSL" terms in Google Scholar since 2018, there are more than 200 documents that mention these datasets in the set of more than 500 citations of the source paper. Besides a lot of criticism of these datasets in the recent literature (Wu and Keogh, 2023; Wagner et al., 2023; Amin Maleki Sadr et al., 2023), there is a common misconception about the number of channels included in these datasets. The data may come from 82 different physical channels in total, but there is a separate fragment for each channel without any synchronization with other channels, so they cannot be used effectively as a multivariate dataset. This is made clear in the description of the dataset in Table 9. NASA LASP WebTCAD (Sanchez et al., 2018) has tens of millions of points, but there are only 5 partially overlapping channels and no annotations of anomalies. ESA Mars Express Power Challenge (Petković et al., 2022) is popular in satellite telemetry forecasting, but does not contain anomalies annotations. The very recently published ESA OPSSAT-AD (Ruszczak et al., 2025) is a toy dataset with 2213 univariate fragments of real OPS-SAT telemetry. It is contains anomaly annotations for whole fragments and is designed specifically for on-board applications.

Non-space-related datasets. In Table 10, there are several related real-life datasets from outside the domain of satellite telemetry that are frequently used to benchmark multivariate TSAD algorithms. Notable examples include the Secure Water Treatment (SWaT) (Mathur and Tippenhauer, 2016) and Water Distribution (WADI) (Ahmed et al., 2017) datasets which contain recordings from tens of channels from a real-world water treatment plant within several days. Server Machine Dataset (SMD) (Su et al., 2019) including 5-week-long data from 38 parameters of 28 machines from 3 servers at a large internet company. The recent TELCO dataset (González et al., 2023) is worth noting due to related ideas of separate annotations for each channel, anomalies in training sets, and gradually increasing training set sizes. It contains 12 channels corresponding to real measurements collected over 7 months at an operation mobile internet service provider. To the best of our knowledge, the Exathlon benchmark (Jacob et al., 2021), including real data traces from tens of repeated executions of streaming jobs with 2283 parameters (channels) on a Spark cluster over 2.5 months, is the only related dataset of volume comparable to ESA-AD, with more than 5 billion samples and 25 GB of data. However, it does not contain per-channel annotations and has been criticized for unrealistic anomaly density and positional bias (Wagner et al., 2023).

Table 11 gives qualitative description of how ESA-ADB addresses main 4 flaws reported by Wu and Keogh (2023).

Appendix C. Methods

C.1 Anomaly types

Figure 11 gives an overview of the proposed taxonomy of anomaly types and Table 12 gives statistics of each anomaly types across all missions in ESA-AD.

Table 9: Quantitative comparison of ESA-AD and other public *space-related* time series anomaly detection datasets.

Dataset name	**Number of fragments and variables**	**Total volume**	**Number of annotated events**
ESA-AD (Ours)	**Mission1**: 1 fragment with 76 channels and 698 commands **Mission2**: 1 fragment with 100 channels and 123 commands **Mission3**: 1 fragment with 48 channels	2,296,122,157 samples 3,512,724 commands executions	1,430 (1.14% of all samples)
NASA SMAP and MSL (Hundman et al., 2018)	**SMAP**: 55 fragments with 1 channel and 24 commands **MSL**: 27 fragments with 1 channel and 55 commands	706,971 samples 410,030 commands executions	105 (8.98% of all samples)
NASA LASP WebTCAD (Sanchez et al., 2018)	1 fragment with 5 partially overlapping channels	55,258,122 samples	not annotated
NASA Shuttle Valve Data (Ferrell and Santuro, 2005)	**TEK**: 12 fragments with 1 channel **VT1**: 27 fragments with 1 channel	552'000 samples	8 whole fragments
CATS (Fleith, 2023) (simulated)	1 fragment with 17 channels	85,000,000 samples	200 (2.15% of all samples)
ESA Mars Express Power Challenge (Petković et al., 2022)	**Train**: 3 fragments with 38 channels (including 5 metadata-related) **Test**: 1 testing fragment with 5 metadata channels	198,045,083 samples	not annotated for anomalies
ESA OPSSAT-AD (Ruszczak et al., 2025)	2123 univariate fragments from 9 different channels	303,493 samples	434 whole segments (20.44%)

Table 10: Quantitative comparison of ESA-AD and other public *non-space-related* time series anomaly detection datasets.

Dataset name	**Number of fragments and variables**	**Total volume**	**Number of annotated events**
ESA-AD (Ours)	3 fragments 1045 variables	2,296,122,157 samples	1,430 (1.14% of samples)
SWaT (Mathur and Tippenhauer, 2016)	36 fragments 51 variables	944,911 samples	36 (12.14% of samples)
WADI (Ahmed et al., 2017)	15 fragments 127 variables	962,172 samples	15 (5.77% of samples)
SMD (Su et al., 2019)	28 fragments 38 variables	708,420 samples	67 (1.02% of samples)
TELCO (González et al., 2023)	1 fragment 12 variables	732,607 samples	36 (1.25% of samples)
Exathlon (Jacob et al., 2021)	93 fragments 2283 variables	5,332,588,023 samples	97 (37.97% of samples)

Table 11: A list of flaws reported by Wu and Keogh (2023) and how they are addressed by ESA-ADB.

Flaw	How does ESA-ADB address it?
Triviality	• ESA-AD is large and contains a diverse set of anomaly types and concept drifts which hamper the usage of simple algorithms. • ESA-AD offers a selection of non-trivial anomalies, so they can be evaluated separately (Appendix B.2). • ESA-ADB includes a set of simple algorithms to verify the potential triviality of anomalies.
Unrealistic anomaly density	• ESA-AD is large and the anomaly density in the dataset is below 2% of data points. • There are only dozens of anomalous events per year. • Series of separate annotated segments within a short region are usually assigned to the same event and are treated as such when computing metrics.
Mislabelled ground truth	• While this flaw cannot be fully resolved in real-life datasets there were several iterations of the annotation refinement process aided by unsupervised and semi-supervised algorithms to identify potential mislabelling (Kotowski et al., 2023).
Run-to-failure bias	• Anomalies are scattered across long, failure-free, operational periods of acquired telemetry data from real satellite missions.

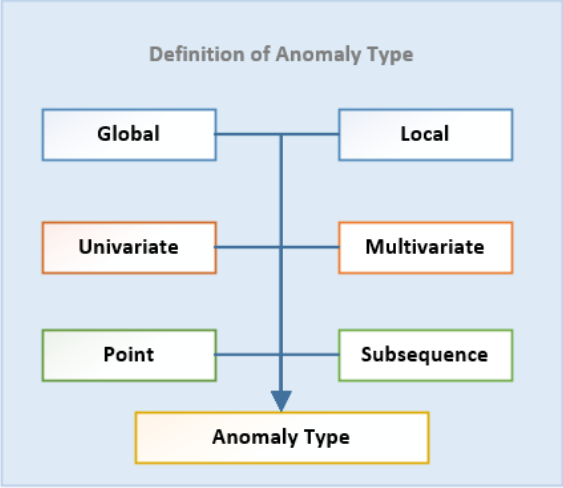

Figure 11: Anomaly types considered in ESA-ADB.

Table 12: Distribution of 8 combinations of anomaly types across missions

Length	Locality	Dimensionality	Mission1	Mission2	Mission3
Point	Global	Univariate	0.00%	0.00%	9.09%
		Multivariate	5.10%	0.00%	0.00%
	Local	Univariate	0.51%	0.00%	0.00%
		Multivariate	0.51%	0.00%	0.00%
Subsequence	Global	Univariate	12.24%	0.00%	60.61%
		Multivariate	40.31%	90.84%	15.15%
	Local	Univariate	3.57%	1.40%	6.06%
		Multivariate	37.76%	7.76%	9.09%

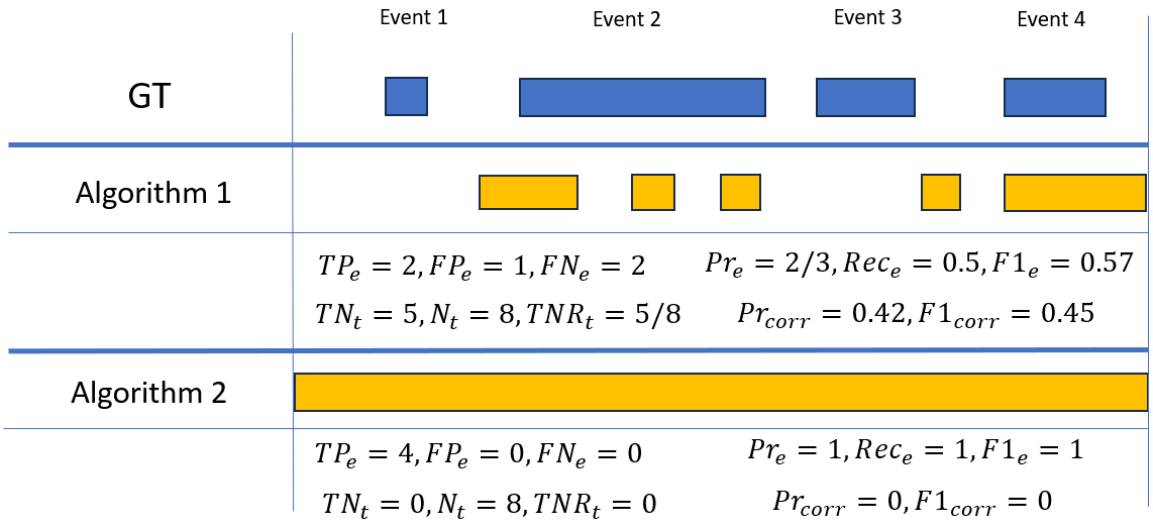

Figure 12: Visualization of differences between the original and corrected event-wise F-scores.

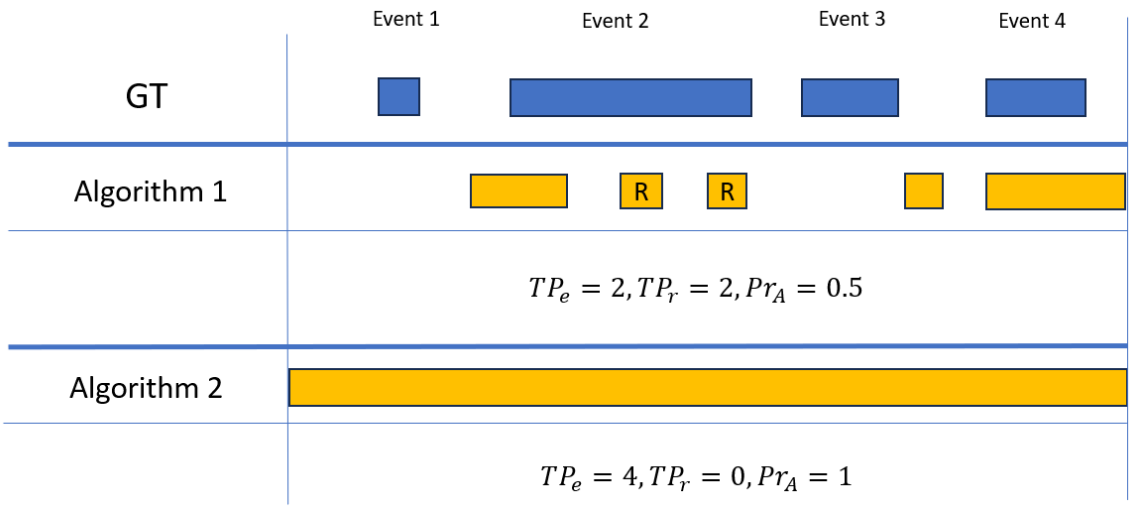

Figure 13: Visualization of the event-wise alarming precision calculation.

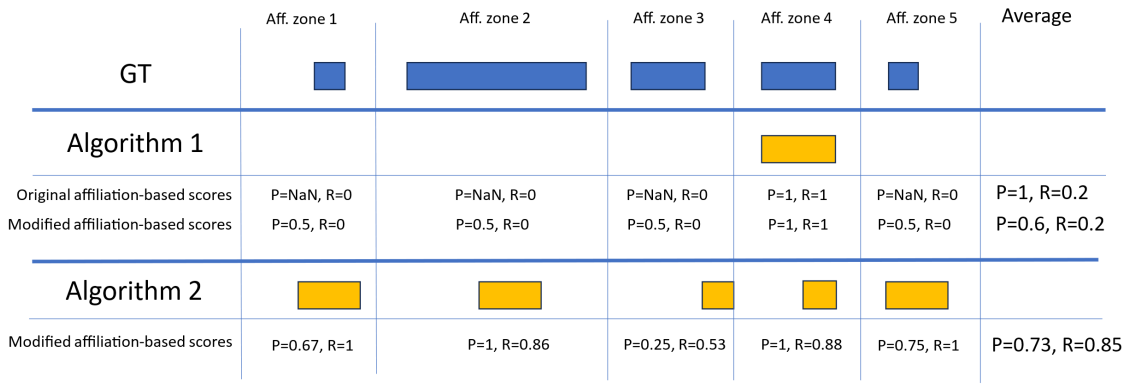

Figure 14: Visualization of differences between the original and modified affiliation-based scores on a simulated example (P – precision, R – Recall).

C.2 Metrics

C.2.1 Analysis of metrics from the literature

There are some recent comprehensive reviews of TSAD metrics in the literature (Kim et al., 2022b; Sørbø and Ruocco, 2023). It is an active area of research in which many new approaches were introduced recently. However, none of these metrics is universal, they focus on different aspects of detection quality which may differ between applications. Table 13 gives an overview of our analysis of state-of-the-art metrics together with their pros and cons in the context of satellite telemetry anomaly detection.

Table 13: Analysis of existing metrics from the literature with reasons for inclusion/exclusion in ESA-ADB.

Category	Metric	Included in ESA-ADB	Reason for inclusion/exclusion
Sample-wise	All classic precision-based and recall-based metrics treating each time point as an independent sample	No	Simple per-sample metrics are not aware of temporal aspects of time series in which anomalies are usually continuous sequences of correlated samples (like a vast majority of anomalies in satellite telemetry). This leads to multiple problems with a proper assessment of algorithms, i.e., longer anomalies are getting more importance than shorter ones, and close detections are not rewarded.
	NAB score (Lavin and Ahmad, 2015)	No	It is only applicable to domains where an anomaly is a single sample rather than a series of samples. Moreover, there are certain ambiguities in the scoring functions and magic numbers for its parameters (Singh and Olinsky, 2017).
Range-based			

Continued on the next page

Table 13: Analysis of existing metrics from the literature with reasons for inclusion/exclusion in ESA-ADB. (continued)

Category	Metric	Included in ESA-ADB	Reason for inclusion/exclusion
	Point-adjust (Xu et al., 2018)	No	It is recently widely criticized for various reasons (Hwang et al., 2022; Huet et al., 2022; Abdulaal et al., 2021; El Amine Sehili and Zhang, 2024). Mainly because it is overly optimistic and even a random anomaly score can reach state-of-the-art results in this metric in specific adversarial cases. Also, it gives more importance to longer anomalies and is unable to give a larger score to a model which finds the true range of the anomaly better.
	Event-wise score (Hundman et al., 2018)	No	It is improvement of point-adjust approach which partially solve the problem with higher importance of long anomalies, i.e., long anomalies have the same "weight" as short ones, but there is still a much higher probability that a random detector will detect the longer anomaly. Moreover, an algorithm that simply detects anomalies for every sample in the dataset would have a perfect event-wise precision.
	Composite F-score (Garg et al., 2022)	No	It calculates the precision in a per-sample manner and recall in an event-wise manner, so it still gives smaller weights to short false positives which are equally or more annoying than longer ones.
	Corrected event-wise F-score (El Amine Sehili and Zhang, 2024)	Yes	It resolves issues with the event-wise precision using a correction for per-sample true negative rate. It increases importance of short false positives and penalizes overly long detections.
	Precision and Recall for Time Series (Tatbul et al., 2018)	No	It requires 4 hyperparameters to be tuned. The recall is not monotonically decreasing with an increasing threshold. Such a behavior can even lead to problems when computing aggregated metrics that assume recall consistency (Wagner et al., 2023). Its calculation time does not scale well for large datasets.
	TaPR and eTaPR (Hwang et al., 2022)	No	It requires 3 hyperparameters and its calculation time does not scale well for large datasets.
	Affiliation-based score (Huet et al., 2022)	Yes	It is parameter-free, locally and statistically interpretable, robust to "adversary" predictions, and scales well for large datasets.
	Volume Under the Surface (Paparrizos et al., 2022)	No	It operates on continuous anomaly scores and we require binary outputs. It has very high computational complexity and does not scale well for large datasets (Boniol et al., 2025).
Detection timing	Early Detection (Buda et al., 2017)	No	It assumes that anomalies can only be detected within the ground truth interval – *after* they appear in the signal. It does not account for too early detections which may be treated as false positives by spacecraft operators.
	Before/After-TP (Nalepa et al., 2022b)	No	This approach requires calculation of two different values (Before-TP and After-TP) which are not obvious to aggregate to assess the overall detection timing quality. Moreover, it is impossible to calculate both values for every detections.

Continued on the next page

Table 13: Analysis of existing metrics from the literature with reasons for inclusion/exclusion in ESA-ADB. (continued)

Category	Metric	Included in ESA-ADB	Reason for inclusion/exclusion
Channels identification	HitRate@P% (Su et al., 2019)	No	It needs information about the relative relevance of detections that is not available when using binary outputs. It does not penalize models detecting too many channels. To always achieve a score of 1 model can simply mark all channels as anomalous. It is a crucial problem in satellite telemetry because we aim to mark only relevant channels to investigate by space operations. Any additional irrelevant detections should be penalized.
	Normalized Discounted Cumulative Gain (Järvelin and Kekäläinen, 2002)	No	It needs information about the relative relevance of detections that is not available when using binary outputs.

C.2.2 Description of metrics priorities

The highest priority aspect relates to the proper identification of anomalous events, but with a strong emphasis on avoiding false alarms at the same time (aspects 1a. "No false alarms" and 1b. "Anomaly existence" in Table 3 in the main text). This is because false positives are costly to resolve and deter operators from using the system. A high false positive rate is reported in the literature as the main obstacle to the wider adoption of anomaly detection algorithms in space operations (Hundman et al., 2018). This fact additionally supports our idea of hierarchical evaluation, since a high false positive rate disqualifies an algorithm even if it obtains perfect scores in other aspects. Moreover, most other aspects focus only on performance for true positive detections (i.e., channel identification, alarming precision, timing quality), so they indirectly depend on the anomaly existence aspect.

The second highest priority for SOEs is to have information about subsystems and channels affected by anomalies (aspects 2a and 2b). Proper subsystem identification is more important for SOEs as it gives a more concise overview of the situation than a long list of specific affected channels. Again, it is of paramount important to avoid false positives. It is strongly preferable to miss some channels rather than to wrongly identify many irrelevant channels. ESA-AD contains tens of target channels which is already hardly manageable for manual analysis, moreover, it is just a fraction of channels from actual missions. Hence, an algorithm which does not provide affected channels is of low practical utility, or even worse, it may amplify the "black box" nature of advanced algorithms and decrease trust in this kind of system among operators. That is why it was considered as the second of two primary aspects of highest priority.

The following 3 secondary aspects are not so crucial for SOEs but are certainly useful to differentiate between algorithms having the same primary scores. The 3rd priority is to avoid algorithms that frequently repeat alarms for the same continuous anomaly segment (aspect 3. "Exactly one detection per anomaly"). It is strongly connected to the 1a. "No false alarms" priority, because even if all repeated alarms are true positives, they would be annoying and confusing to operators, nearly as badly as false positives. The last 2 priority levels directly relate to the anomaly detection timing. It is better to detect anomalies earlier

than later (aspect 4. "Detection timing"), it is preferable to detect a whole time range of an anomaly instead of just a part of it, and, in case of false detections, it is better to show them close to real anomalies (aspect 5. "Anomaly range and proximity"). These aspects are often highly emphasized in TSAD benchmarks from the literature, e.g., NAB (Lavin and Ahmad, 2015) and Exathlon (Jacob et al., 2021). However, they are relatively less important for on-ground mission control. Additionally, the latter aspect cannot be precisely assessed due to the problems with the objective identification of anomaly end times (discussed in Appendix B.3).

C.2.3 ESA-ADB metrics definitions

The definitions of the proposed metrics are given in the following paragraphs. All implementations are available in the published code. All metrics are defined in the <0, 1> range where 1 is the perfect score. Technical details of implementations are listed in Section 3.2.4.

Subsystem/channel-aware F-scores. Typical TSAD metrics are applicable only in univariate settings. To get a single score for multiple channels, there must be some aggregation performed. Such aggregation loses information about the performance for individual subsystems or channels, so it is impossible to assess their correct identification. In recent articles (Zhao et al., 2020; Tuli et al., 2022), special *anomaly diagnosis* metrics are proposed to address this issue, namely *HitRate* and *Normalized Discounted Cumulative Gain*. These metrics measure how relevant are the detected channels according to the list of annotated channels. However, they need information about the relative relevance of detections which is not available when using binary outputs. Thus, a new approach is proposed based on precisions and recalls of identifying the list of affected subsystems and channels.

SOEs inspect potential anomaly sources at two levels of detail. First, they check which subsystems are affected by the anomaly. Later on, they look at the specific channels affected in those subsystems. The usefulness of algorithms supporting such inspection is proposed to be measured with *subsystem-aware* (SA) and *channels-aware* (CA) F-scores. A subsystem is counted as true positive (TP_{SA}) if it has at least one annotated channel and at least one detected channel (not necessarily the same) overlapping with the full time span of an event. A subsystem is considered a false negative (FN_{SA}) if it has at least one annotated channel but no such detections. A false positive subsystem (FP_{SA}) has no annotated channels but has at least one such detection. Thus, the *subsystem-aware F-score* $F_{\beta_{SA}}$ is given by:

$$F_{\beta_{SA}} = \left(1+\beta^2\right) \frac{Pr_{SA} \cdot \ Rec_{SA}}{(\beta^2 \cdot \ Pr_{SA}) + Rec_{SA}},$$

$$Pr_{SA} = \frac{TP_{SA}}{TP_{SA} + FP_{SA}}, \qquad Rec_{SA} = \frac{TP_{SA}}{TP_{SA} + FN_{SA}}$$

The *channel-aware F-score* is defined analogously, taking into account separate channels instead of subsystems. Again, 0.5 is used for β as a baseline to be consistent with the event-wise F-score. For the lightweight subsets of channels (selected from a single subsystem), the subsystem-aware F-score is not reported.

Event-wise alarming precision. The corrected event-wise F-score counts only a single true positive even if there are multiple separated detections for the same fragment in the

ground truth (Figure 13). In practice, such redundant detections may cause repeated alarms which may be annoying for operators. The event-wise alarming precision measures the ratio of correctly detected events to the sum of correctly detected events and redundant alarms. This metric may seem too strict in some cases, i.e., many short detections close to each other, but it represents a practical aspect of mission operations and encourages applying better thresholding or postprocessing approaches to avoid redundant alarms.

Anomaly detection timing quality curve (ADTQC). The goal of this novel metric is to assess the accuracy of the anomaly start time identification from the SOEs point of view. Some existing metrics of the anomaly detection latency, such as *After-TP* (Nalepa et al., 2022b) or *Early Detection* (ED) (Buda et al., 2017), assume that an anomaly can be detected only within its ground truth interval – *after* it appears in the signal. However, the question arises how to assess algorithms that detect anomalies too early – *before* they start. They cannot be assessed using *After-TP* or *ED* metrics but they certainly have some value. The *Before-TP* metric (Nalepa et al., 2022b) and the *NAB score* (Lavin and Ahmad, 2015) rank earlier anomaly *predictions* (to distinguish them from *detections*) as better. However, in practice, as suggested by SOEs, too-early detections may be seen as false positives by operators if they cannot confirm the existence of an anomaly within a definable time. Thus, too early detections may decrease operators' trust in an algorithm and, in this context, are much worse than late detections of comparable distance from an anomaly start time. According to SOEs, the quality of anomaly detection timing should decrease exponentially for detections before the actual start time as opposed to much slower degradation of quality for moderately late detections. A survey was conducted and confronted across SOEs from different missions in ESA and KP Labs to define the timing quality in the range from 0 to 1 as a function of detection start time. The resulting consensus reflecting the common operators' point of view is reflected in the ADTQC described in Appendix Section C.2.3.

Modified affiliation-based F-score. The affiliation-based approach by Huet et al. (2022) claims to resolve all the major flaws of previous range-based metrics. That is, it is aware of the temporal adjacency of samples and anomalies duration, has no parameters, and is locally and statistically interpretable. The main idea is to divide ground truth into local zones affiliated with consecutive anomaly ranges. The borders of such *affiliation zones* lie in midpoints between consecutive anomalies. Precision and recall are calculated separately for each affiliation zone based on the average directed distance between sets of annotated and detected points, either the distance from annotated to detected (precision) or from detected to annotated (recall). Affiliation-based F-score with β of 0.5 is calculated to underscore the strong practical need to minimize the number of false positives. The final global F-score is calculated as the arithmetic average of all local F-scores (with each affiliation zone having the same weight). The important modification to the original implementation relates to frequent situations when it is impossible to calculate the precision in an affiliation zone (when there is no detection there are no true positives or false positives). In the original formulation, such an affiliation zone was simply ignored when calculating an average precision over all affiliation zones. However, this approach makes it hard to robustly compare different algorithms because of the different numbers of affiliation zones taken into account, e.g., it gives a higher score to an algorithm that detects a single anomaly very precisely and misses 4 others than to an algorithm that detects all 5 anomalies relatively well – see Fig. 14. In

our formulation, empty detections get a precision of 0.5. Such a value can be interpreted as a random detection, so this modification promotes algorithms that would rather give an empty detection than a false detection that is worse than random. There are also some other technical adaptations to handle point anomalies and fragmented annotations, as described in Appendix Section C.2.2.

C.2.4 Implementation details

Operating in the time domain . ESA-AD has varying sampling rates and we keep them on purpose to maintain the true characteristics of satellite telemetry data. Our evaluation pipeline handles this issue in order to consistently evaluate the results of algorithms using different sets of timestamps on the output. We achieved this by implementing all metrics to operate in the time domain instead of the samples domain. In this case, ground truth and detections can use completely different sets of timestamps (of different lengths and varying sampling rates). Traditional implementations of most metrics assume that ground truth and detections have the same uniformly sampled timeline. Our metrics operate on arbitrarily timestamped ground truth and detection arrays (possibly of different lengths and sampling frequencies). Hence, no matter the sampling frequency used in the algorithm, the metrics are always calculated relative to the original non-uniformly sampled ground truth. For operations on time ranges, we use the *portion* library (github.com/AlexandreDecan/portion).

Timestamps as real numbers. Our modified version of the affiliation-based metric operates on timestamped arrays, but the timestamps are transformed into the number of nanoseconds since the beginning of the dataset, so the internal implementation of the affiliation-based score is unchanged (it can operate on real numbers only). Additionally, point events (with the same start and end times) are adjusted, so that the end time is 1 nanosecond later than the start time. Such modification had to be applied because the affiliation-based score cannot be calculated for point events of zero length. The same point anomalies adjustment was applied to the channel-aware F-scores.

Multiple annotations for the same event. It is a common situation in our dataset that multiple non-overlapping fragments close to each other are annotated with the same event ID (e.g., Fig 2. in the main text). This is usually because the source of the anomaly is the same for all fragments. In such cases, we should treat all fragments as a single anomaly (i.e., when selecting affiliation zones and calculating distances) as suggested in recent literature (Wu and Keogh, 2023; Huet et al., 2022). To implement such correction in the affiliation-based score without changing its internal assumptions and implementation, a macro-averaging across anomaly IDs was introduced, i.e., it first aggregates zones affiliated with the same anomaly IDs by averaging their precision and recall scores and then calculates an average across all anomaly IDs. It also affects the implementation of the alarming precision metric which does not penalize redundant detections belonging to non-overlapping ground truth fragments.

Overlapping events. There are several cases in our dataset where annotations of different events for the same channel are overlapping in time, i.e., when an anomaly occurred during a longer rare event. The affiliation-based metric is unable to separate such events because it is impossible to create non-overlapping affiliation zones for them, so there are no corrections

for this situation to not interfere with the main principles and assumptions of the metric. For subsystem-aware and channel-aware metrics, each event is analyzed separately, i.e., separate true positives and false negatives are counted for each event. Moreover, any false positives related to correct detections of other overlapping anomalies are discarded.

ADTQC details. The *anomaly detection timing quality curve (ADTQC)* is visualized in Figure 15 and defined by equation

$$ADTQC(x) = \begin{cases} 0, & -\infty < x \leq -\alpha \\ \left(\frac{x+\alpha}{\alpha}\right)^e, & -\alpha < x \leq 0 \\ \frac{1}{1+\left(\frac{x}{\beta-x}\right)^e}, & 0 < x < \beta \\ 0, & \beta \leq x < +\infty \end{cases}, \qquad \alpha, \beta > 0$$

$$ADTQC(x) = \begin{cases} 0, & x \neq 0 \\ 1, & x = 0 \end{cases}, \quad (\alpha = 0 \wedge x \leq 0) \vee (\beta = 0 \wedge x \geq 0)$$

$\alpha = \min(\text{anomaly length},\ \text{anomaly start time} - \text{previous anomaly start time})$

$\beta = \text{anomaly length}$

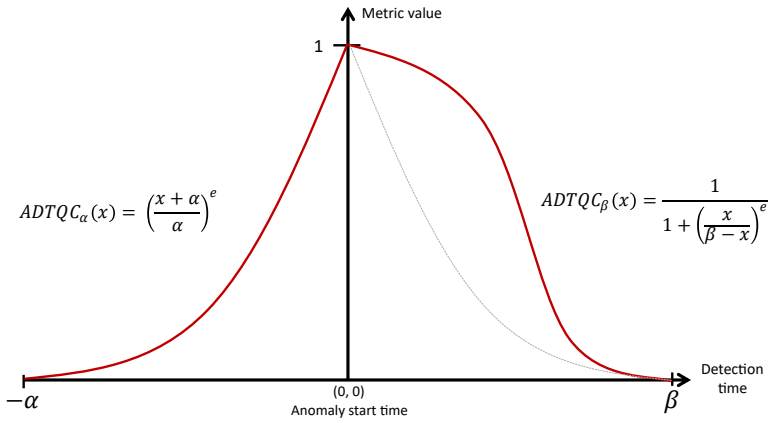

Figure 15: Anomaly detection timing quality curve (ADTQC).

After agreeing on the shape of ADTQC, the most important issue was to select the operational range of values for which the function should return a quality higher than 0, that is, for which the detection is not useless from the practical point of view. The first straightforward step was to define detections later than the anomaly end time (β) as useless. Accordingly, detections earlier than the anomaly length from the start time were also considered useless. Hence, the shorter the anomaly the more accurately it must be detected to achieve similar quality value. In the extreme case of point anomalies, ADTQC returns a value of 1 for exact detections and 0 otherwise. It makes sense from the practical point of view for two reasons, 1) detections for short, hardly noticeable anomalies are likely to be considered false alarms if not well-timed, and 2) end times of long anomalies are

usually much harder to annotate precisely than for short anomalies. Another unacceptable situation was identified when a detection is earlier than the previous anomaly start time. When anomalies are close to each other, the detection timing must be even more accurate to ensure their better separation.

The ADTQC metric value for the specific anomaly is determined by simply calculating the value of the $ADTQC(x)$ function where x is the difference between the detection start time and the anomaly start time. Similarly to Before/After-TP (Nalepa et al., 2022b), the metric is calculated and averaged across all correctly detected events to get a final score in the range from 0 to 1. To support the analysis of the results, the ratio of detections after the anomaly starting points to all detections is calculated (called the *after ratio*).

For multivariate anomalies, the ADTQC metric is calculated between the logical sums of annotations and detections across all target channels. It does not matter if the detections are for correct channels because the metric focuses on the timing alone. The second possible approach in the multivariate setting would be to calculate the ADTQC metric for each affected channel separately. The average across all affected channels would be the final ADTQC score for a specific anomaly. While this alternative approach would allow for more detailed quantification of the anomaly detection timing across channels, it does not reflect the operators' perspective in which the first detection is the most important one, because it already enforces an action. Later detections for any other channel do not matter so much, because operators are already aware of the potential anomaly.

All metrics can be calculated excluding some specific event categories, classes, or types. For the corrected event-wise F-score, detections for excluded events are ignored when counting true and false positives, and a lack of detection is not counted as a false negative for them. For other metrics, excluded events are simply not considered when calculating the mean across events.

C.2.5 Approach for rare nominal events

Most algorithms in the TimeEval framework (and in the literature) do not support learning rare nominal events explicitly (i.e., by one-shot learning or keeping rare events in memory). For such standard algorithms, rare events will always be detected as anomalies, so for simplicity, rare nominal events are treated as anomalies in the current benchmark. However, we strongly encourage to use ESA-AD to design models that learn nominal rare events and avoid detecting them in the future, which would be of high practical importance for mission control. For this purpose, we propose a framework to assess them:

1. The first detection of a novel rare nominal event (not seen during training) should not be penalized. However, the algorithm should be able to actively learn from the operators' feedback (i.e., "*this is not an anomaly*") and should not detect similar events in the future (one-shot learning).

2. For known rare events (seen during training or actively learned during inference), every subsequent detection should be penalized, i.e., we should minimize per-event false positive rate (FP / (FP + TN)) where FP is falsely detected rare event and TN is a correctly undetected rare event.

C.3 Preprocessing

Figure 16 presents an example of the proposed zero-order hold resampling scheme. It is implemented as follows:

1. Construct a uniformly sampled list of timestamps in the target sampling frequency. Set the first/last timestamp in the list to the value of the earliest/latest original timestamp across all channels, rounded down/up to the target sampling resolution. Fill the list between the first and the last element using uniformly sampled timestamps in the target frequency.
   Example: If we resample a list of original timestamps ⟨8:10:12, 8:10:14, 8:10:38⟩ to the target frequency of 1/10 Hz (target resolution of 10 seconds), the resampled list will be ⟨8:10:10, 8:10:20, 8:10:30, 8:10:40⟩.

2. Propagate the last known value and label from the original samples (zero-order hold) to each timestamp in the constructed list. If there are still any missing values for the initial element of the list (i.e., when some channels start a little earlier than others), backpropagate the first known value from the original samples. This introduces a bit of information from the future, but it usually concerns only a few samples at the beginning of a test set.

3. Apply a correction for missing anomalies to ensure that no point events are removed due to the resampling. Iterate through consecutive pairs of unannotated timestamps in the resampled list and, if there are any annotated original points in between, take the last annotated sample and assign its value and label to the latter timestamp from the pair. The result of such a correction is visible in the rightmost sample of `Channel_1` in Figure 16.

Target sampling frequencies have been selected separately for each mission (0.033 Hz, 0.056 Hz, 0.065 Hz, for Missions 1, 2, 3, respectively) based on the analysis of the most densely sampled target channels to prevent losing any annotated anomalies, especially point anomalies.

Channels with categorical values and status flags are enumerated according to the order of occurrence of each state in the training set before standardization. This is a very naïve approach, but it does not require laborious manual analysis of all channels and preparation of state mappings for each potential mission. Also, it does not require special handling of categorical anomalies. Moreover, categorical channels are usually non-target.

C.4 Algorithms

This section provides all details on algorithms implementation, selection, and parametrization in the benchmark.

C.4.1 Telemanom adaptation (Telemanom-ESA)

The semi-supervised Telemanom algorithm proposed by NASA engineers (Hundman et al., 2018) is an important point of reference in the domain. It can be considered the most popular algorithm for anomaly detection in satellite telemetry. Its core element is an LSTM-based

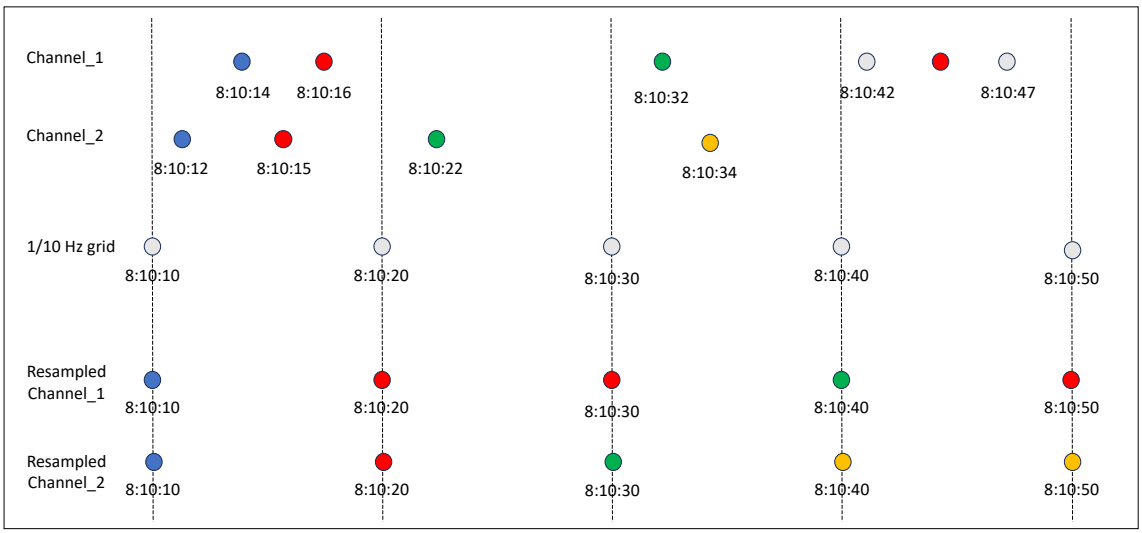

Figure 16: Visualization of our resampling procedure for two non-uniformly sampled channels. Colors represent different values of the signal for each channel.

RNN that learns to forecast a small number of time points (10 by default) for a single channel based on the hundreds of preceding samples (250 by default) from multiple input channels. The mean absolute difference between the forecasted samples and the real signal is treated as an anomaly score, which is thresholded using the non-parametric dynamic algorithm (NDT) to find anomalies. However, this "non-parametric" approach (in the sense that it does not use Gaussian distribution parameters to estimate thresholds) has several hyperparameters. In one of our previous works, a genetic algorithm was used to find optimal hyperparameters (Benecki et al., 2021). However, this wrapper approach would be too computationally expensive to run on our large datasets, so the default settings proposed by the authors were used.

Our proposed *Telemanom-ESA* solves several technical issues of the original Telemanom identified when working on ESA-ADB, including memory inefficiency, magic numbers causing problems in thresholding (*Telemanom-ESA-Pruned*), no proper handling of anomalies in training data, and a lack of scalability to hundreds of channels.

- **Memory inefficiency** – Telemanom was designed for small and simplified datasets provided by NASA. Hence, the code is not optimized to handle very large datasets and it results in out-of-memory errors, e.g., there are many unnecessary copies of data, all training windows are loaded into memory at once, and binary annotations are loaded to memory as floating-point numbers. **Telemanom-ESA:** The code is optimized for memory consumption by using lazy generators to prepare training batches, in-place operations instead of copying data to new variables, and optimized data types.
- **Magic numbers in thresholding** – there are several conditions in the thresholding code that are not well documented in the original article. Especially impactful is

that windows with smoothed errors below 0.05 are never anomalous[1]. This is a very data-specific condition that is not well-suited for channels with certain signal values. **Telemanom-ESA**: This specific condition was removed from the code. **Telemanom-ESA-Pruned**: The threshold of 0.05 is much too high for ESA-ADB, so it was changed to 0.007 based on the manual analysis of smoothed errors in the training data of both missions. This selection is highly subjective and is probably not optimal, but allows to assess the effect of such a pruning on the results.

- **No proper handling of anomalies in training data** – Telemanom assumes that there are no anomalies in the training set which is not true in our real-life setting. **Telemanom-ESA**: only continuous nominal parts longer than 260 samples and without any anomalies in any target channel are used for training and validation.
- **Only a single output from the LSTM model** – a single Telemanom model can take multiple input channels but it always outputs a prediction for a single target channel. This is a significant shortcoming when scaling this approach to hundreds of channels and gigabytes of data. The training of a single model may last hours or days, so training separate models for tens of channels can take months on a single PC. Also, it is impossible to provide different sets of input (non-target channels, telecommands) and output (target) channels. **Telemanom-ESA**: the output of Telemanom is extended, so that is possible to forecast any number of channels at once from a single model, like in DC-VAE (González et al., 2023). The channels are still analyzed separately, but there is no need to train a separate model for each channel.
- **Problems with GPU support** – the original implementation of Telemanom is based on TensorFlow version 2.0 which does not natively support the CUDA compute capability 8.6 of our Nvidia GPUs. Also, the TimeEval framework lacks GPU support. **Telemanom-ESA**: TensorFlow is upgraded to version 2.5 and the GPU support is added to TimeEval.

C.4.2 GlobalSTD

In this simple distribution-based approach, any samples deviating from the mean of the channel by more than N its standard deviations are detected as anomalies. This approach is categorized as semi-supervised because only nominal samples (excluding annotated events) from the training set are used to compute means and standard deviations for each channel to avoid the influence of outliers. In practice, the threshold of 3 standard deviations (*GlobalSTD3*) is frequently used (following the empirical statistical rule that 99.7% of data occurs within 3 standard deviations from the mean within a normal distribution (Grafarend, 2006)), but it may not be optimal when the number of false positives should be minimized, so the threshold of 5 standard deviations (*GlobalSTD5*) is also tested to provide a versatile baseline for other algorithms. This algorithm is unable to detect local anomalies, so it is not a good choice in practice. It is also not aware of dependencies between channels and it is very vulnerable to changes in the data distribution during the mission. It also cannot use the information about non-target channels and telecommands.

1. github.com/khundman/telemanom/blob/26831a05d47857e194a7725fd982d5dea5402dd4/telemanom/errors.py/#L339

C.4.3 DC-VAE and its adapted version (DC-VAE-ESA)

Dilated Convolutional-Variational Auto Encoder (DC-VAE) (González et al., 2023) is one of the latest published multivariate TSAD algorithms. It is a reconstruction-based method that relies on dilated convolutions to capture long and short-term dependencies without using computation- and memory-intensive multi-layer RNNs. Unlike the original Telemanom, it does not need a complicated thresholding scheme, because it also estimates nominal standard deviations for each sample in each channel, so thresholding can simply be applied by looking for real samples exceeding reconstructions by more than N standard deviations. In the original implementation, N is selected from integers between 2 and 7 to maximize the range-based F1-score for each channel in the training set. This approach does not scale well with the number of channels and assumes the similarity of anomalies between the training and test sets. Thus, in DC-VAE-ESA, only two values of N are considered, 3 (*STD3*) and 5 (*STD5*).

The modified DC-VAE-ESA introduces only two small technical improvements to fully cover 7 of the 9 mentioned requirements, 1) an option to handle different numbers of input and output channels, 2) L2 regularization of convolutional layers with the 0.001 rate to stabilize the training of VAE in the presence of concept drifts.

C.4.4 Experiments with transformers

We have experimented with transformer-based anomaly detectors, namely TranAD (Tuli et al., 2022) and Anomaly Transformer (Xu et al., 2021) algorithms, using the code from their original repositories. We have not included the results in the current benchmark for several reasons:

1. We were not able to achieve any reasonable qualitative results using the default hyperparameters of these algorithms. They would require additional investigation and hyperparameters selection that go beyond the scope of the study and computational resources allocated to it.

2. The algorithms are not a part of the TimeEval framework and it would require additional effort to integrate and make them reproducible within our pipeline.

3. The original implementations do not handle irregular sampling rates of data. Their data loaders assume uniform sampling when creating windows for training, so it is not possible to take advantage of the positional enconding of transformers.

4. The original implementation of Anomaly Transformer provides just a single global anomaly score, so it does not meet the requirement R5 from Table 2 in the main text ("provide a list of affected channels").

Nevertheless, transformers have many features desirable in satellite anomaly detection (e.g., long context, few-shot learning, and support for irregular time series) and remain a promising direction in satellite anomaly detection.

C.4.5 Algorithms' selection

Based on the initial requirements analysis, 20 algorithms were preselected among those available (or added) in the TimeEval framework that at least partially meet all primary requirements. Table 14 summarizes the detailed requirements analysis for those algorithms. Some examples of partially fulfilled requirements are for algorithms that R1) do not provide dedicated thresholding mechanisms, R2) technically allow for the online detection but with a large computational overhead, R4) handle anomalies in training data but cannot learn from them, R5) would need additional mechanisms or modifications of external libraries (i.e., PyOD (Zhao et al., 2019)) to provide a list of affected channels, R7) give only a theoretical option to learn rare nominal events, or R9) are only possible to run for the lightweight subsets of channels (i.e., Windowed iForest and KNN). None of the preselected algorithms are able to explicitlyearn rare nominal events (R7) or handle varying sampling rates (R8).

Based on the detailed analysis of the requirements, eight algorithms of various types were selected for ESA-ADB, five unsupervised – principal components classifier (PCC) (Shyu et al., 2003), histogram-based outlier score (HBOS) (Goldstein and Dengel, 2012), isolation forest (iForest) (Liu et al., 2008), k-nearest neighbours (KNN) (Ramaswamy et al., 2000), and three semi-supervised ones – global standard deviations from nominal (GlobalSTD), Telemanom (Hundman et al., 2018), and DC-VAE (González et al., 2023). The selected unsupervised algorithms have several important limitations in terms of TSAD. They may be give suboptimal results because of the assumptions of independence of samples and identical fractions of anomalies in training and test data (they fulfil R4 because they learn contamination levels from the training data). They only give global scores, so it is impossible to calculate subsystem-aware and channel-aware scores for them. They also do not support non-target channels and telecommands on input, so this information was not used. However, they establish a baseline for more advanced algorithms.

Among the rejected ones, Matrix Profile-based methods like DAMP (Lu et al., 2023b) or MADRID (Lu et al., 2023a) seem to be promising candidates due to their outstanding speed, high interpretability, and a theoretical possibility to memorize rare nominal events. However, they would need a special adaptation to support multidimensional data (Yeh et al., 2017), they do not give option to annotate known anomalies in training data, and their implementations in Matlab pose several technical and licensing problems when integrated with TimeEval. The COPOD algorithm does not fulfil R9 after adapting it to online detection required by R2. LOF (Breunig et al., 2000), k-Means (Yairi et al., 2001), Torsk (Heim and Avery, 2019), and RobustPCA (Paffenroth et al., 2018) showed very poor results in initial experiments. All semi-supervised algorithms that only partially fulfil R9 were rejected. The published code contains implementations of all methods listed in Table 14.

C.4.6 Algorithms' parametrization

To support the full reproducibility of our results, Table 15 lists all the algorithms' parameters and their values used in our experiments. The parameters' names directly correspond to the published code based on the TimeEval framework (Wenig et al., 2022). They use default values or settings recommended by algorithms' authors, sometimes adjusted to the specific features of our datasets (boldfaced in the table).

Table 14: Comparison of algorithms (1 = success, 0.5 = partial, 0 = failure) across R1–R9 criteria and inclusion in ESA-ADB.

Algorithm	R1	R2	R3	R4	R5	R6	R7	R8	R9	In ESA-ADB
UNSUPERVISED										
COPOD (Li et al., 2020)	1	1	0.5	0.5	1	0	0	0	0.5	NO
HBOS (Goldstein and Dengel, 2012)	1	0	1	0.5	0.5	0	0	0	1	YES
iForest (Liu et al., 2008)	1	1	1	0.5	0.5	0	0	0	1	YES
Windowed iForest (Liu et al., 2008)	1	1	1	0.5	0.5	0	0	0	0.5	SUBSETS
k-Means (Yairi et al., 2001)	1	1	1	0.5	0.5	0	0	0	0.5	NO
KNN (Ramaswamy et al., 2000)	1	1	1	0.5	0.5	0	0.5	0	0.5	SUBSETS
LOF (Breunig et al., 2000)	1	1	1	0.5	0.5	0	0	0	0.5	NO
Matrix Profile (Lu et al., 2023b,a)	1	0.5	1	0	0.5	0	0.5	0	1	NO
PCC (Shyu et al., 2003)	1	1	0.5	0.5	0.5	0	0	0	1	YES
Torsk (Heim and Avery, 2019)	0.5	1	1	0.5	1	0	0	0	0.5	NO
DAE (Sakurada and Yairi, 2014)	0.5	1	1	0	1	0	0	0	0.5	NO
DC-VAE (González et al., 2023)*	0.5	1	1	0	1	0	0	0	0.5	NO
DC-VAE-ESA*	1	1	1	0.5	1	0	1	0	1	YES
SEMI-SUPERVISED										
GlobalSTD*	1	0	1	0.5	1	0	0	0	1	YES
Hybrid KNN (Song et al., 2017)	1	1	1	0	0.5	0.5	0	0	0.5	NO
LSTM-AD (Malhotra et al., 2015)	0.5	1	1	0	1	0	0	0	0.5	NO
OmniAnomaly (Su et al., 2019)	0.5	1	1	0	0.5	0	0	0	0.5	NO
RobustPCA (Paffenroth et al., 2018)	0.5	1	0.5	0.5	0.5	0	0	0	1	NO
Telemanom (Hundman et al., 2018)	1	1	1	0	1	0	0	0	0.5	NO
Telemanom-ESA*	1	1	1	0.5	1	0	1	0	1	YES

The number of 50 bins in HBOS was arbitrarily selected based on the analysis of the histograms of channels because the default value of 10 seemed to be much too small for our dataset. The default window size in Windowed iForest was decreased from 100 to 17 to avoid out-of-memory errors for our datasets. Many parameters of DC-VAE were adjusted to our dataset. The scaling is not used because it is already present in our preprocessing. Outliers are not rejected (*wo_outliers* is False) because our preprocessing code removes known anomalies. The window size is increased to 256 to be similar to the default Telemanom's window size (250). Also, the value of 256 showed good results on similar data in the original DC-VAE paper (González et al., 2023). The number of CNN units is decreased from the default 64 to 32 because a significant overfitting was noticed in the validation scores for 64 units. The latent space dimensionality depends on the number of input channels in the same way as suggested for the TELCO dataset in the original DC-VAE code. The two main changes to Telemanom are 1) the increased number of units for full set training sets depending on the total number of input and output channels, and 2) the new min_error_value parameter to avoid magic numbers in the Telemanom code. The default value of the min_error_value is set to 0 (no magic numbers), but for Telemanom-ESA-Pruned it is arbitrarily selected to be 0.007 based on a manual analysis of reconstruction errors for the validation set, since the default value of 0.05 was much too high for some channels.

Importantly, the number of batches per epoch was limited to 1000 to avoid extremely long epoch training times for our datasets and to provide frequent validation score updates. Thus, the number of (sub)epochs was increased tenfold to 1000, and the early stopping patience was doubled to 20 for both DC-VAE-ESA and Telemanom-ESA to compensate for this.

Table 15: Parametrization of algorithms used in ESA-ADB. Boldfaced parameters mark values different from the default ones.

Algorithm	**Parameter name**	**Value(s)**
PCC	max_iter	None
	n_components	None
	n_selected_components	None
	random_state	42
	svd_solver	auto
	tol	0.0
	whiten	False
HBOS	**n_bins**	**50**
	alpha	0.1
	bin_tol	0.5
	random_state	42
iForest	n_trees	100
	bootstrap	False
	max_features	1.0
	max_samples	None
	random_state	42
Windowed iForest	**n_trees**	**200**
	window_size	**17**
	bootstrap	False

Continued on the next page

Algorithm	Parameter name	Value(s)
	max_features	1.0
	max_samples	None
	random_state	42
KNN	distance_metric_order	2
	leaf_size	30
	method	Largest
	n_neighbors	5
GlobalSTD	tol	**3 (STD3) and 5 (STD5)**
	random_state	42
DC-VAE-ESA	**alpha**	**3 (STD3) and 5 (STD5)**
	T (window size)	**256**
	cnn_units	**32 (16 for Phase 1)**
	dil_rate	[1,2,4,8,16,32,64]
	kernel	2
	strs (stride length of CNN layers)	1
	batch_size	64
	J (latent space dimensionality)	**1/3 × total number of input channels and telecommands**
	epochs	**1000**
	lr (learning rate)	10^{-3}
	seed	123
	early_stopping_delta	0.001
	early_stopping_patience	**20**
Telemanom-ESA	batch_size	70
	dropout	0.3
	early_stopping_delta	0.0003
	early_stopping_patience	**20**
	epochs	**1000**
	error_buffer	100
	layers	2
	number of units per layer	**80 for lightweight subsets, 134 for full sets of channels.**
	lstm_batch_size	64
	min_error_value (newly introduced to avoid magic numbers)	**0 (0.007 for Telemanom-ESA-Pruned)**
	prediction_window_size	10
	random_state	42
	smoothing_perc	0.05
	smoothing_window_size	30
	window_size	250

Appendix D. Benchmarking results

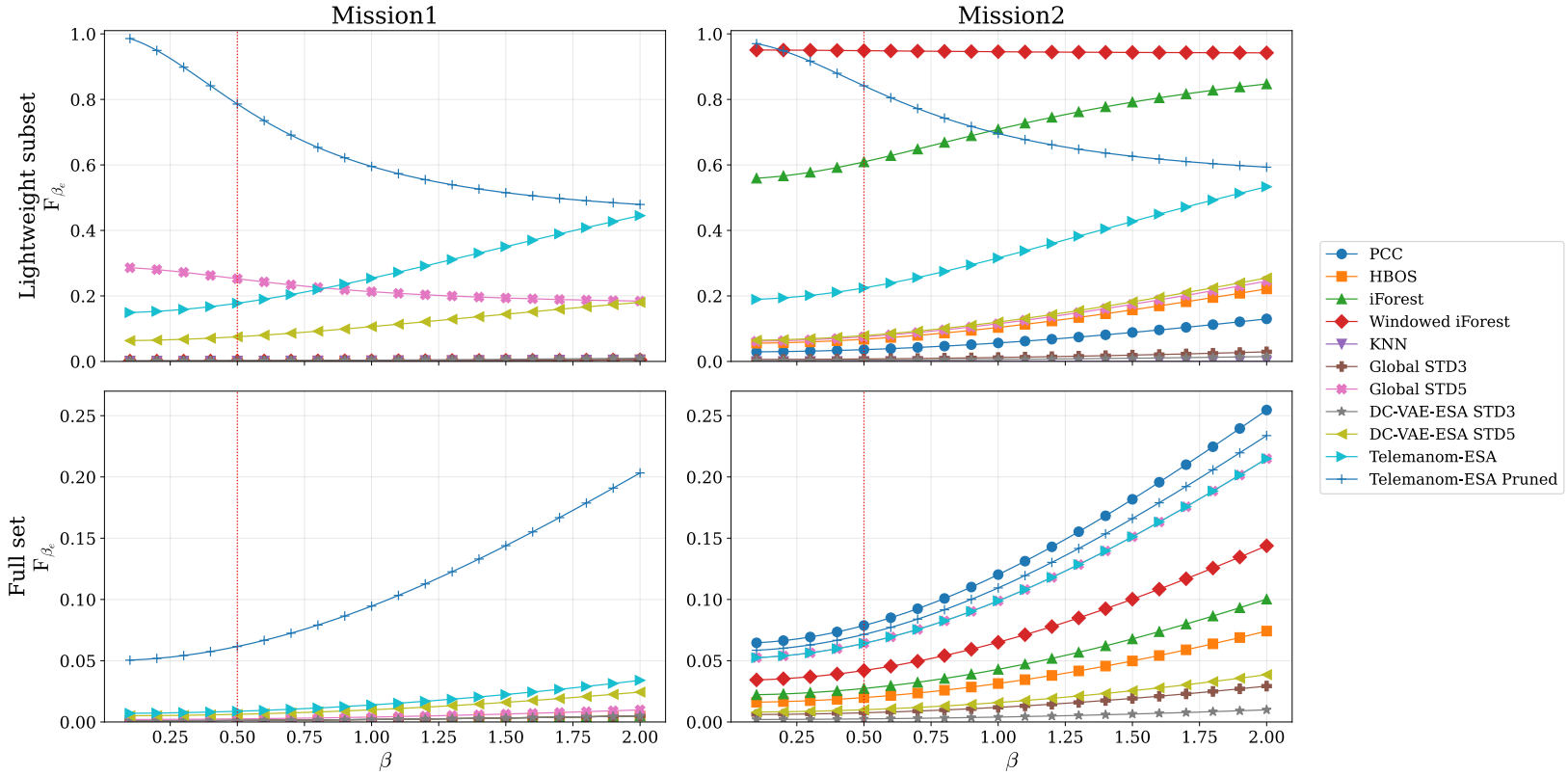

Figure 17: Analysis of F_{β_e} for different β across all algorithms in the benchmark. The selected $\beta = 0.5$ is marked with red dotted vertical line.

Table 16: Benchmarking results for detection of all events (excluding communication gaps) for Mission1. Boldfaced results indicate the best values among all algorithms (excluding After ratio of ADTQC which is just a helper value).

Mission1 – trained and tested on the lightweight subset of channels 41-46

Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	<0.001	<0.001	<0.001	<0.001	<0.001	0.001	0.288	0.002	0.063	0.148	**0.999**
	Recall	0.554	0.585	0.585	0.738	0.754	0.431	0.169	0.554	0.338	**0.894**	0.424
	F0.5	<0.001	<0.001	<0.001	<0.001	<0.001	0.001	0.253	0.003	0.075	0.178	**0.786**
Channel-aware	Precision	Not available for unsupervised algorithms					0.431	0.169	0.550	0.338	**0.894**	0.424
	Recall						0.285	0.159	0.463	0.221	**0.738**	0.275
	F0.5						0.351	0.167	0.514	0.283	**0.837**	0.362
Alarming	Precision	0.033	0.047	0.017	0.015	0.017	0.057	0.035	0.070	0.028	0.868	**0.875**
ADTQC	After ratio	0.833	0.763	0.711	0.375	0.612	0.929	0.909	0.972	0.955	0.136	0.143
	Score	0.840	0.781	0.784	0.563	0.803	0.770	0.688	**0.901**	0.803	0.428	0.197
Affiliation-based	Precision	0.535	0.543	0.543	0.599	0.522	0.559	0.699	0.584	**0.780**	0.727	0.711
	Recall	0.334	0.352	0.357	0.424	0.322	0.375	0.422	0.377	0.593	**0.662**	0.423
	F0.5	0.477	0.490	0.492	0.553	0.464	0.509	0.618	0.526	**0.734**	0.713	0.626

Mission1 – trained and tested on the full set of channels

Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	<0.001	<0.001	<0.001			<0.001	0.002	<0.001	0.005	0.007	**0.050**
	Recall	0.870	0.957	**0.967**			0.848	0.761	0.924	0.804	0.946	0.870
	F0.5	<0.001	<0.001	<0.001			<0.001	0.003	<0.001	0.007	0.008	**0.061**
Subsystem-aware	Precision	Not available					0.520	**0.728**	0.526	0.640	0.676	0.395
	Recall						0.694	0.538	0.764	0.670	0.859	**0.861**
	F0.5						0.528	0.664	0.538	0.623	**0.689**	0.436
Channel-aware	Precision	Not available			OOM	OOM	0.380	0.276	0.398	0.359	**0.514**	0.267
	Recall						0.292	0.208	0.414	0.266	0.569	**0.725**
	F0.5						0.325	0.241	0.350	0.282	**0.477**	0.291
Alarming	Precision	0.003	0.002	0.001			0.004	0.049	0.002	0.017	0.074	**0.206**
ADTQC	After ratio	0.613	0.443	0.438			0.718	0.743	0.647	0.716	0.322	0.463
	Score	0.642	0.603	0.685			0.723	0.691	**0.752**	0.692	0.673	0.684
Affiliation-based	Precision	0.563	0.539	0.538			0.560	0.575	0.559	0.578	0.545	**0.649**
	Recall	0.522	**0.578**	0.456			0.492	0.462	0.476	0.511	0.368	0.484
	F0.5	0.554	0.547	0.519			0.545	0.548	0.540	0.563	0.497	**0.607**

Table 17: Benchmarking results for detection of all events (excluding communication gaps) for Mission2. Boldfaced results indicate the best values among all algorithms (excluding After ratio of ADTQC which is just a helper value).

Mission2 – trained and tested on the lightweight subset of channels 18–28

Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	0.029	0.055	0.557	0.951	**0.951**	0.006	0.061	0.063	0.064	0.063	0.978
	Recall	**1.000**	0.911	0.974	0.940	1.000	1.000	1.000	1.000	1.000	0.986	0.540
	F0.5	0.036	0.068	0.609	0.949	0.001	0.007	0.075	0.079	0.224	0.809	**0.842**
Channel-aware	Precision	Not available for unsupervised algorithms					0.951	0.992	0.904	0.995	0.831	0.465
	Recall						0.462	0.372	0.554	0.451	0.870	**0.384**
	F0.5						0.767	0.723	0.787	0.783	**0.822**	0.442
Alarming	Precision	0.061	0.105	0.075	0.217	0.060	0.054	0.061	0.052	0.068	**0.912**	0.862
ADTQC	After ratio	**0.983**	0.994	1.000	0.948	0.391	0.946	0.989	0.991	0.987	0.997	0.997
	Score	0.999	0.990	0.991	0.985	0.724	0.997	0.997	0.994	0.994	0.987	**0.997**
Affiliation-based	Precision	0.890	0.936	**0.982**	0.968	0.561	0.740	0.935	0.939	0.688	0.759	0.873
	Recall	0.580	0.867	0.952	0.925	0.243	0.296	0.717	0.939	0.688	0.759	**0.873**
	F0.5	0.804	0.921	**0.976**	0.959	0.445	0.612	0.881	0.881	0.733	0.799	0.881

Mission2 – trained and tested on the full set of channels

Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	**0.064**	0.016	0.022	0.034		0.006	0.052	0.002	0.008	0.052	0.058
	Recall	**0.997**	0.820	0.903	0.746		**0.997**	0.992	**0.997**	0.994	0.992	0.964
	F0.5	**0.079**	0.020	0.027	0.042		0.008	0.064	0.002	0.011	0.064	0.071
Subsystem-aware	Precision	Not available					0.910	**0.983**	0.672	0.911	0.409	0.258
	Recall						0.965	0.951	0.967	0.952	**0.984**	0.896
	F0.5						0.910	**0.969**	0.699	0.907	0.451	0.298
Channel-aware	Precision	Not available				OOM	0.899	**0.941**	0.774	0.931	0.584	0.326
	Recall						0.550	0.490	0.592	0.507	0.783	**0.823**
	F0.5						**0.791**	0.787	0.713	0.783	0.592	0.368
Alarming	Precision	0.094	0.148	0.112	0.179		0.060	0.077	0.066	0.083	0.771	**0.790**
ADTQC	After ratio	0.975	0.906	0.939	0.852		0.925	0.989	0.663	0.930	0.104	0.274
	Score	0.990	0.939	0.967	0.928		0.983	**0.993**	0.825	0.985	0.513	0.648
Affiliation-based	Precision	0.814	0.570	0.621	0.608		0.680	**0.904**	0.603	0.859	0.586	0.591
	Recall	**0.631**	0.455	0.499	0.474		0.280	0.628	0.324	0.625	0.348	0.347
	F0.5	0.770	0.543	0.592	0.575		0.529	**0.831**	0.515	0.799	0.516	0.518

D.1 The effect of β on the corrected event-wise F_β-score

As shown in Figure 17, *Telemanom-ESA Pruned* consistently achieves the best performance for Mission1 across all β values and channel subsets (i.e., both lightweight and full). Moreover, its non-pruned variant ranks as the 2^{nd} best method for $\beta \geq 0.9$. For Mission2, β does not affect the ranking when using the full channel set. However, for the lightweight subset, *Telemanom-ESA Pruned* outperforms *Windowed iForest* only at the lowest value, $\beta = 0.1$. For $\beta \geq 1$, it is outperformed even by the non-windowed iForest variant.

D.2 High-dimensional false alarm amplification issue

A key phenomenon observed in the ESA-ADB benchmark is the amplification of false alarms as the dimensionality of the monitored system increases (e.g., from the lightweight to full set of channels). Figure 18 provides a more detailed view on this effect by showing how the number of event-wise false positives changes when we aggregate detections and compare them against ground truth for specific number of channels (random selection, repeated 100 times). It shows an approximately logarithmic growth in the number of false positives when dimensionality increases.

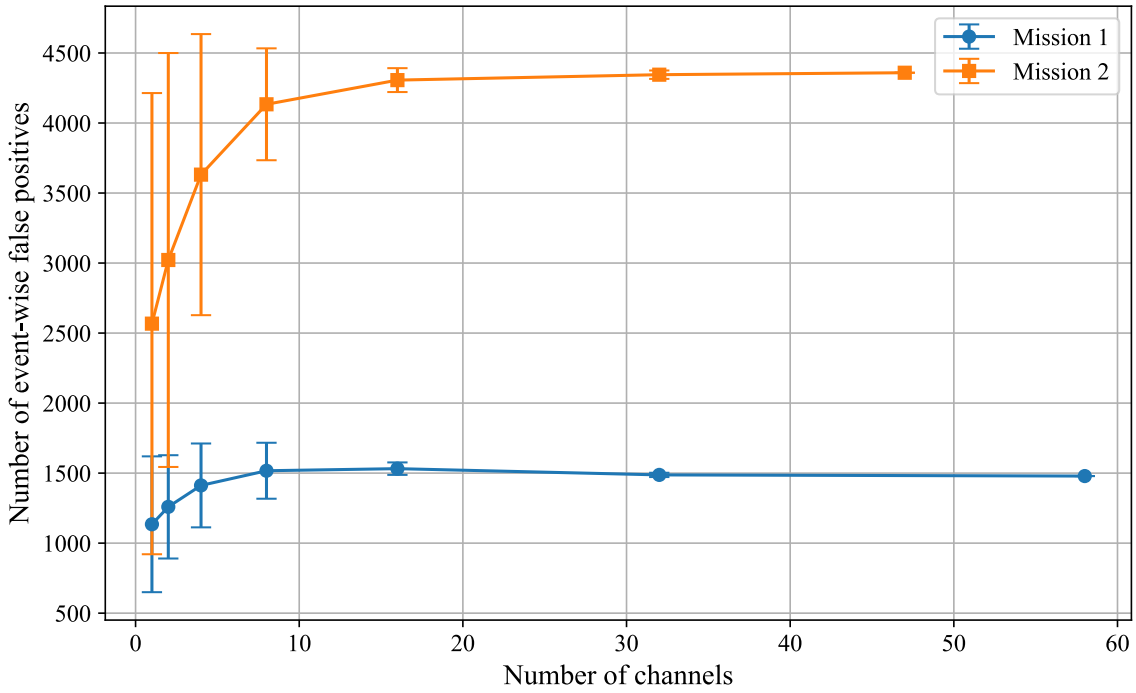

Figure 18: The false alarm amplification effect with the growing number of analyzed channels in ESA-ADB.

This effect can be understood through several complementary theoretical views. First, the probability of false positives grows with the number of channels N. If each channel has

a false positive rate p_{fp}, then the system-level false alarm probability under mild channel independence assumption becomes:

$$P(\text{FP}_{\text{system}}) = 1 - (1 - p_{\text{fp}})^N,$$

which quickly approaches 1 when $N \to \infty$, even for small p_{fp}.

Second, many anomaly detection methods rely on distances (e.g., kNN or iForest) or reconstruction errors (e.g., Telemanom or DC-VAE), both of which are affected by the shrinking contrast between normal and anomalous regions as dimensionality increases. As a result, small fluctuations in individual channels are more likely to be interpreted as anomalies when aggregated across many dimensions.

Third, real-world telemetry exhibits heterogeneous noise, non-stationarity, and temporal dependencies which exacerbate the issue. When models assume simplified distributions or independence, residual errors accumulate across channels, effectively inflating anomaly scores and increasing the likelihood of false alarms.

Overall, high-dimensional false alarm amplification is an inherent challenge in real-world multivariate anomaly detection. It highlights the importance of dimensionality-aware modeling, channel selection (as in the lightweight subset), and robust aggregation strategies to control system-level false positive rates.

D.3 Example detections

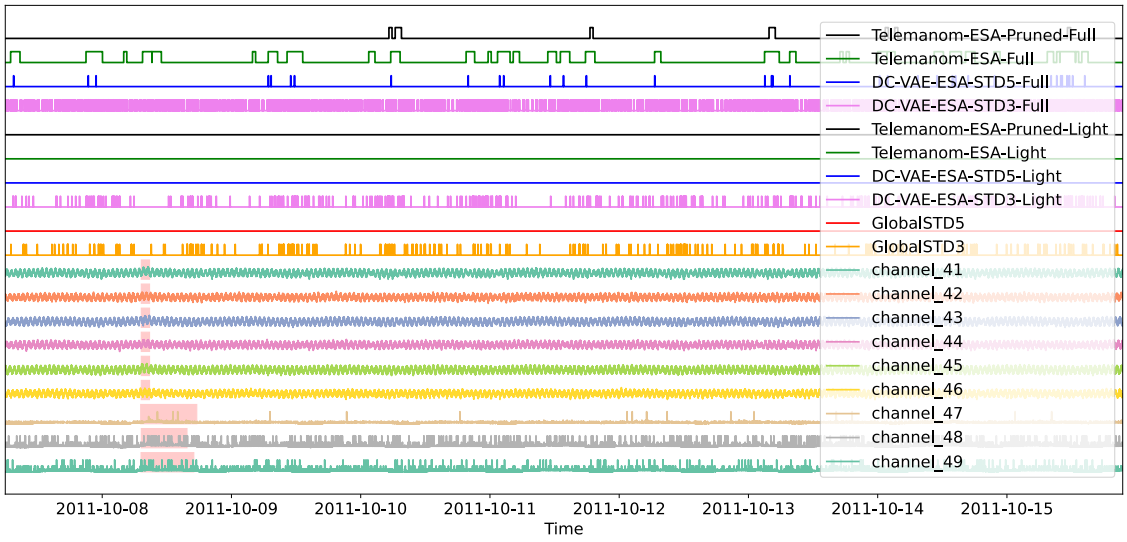

Figure 19: Detections of rare nominal event id_49 (marked in red) for Mission1. It is not detected when using only the lightweight subset of channels 41–46. For the full set, only Telemanom-ESA shows a reasonable detection, but it is surrounded by many false detections.

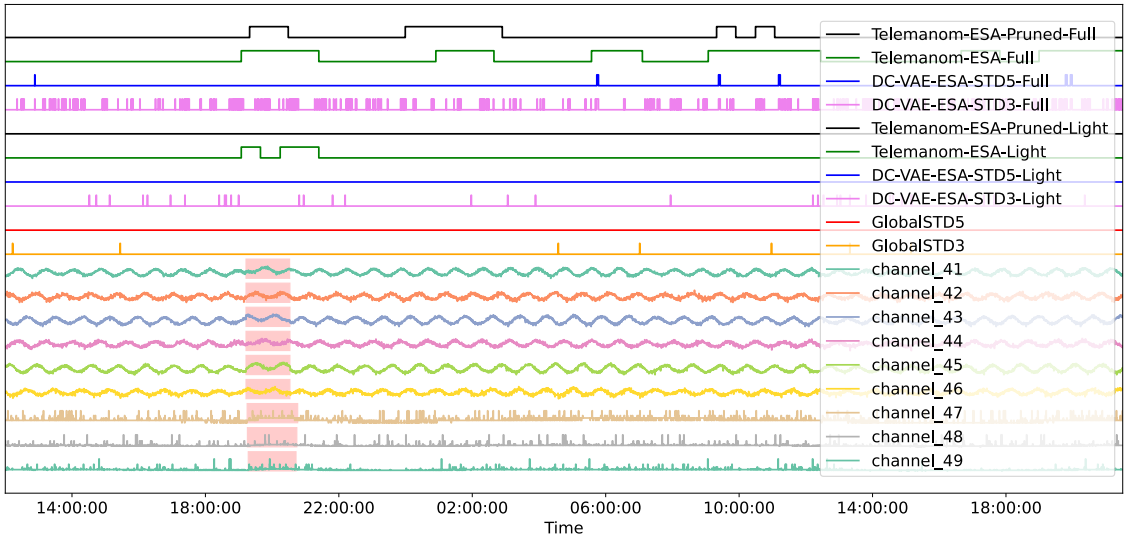

Figure 20: Detections of rare nominal event id_51 (marked in red) for Mission1. It is reasonably detected only by Telemanom-ESA. Surprisingly, Telemanom-ESA trained on the lightweight subset was also able to detect this.

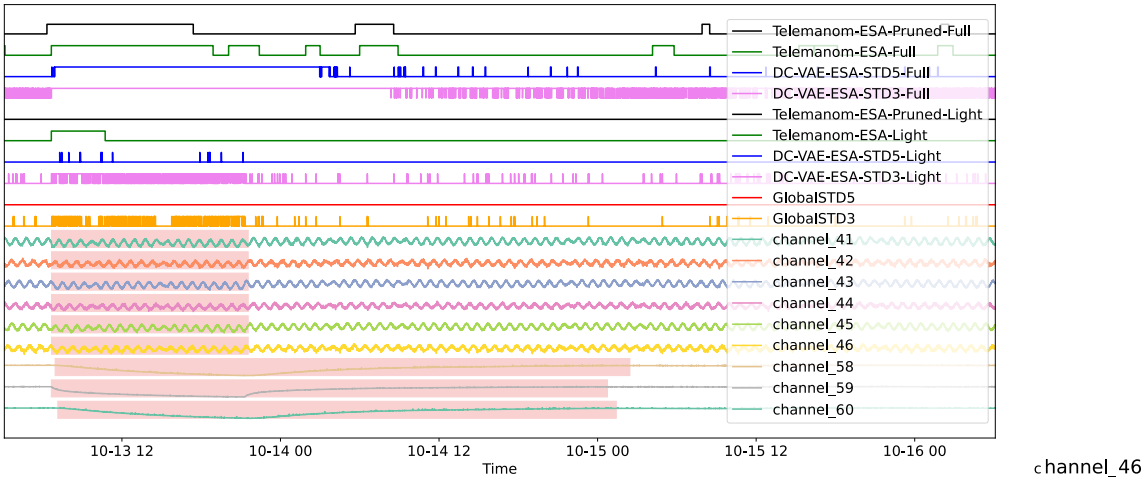

Figure 21: Detections of anomaly id_138 (marked in red) for Mission1. It is clearly visible in channels 58–60, so it is detected well by models trained on full sets of channels. However, it is not so easy using only the lightweight subset, i.e., Telemanom-ESA-Pruned-Light shows no response.

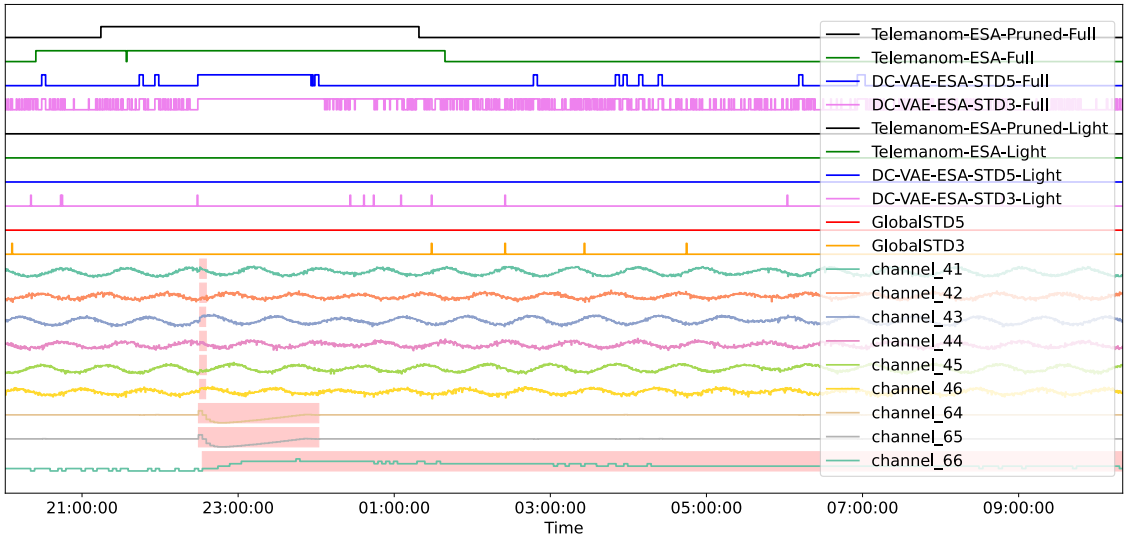

Figure 22: Detections of anomaly id_153 (marked in red) for Mission1. It is not detected when using only the lightweight subset of channels 41–46. For the full set, it is detected by all algorithms. Telemanom-ESA-Full detects it too early.

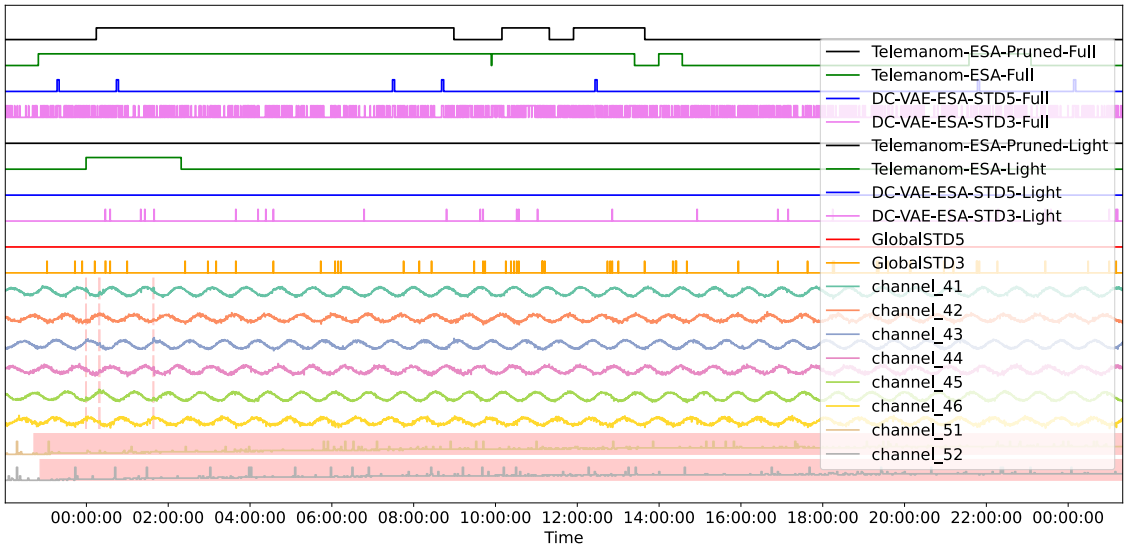

Figure 23: Detections of rare nominal event id_155 (marked in red) for Mission1. Only Telemanom-ESA was able to correctly detect this event in both lightweight and full sets of channels.

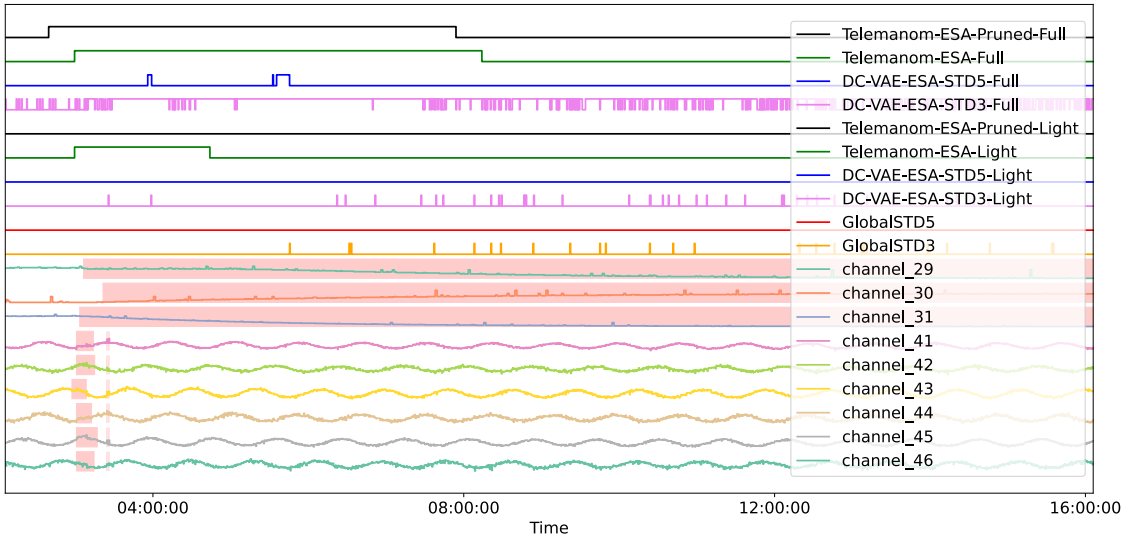

Figure 24: Detection of rare nominal event id_159 (marked in red) for Mission1. Only Telemanom-ESA was able to correctly detect this event in both lightweight and full sets of channels, with a good timing. DC-VAE-ESA-STD3-Full also seems to detect it relatively well.

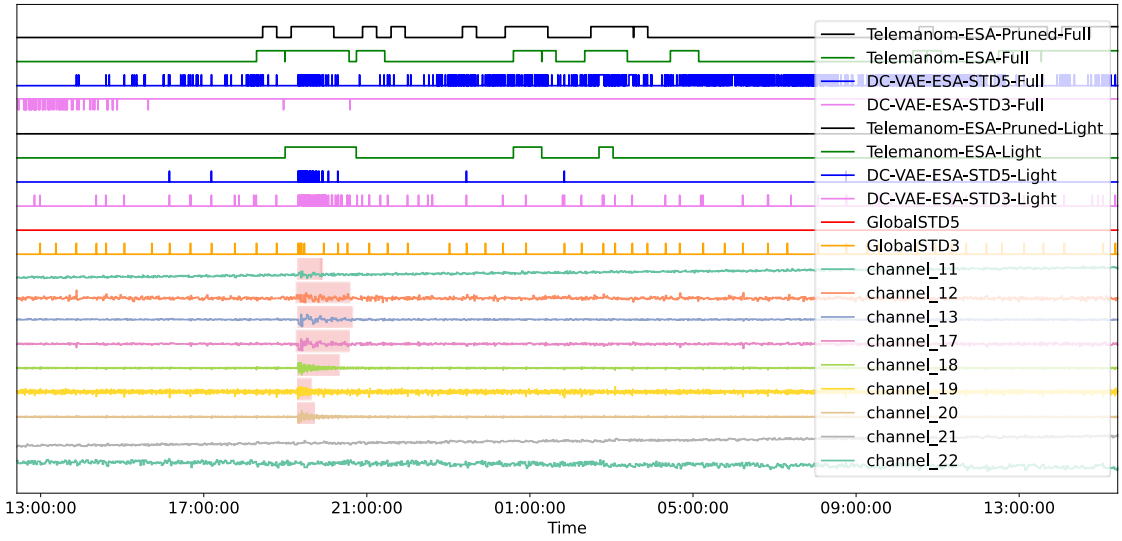

Figure 25: Detection of anomaly id_631 (marked in red) for Mission2. This anomaly is not so easy to spot manually but was detected by most algorithms, surprisingly, not by Telemanom-ESA-Pruned-Light.

D.4 Results for anomalies only

The analysis of the results for anomalies alone (excluding rare nominal events and communication gaps) in Tables 18 and 19 are important for understanding the performance of the algorithms in detecting the actual anomalies desired by SOEs. In this analysis, any true positives, false positives, or false negatives related to events different than anomalies are ignored (see implementation details in Appendix Section C.2.2). For Mission2, there are only 9 anomalies in the full test set and only 4 anomalies in the lightweight test set (see Table 22), so the results should be interpreted with caution. A more reliable analysis can be conducted for Mission1 with 55 and 29 anomalies, respectively (see Table 21).

Table 18: Benchmarking results for detection of anomalies only for Mission1. Boldfaced results indicate the best values among all algorithms (excluding After ratio of ADTQC which is just a helper value).

Mission1 – trained and tested on the lightweight subset of channels 41–46 – only anomalies

Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.205	<0.001	0.021	0.074	**0.999**
	Recall	0.310	0.379	0.414	0.552	0.448	0.310	0.241	0.310	0.241	**0.931**	0.862
	F0.5	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.211	<0.001	0.026	0.090	**0.968**
Channel-aware	Precision	Not available for unsupervised algorithms					0.310	0.241	0.302	0.241	**0.931**	0.529
	Recall						0.282	0.241	0.285	0.241	**0.882**	0.862
	F0.5						0.293	0.241	0.297	0.241	**0.914**	0.722
Alarming	Precision	0.102	0.054	0.444	**0.889**	0.120	0.034	0.024	0.048	0.010	0.818	0.862
ADTQC	After ratio	0.889	0.636	0.500	0.063	0.385	0.889	**1.000**	**1.000**	**1.000**	0.037	0.040
	Score	0.826	0.676	0.730	0.308	0.670	0.826	0.919	0.911	**0.921**	0.220	0.159
Affiliation-based	Precision	0.536	0.543	0.532	0.562	0.521	0.561	0.919	0.559	0.906	0.774	**0.927**
	Recall	0.276	0.352	0.294	0.366	0.271	0.335	0.854	0.279	0.850	0.673	**0.859**
	F0.5	0.451	0.490	0.458	0.508	0.440	0.494	0.906	0.466	0.894	0.752	**0.912**

Mission1 – trained and tested on the full set of channels – only anomalies

Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	<0.001	<0.001	<0.001			<0.001	0.001	<0.001	0.003	0.004	**0.032**
	Recall	0.891	0.964	**0.945**			0.873	0.818	0.891	0.818	0.945	0.909
	F0.5	<0.001	<0.001	<0.001			<0.001	0.002	<0.001	0.004	0.005	**0.039**
Subsystem-aware	Precision	Not available					0.491	**0.782**	0.424	0.648	0.712	0.355
	Recall						0.721	0.676	0.739	0.721	0.855	**0.909**
	F0.5						0.507	**0.748**	0.448	0.644	0.717	0.397
Channel-aware	Precision	Not available			OOM	OOM	0.327	0.355	0.272	0.311	**0.497**	0.195
	Recall						0.332	0.298	0.398	0.324	0.561	**0.705**
	F0.5						0.309	0.315	0.272	0.291	**0.472**	0.217
Alarming	Precision	0.005	0.008	0.005			0.009	0.088	0.003	0.020	0.132	**0.278**
ADTQC	After ratio	0.633	0.415	0.423			0.708	0.756	0.673	0.733	0.327	**0.380**
	Score	0.611	0.553	0.633			0.728	0.654	**0.730**	0.654	0.561	0.536
Affiliation-based	Precision	0.527	0.512	0.501			0.521	0.531	0.512	0.531	0.512	**0.611**
	Recall	0.486	**0.563**	0.445			0.462	0.434	0.452	0.473	0.344	0.436
	F0.5	0.519	0.521	0.489			0.508	0.508	0.499	0.518	0.467	**0.566**

Table 19: Benchmarking results for detection of anomalies only for Mission2. Boldfaced results indicate the best values among all algorithms (excluding After ratio of ADTQC which is just a helper value).

Mission2 – trained and tested on the lightweight subset of channels 18–28 – only anomalies												
Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	<0.001	0.000	0.004	0.000	<0.001	<0.001	<0.001	<0.001	<0.001	**0.001**	0.000
	Recall	**1.000**	0.000	**1.000**	0.000	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.000
	F0.5	<0.001	0.000	**0.005**	0.000	<0.001	<0.001	<0.001	<0.001	<0.001	0.001	0.000
Channel-aware	Precision	Not available for unsupervised algorithms					**1.000**	**1.000**	**1.000**	**1.000**	0.600	0.000
	Recall						0.667	0.667	**1.000**	0.667	**1.000**	0.000
	F0.5						0.909	0.909	**1.000**	0.909	0.652	0.000
Alarming	Precision	0.032	0.000	0.143	0.000	0.029	0.026	0.036	0.027	0.037	**1.000**	0.000
ADTQC	After ratio	**1.000**	–	**1.000**	–	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.000	–
	Score	**1.000**	–	**1.000**	–	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.358	–
Affiliation-based	Precision	0.845	0.500	**1.000**	0.500	0.705	0.826	0.894	0.816	0.950	0.781	0.500
	Recall	0.925	0.000	0.971	0.000	0.517	0.862	0.994	0.888	0.981	**1.000**	0.000
	F0.5	0.860	0.000	**0.994**	0.000	0.657	0.833	0.912	0.830	0.956	0.817	0.000
Mission2 – trained and tested on the full set of channels – only anomalies												
Metric		PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
Event-wise	Precision	0.001	<0.001	<0.001	<0.001		<0.001	0.001	<0.001	<0.001	**0.001**	0.001
	Recall	0.667	0.667	0.667	0.500		0.667	0.167	0.833	0.667	**1.000**	**1.000**
	F0.5	0.001	<0.001	<0.001	<0.001		<0.001	0.001	<0.001	<0.001	**0.001**	**0.001**
Subsystem-aware	Precision	Not available					0.167	0.000	0.333	0.333	**0.417**	0.278
	Recall						0.167	0.000	0.500	0.333	0.833	**1.000**
	F0.5						0.167	0.000	0.352	0.333	**0.452**	0.324
Channel-aware	Precision	Not available				OOM	0.083	0.000	0.095	0.111	**0.296**	0.082
	Recall						0.021	0.000	0.229	0.042	0.573	**0.833**
	F0.5						0.052	0.000	0.098	0.083	0.325	**0.096**
Alarming	Precision	0.364	0.308	0.143	0.158		0.031	**1.000**	0.026	0.040	0.375	0.462
ADTQC	After ratio	0.750	0.500	0.500	0.333		**1.000**	**1.000**	0.600	0.750	0.500	0.500
	Score	0.542	0.489	0.493	0.437		**0.992**	0.612	0.698	0.709	0.660	0.766
Affiliation-based	Precision	0.660	0.608	0.618	0.616		0.523	0.500	0.539	0.418	**0.671**	0.620
	Recall	0.380	0.333	0.358	0.355		0.318	0.000	0.522	0.398	**0.709**	0.604
	F0.5	0.575	0.522	0.539	0.537		0.466	0.000	0.536	0.414	**0.678**	0.617

D.5 Results for lightweight test sets using algorithms trained on full sets

For algorithms that provide separate anomaly scores for each channel, it is possible to limit the analysis of the global scores to an arbitrary subset of the channels used in training. It is especially useful to directly compare the results between models trained on full sets of channels and models trained only on lightweight subsets. Such a comparison is presented in Table 20 for the DC-VAE-ESA and Telemanom-ESA algorithms. GlobalSTD is omitted because its results do not depend on the number of training channels.

D.6 Results for different mission phases

It is a common practice to periodically retrain or adapt algorithms when new telemetry becomes available from satellites, especially in the presence of significant changes in operational conditions. The experiments in this section simulate such an approach in ESA-ADB to assess the robustness of algorithms to changing conditions and to identify the earliest mission phase in which reliable detectors can be trained. These aspects are crucial for the selection of algorithms in different mission phases. Some classic algorithms may perform much better than others in early mission phases when very limited data is available, but they may be overcome by deep learning techniques in late mission phases. The goal of this section is to provide a basic analysis of these aspects in ESA-ADB. For this purpose, the effect of training set size (representing different mission phases) on the corrected event-wise F0.5-score for the test set is analyzed for the lightweight subsets of each. The analysis for full sets is not conducted as the scores are very low even for the longest training set. There are 5 training set lengths (phases) proposed for Mission1 and 4 for Mission2 following the idea presented in Fig. 24. Starting from just a few percent of the mission timeline (initial phases) to 50% of the mission (the default setting in ESA-ADB). The statistics of the phases are listed in Table 21 (Mission1) and Table 22 (Mission2).

As visualized in Table 23, there is a clear correlation between the training set length and the event-wise F0.5 scores for test sets for both missions. Especially significant improvements are visible between phases 2 and 3 for Mission1 and phases 1 and 2 for Mission 2. A clear example is Windowed iForest for which the event-wise F0.5-score goes from 0.020 to 0.901 for Mission2 in the phase 2. Based on this observation, the minimal reasonable training length can be estimated to be 21 months for Mission1 and 5 months for Mission2. Surprisingly, the longest training sets do not always ensure the best results. There are some exceptions for which training on the longest training set does not give optimal results, i.e., PCC, DC-VAE-ESA, and Telemanom-ESA. We can only hypothesize what is the reason behind that, but it may be related to the concept drift present in the data.

Table 20: Benchmarking results for detection of all events in lightweight test sets in ESA-ADB by algorithms trained on lightweight and full sets of channels. Boldfaced results indicate the better value for each pair of training sets for each algorithm (excluding After ratio of ADTQC which is just a helper value).

Mission1 – tested on the lightweight test set									
Algorithm →		DC-VAE STD3		DC-VAE STD5		Telaman-ESA		Telaman-ESA-Pruned	
Trained on →		Light	Full	Light	Full	Light	Full	Light	Full
Event-wise	Precision	0.001	**0.008**	0.014	**0.216**	**0.148**	0.027	**0.999**	0.043
	Recall	**0.576**	0.167	**0.318**	0.076	**0.894**	0.439	0.424	0.848
	F0.5	0.001	**0.009**	0.017	**0.158**	**0.178**	0.033	**0.786**	0.054
Channel-aware	Precision	**0.568**	0.167	**0.318**	0.076	**0.894**	0.439	0.424	**0.833**
	Recall	**0.442**	0.101	**0.207**	0.066	**0.738**	0.328	0.275	**0.848**
	F0.5	**0.506**	0.134	**0.262**	0.071	**0.837**	0.377	0.362	**0.834**
Alarming	Precision	0.052	**0.072**	0.034	**0.119**	**0.868**	0.659	**0.875**	0.505
ADTQC	After ratio	0.921	0.909	**0.952**	0.800	0.136	0.586	0.143	0.286
	Score	**0.805**	0.607	**0.799**	0.728	0.428	**0.625**	0.197	**0.431**
Affiliation-based	Precision	**0.577**	0.562	**0.741**	0.524	**0.727**	0.616	**0.711**	0.621
	Recall	**0.373**	0.238	**0.555**	0.071	**0.662**	0.400	0.423	**0.512**
	F0.5	**0.520**	0.441	**0.694**	0.231	**0.713**	0.556	**0.626**	0.596
Mission2 – tested on the lightweight test set									
Algorithm →		DC-VAE STD3		DC-VAE STD5		Telaman-ESA		Telaman-ESA-Pruned	
Trained on →		Light	Full	Light	Full	Light	Full	Light	Full
Event-wise	Precision	**0.003**	**0.003**	**0.064**	0.017	**0.188**	0.152	**0.978**	0.268
	Recall	**1.000**	**1.000**	**1.000**	**1.000**	0.986	**0.989**	0.540	**0.911**
	F0.5	0.003	**0.004**	**0.079**	0.021	**0.224**	0.183	**0.842**	0.312
Channel-aware	Precision	0.904	**0.912**	**0.995**	0.985	0.831	**0.875**	0.465	**0.690**
	Recall	**0.554**	0.543	**0.451**	0.445	**0.870**	0.739	0.384	**0.848**
	F0.5	0.787	**0.788**	**0.783**	0.772	0.822	**0.823**	0.442	**0.708**
Alarming	Precision	0.052	**0.034**	**0.068**	0.046	**0.912**	0.907	**0.862**	0.861
ADTQC	After ratio	0.908	0.848	0.991	0.966	0.087	0.105	0.351	0.350
	Score	**0.996**	0.934	**0.997**	0.989	0.507	**0.508**	**0.757**	0.676
Affiliation-based	Precision	**0.680**	0.675	**0.939**	0.914	**0.688**	0.681	**0.759**	0.738
	Recall	0.293	**0.345**	**0.788**	0.782	**0.544**	0.503	0.530	**0.623**
	F0.5	0.538	**0.566**	**0.904**	0.884	**0.654**	0.636	0.699	**0.712**

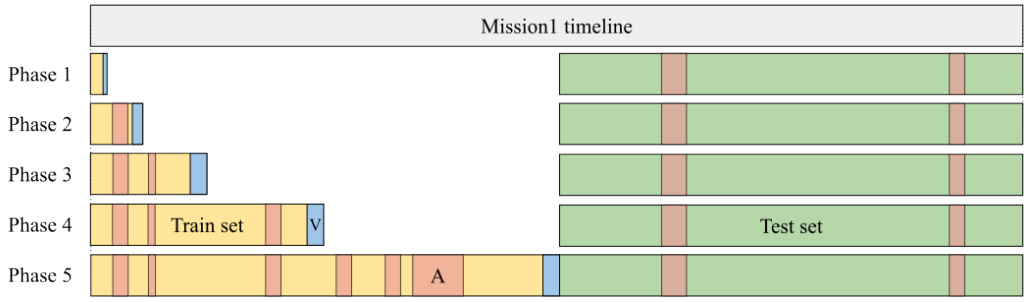

Figure 26: Illustration of the idea of mission phases for Mission1. "A" marks light red anomalous fragments and "V" marks blue validation fragments.

Table 21: Statistics of training, validation, and test sets for different phases of Mission1 considering the lightweight subset of channels (top panel) and the full set (bottom panel).

Mission1 lightweight subset	Phase 1		Phase 2		Phase 3		Phase 4		Phase 5		**Test**
	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val	
Data points	**1,125,600**	**314,399**	**3,900,977**	**997,530**	**8,900,105**	**1,479,360**	**19,171,279**	**1,463,274**	**39,774,080**	**1,479,370**	**40,925,288**
Telecommands' executions	7,769	15,918	94,426	45,194	271,882	13,295	414,927	9,001	764,648	60,157	769,917
Duration (anonymised)	9 weeks	3 weeks	8 months	2 months	18 months	3 months	39 months	3 months	81 months	3 months	84 months
Annotated points [%]	1.41	17.29	2.76	1.49	3.24	0.02	1.84	0.11	1.74	1.23	1.81
Annotated events	**1**	**1**	**6**	**3**	**17**	**2**	**28**	**1**	**52**	**3**	**65**
Anomalies	0	1	3	1	5	0	10	0	22	2	29
Rare nominal events	1	0	3	2	9	2	14	1	26	1	36
Communication gaps	0	0	0	0	3	0	4	0	4	0	0
Univariate / Multivariate	0 / 1	0 / 1	0 / 6	0 / 3	0 / 14	0 / 2	0 / 24	0 / 1	0 / 48	0 / 3	1 / 64
Global / Local	1 / 0	1 / 0	4 / 2	2 / 1	11 / 3	1 / 1	18 / 6	1 / 0	39 / 9	3 / 0	40 / 25
Point / Subsequence	0 / 1	0 / 1	0 / 6	0 / 3	0 / 14	0 / 2	0 / 24	0 / 1	1 / 47	2 / 1	9 / 56
Distinct event classes	**1**	**1**	**5**	**2**	**10**	**2**	**15**	**1**	**17**	**2**	**13**

Mission1 full set	Phase 1		Phase 2		Phase 3		Phase 4		Phase 5		**Test**
	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val	
Data points	**8,954,221**	**2,176,171**	**29,416,435**	**7,890,008**	**68,888,013**	**10,761,293**	**144,775,815**	**10,273,971**	**305,515,601**	**10,741,556**	**428,599,738**
Annotated points [%]	2.11	10.62	1.87	1.52	1.96	0.93	1.32	0.03	1.33	1.62	2.25
Annotated events	**5**	**4**	**20**	**8**	**54**	**4**	**73**	**1**	**104**	**5**	**91**
Anomalies	4	1	13	2	27	2	40	0	59	4	55
Rare nominal events	1	3	7	6	24	2	29	1	41	1	36
Communication gaps	0	0	0	0	3	0	4	0	4	0	0
Univariate / Multivariate	3 / 2	3 / 1	11 / 9	5 / 3	31 / 20	0 / 4	31 / 38	0 / 1	31 / 69	0 / 5	1 / 90
Global / Local	3 / 2	4 / 0	11 / 9	5 / 3	36 / 15	1 / 3	44 / 25	1 / 0	67 / 33	3 / 2	43 / 48
Point / Subsequence	0 / 5	0 / 4	0 / 20	0 / 8	0 / 51	0 / 4	1 / 68	0 / 1	2 / 98	2 / 3	9 / 82
Distinct event classes	**3**	**2**	**6**	**3**	**11**	**3**	**16**	**1**	**18**	**3**	**17**

Table 22: Statistics of training, validation, and test sets for different phases of Mission2 considering the lightweight subset of channels (top panel) and the full set (bottom panel). There are no communication gaps and all events are of subsequence type, so these statistics are omitted.

Mission2 lightweight subset	Phase 1		Phase 2		Phase 3		Phase 4		**Test**
	Train	Val	Train	Val	Train	Val	Train	Val	
Data points	**1,457,269**	**506,869**	**7,741,250**	**2,032,657**	**15,714,523**	**3,867,017**	**34,998,975**	**5,830,297**	**46,153,954**
Telecommands' executions	34,185	11,694	179,930	48,313	372,643	93,496	815,370	130,968	1,077,677
Duration (anonymised)	3 weeks	1 week	4 months	1 month	8 months	2 months	18 months	3 months	21 months
Annotated points [%]	0.83	0.49	2.62	2.02	1.94	3.10	3.74	1.02	2.02
Annotated events	**14**	**4**	**83**	**19**	**140**	**27**	**246**	**27**	**349**
Anomalies	0	0	2	2	11	2	18	0	4
Rare nominal events	14	4	81	17	129	25	228	27	345
Univariate / Multivariate	0 / 14	0 / 4	0 / 83	0 / 19	0 / 140	0 / 27	1 / 245	0 / 27	1 / 348
Global / Local	12 / 2	3 / 1	67 / 16	17 / 2	119 / 21	24 / 3	214 / 32	26 / 1	333 / 16
Distinct event classes	**3**	**3**	**12**	**6**	**15**	**9**	**21**	**5**	**22**

Mission2 full subset	Phase 1		Phase 2		Phase 3		Phase 4		**Test**
	Train	Val	Train	Val	Train	Val	Train	Val	
Data points	**13,914,918**	**4,841,396**	**74,356,579**	**19,067,743**	**151,093,710**	**37,624,768**	**338,658,318**	**56,746,734**	**444,603,954**
Annotated points [%]	0.20	0.12	0.64	0.51	0.50	0.59	0.66	0.21	0.54
Annotated events	**14**	**4**	**85**	**22**	**146**	**28**	**256**	**27**	**361**
Anomalies	0	0	4	5	16	3	25	0	9
Rare nominal events	14	4	81	17	130	25	231	27	352
Univariate / Multivariate	0 / 14	0 / 4	1 / 84	2 / 20	3 / 143	1 / 27	5 / 251	0 / 27	4 / 357
Global / Local	12 / 2	3 / 1	67 / 18	17 / 5	120 / 26	24 / 4	217 / 39	26 / 1	340 / 21
Distinct event classes	**3**	**3**	**14**	**8**	**18**	**10**	**24**	**5**	**26**

Table 23: The effect of mission phase on the corrected event-wise F0.5-score for selected algorithms trained and tested on the lightweight subsets of channels from missions in ESA-ADB. Boldfaced results indicate the best values among all phases.

Mission1 – trained and tested on the lightweight subset of channels 41–46											
Phase	PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
1						<0.001	0.041	<0.001	0.007	0.059	0.227
2						<0.001	0.037	<0.001	0.012	0.058	0.311
3			**<0.001**			<0.001	0.104	0.007	**0.085**	0.122	0.776
4						<0.001	0.217	**0.009**	0.030	**0.309**	0.776
5						**0.001**	**0.253**	0.003	0.075	0.178	**0.786**
Mission2 – trained and tested on the lightweight subset of channels 18–28											
Phase	PCC	HBOS	iForest	Window iForest	KNN	Global STD3	Global STD5	DC-VAE STD3	DC-VAE STD5	Teleman-ESA	Teleman-ESA-Pruned
1	<0.001	<0.001	0.006	0.020	<0.001	<0.001	<0.001	<0.001	<0.001	0.234	0.622
2	**0.062**	0.007	0.456	0.901	<0.001	0.001	0.011	<0.001	0.001	**0.259**	0.757
3	0.013	0.040	0.585	0.947	<0.001	0.006	0.014	0.001	0.009	0.253	0.731
4	0.036	**0.068**	**0.609**	**0.949**	**0.001**	**0.007**	**0.075**	**0.003**	**0.079**	0.224	**0.842**

D.7 Computational resources and limitations

Experiments were run on three different machines:

1. Nvidia Tesla T4 GPU (16 GB VRAM), Intel Xeon Gold 5222 CPU 3.80 GHz, and 64 GB RAM, (for CPU-intensive and memory-intensive algorithms)
2. Nvidia 3060 RTX GPU (6 GB VRAM), Intel i7-10870H CPU 2.20 GHz, and 32 GB RAM
3. Nvidia 3090 RTX GPU (24 GB VRAM), Intel i7-8700H CPU 3.20 GHz, and 32 GB RAM (for GPU-intensive algorithms)

Given the limited resources, there are limits to the amount of time and memory that each algorithm can run. The algorithm is rejected with an *out-of-memory error* if Machine 1 goes out of RAM. Algorithms are rejected with an *out-of-time error* if it takes more than 5 days to train or test a CPU-intensive algorithm on Machine 1, or a GPU-intensive algorithm on Machine 3.

D.8 Processing times

Algorithms for satellite telemetry monitoring must not only be accurate but also fast enough to run in real-time on computational resources available to mission control and, in the extreme case, on board satellites. We measured the times of training (Table 24) and execution (Table 25) of algorithms on our hardware resources (listed in Appendix D.7). These numbers are not directly comparable because the algorithms were run in parallel processes on different machines. They give a rough approximation of the computational burden of each algorithm based on a single run in ESA-ADB. The training and execution times do not include resampling which was done once as an intermediate step before all experiments. The resampling of the test sets took about 1.5 hours for Mission1 and around 1 hour for Mission2, both on Machine 2.

The deep learning-based Telemanom has the longest training and execution times (excluding the execution time of KNN for channels 18-28 of Mission2), but it is still fast enough to provide real-time anomaly detection in both missions using the proposed resampling (0.033 Hz for Mission1, 0.056 Hz for Mission2). The total execution time (including resampling) for the full Mission1 test set is 3.5h which is just 0.02% of the test set duration, for Mission2 it is 4.5h and 0.08%, respectively. Thus, real-time execution should be possible even for sampling rates higher than 30 Hz. Moreover, in our previous works, we have shown that Telemanom can be run in real-time on-board the OPS-SAT satellite with a limited number of channels (Nalepa et al., 2022a). The important advantage of simple algorithms is that they are very fast and their training and execution times do not grow significantly with the number of channels, so it may be feasible to retrain them frequently during a mission.

In Figure 27, we include a Pareto front analysis for $F_{0.5_e}$-scores versus execution times of algorithms. It shows which algorithms dominate in terms of the runtime-performance balance. For Mission1, both *Telemanom-ESA Pruned* and *GlobalSTD5* are in the Pareto front for both lightweight and full channel sets, together with *DC-VAE-ESA STD5* for the

full set. Mission2 is dominated by traditional algorithms such as *iForest*, *Global STD5*, and *HBOS* for the lightweight subset, and *PCC* alone for the full set.

Table 24: Training times (in seconds) of algorithms used in ESA-ADB.

Algorithm	**Mission1 train set**		**Mission2 train set**	
	Channels 41–46	Full	Channels 18–28	Full
PCC	90	143	63	75
HBOS	110	111	66	68
iForest	655	714	345	308
Windowed iForest	2833 (0.8h)	Out-of-memory	1998 (0.6h)	14585 (4h)
KNN	3844 (1h)	Out-of-memory	4754 (1.5h)	Out-of-memory
GlobalSTD	101	108	60	90
DC-VAE-ESA	13466 (3.7h)	18210 (5h)	12440 (3.5h)	4679 (1.3h)
Telemanom-ESA	13115 (3.6h)	30451 (8.5h)	19725 (5.5h)	12328 (3.5h)

Table 25: Execution times (in seconds) of algorithms used in ESA-ADB.

Algorithm	**Mission1 test set**		**Mission2 test set**	
	Channels 41–46	Full	Channels 18–28	Full
PCC	124	141	73	76
HBOS	135	137	74	76
iForest	393	369	199	174
Windowed iForest	586	Out-of-memory	381	939
KNN	1233	Out-of-memory	21673 (6h)	Out-of-memory
GlobalSTD	178	182	95	289
DC-VAE-ESA	5251 (1.5h)	6010 (1.7h)	3068 (0.9h)	7900 (2.2h)
Telemanom-ESA	6931 (1.9h)	7271 (2h)	4666 (1.3h)	11078 (3.1h)

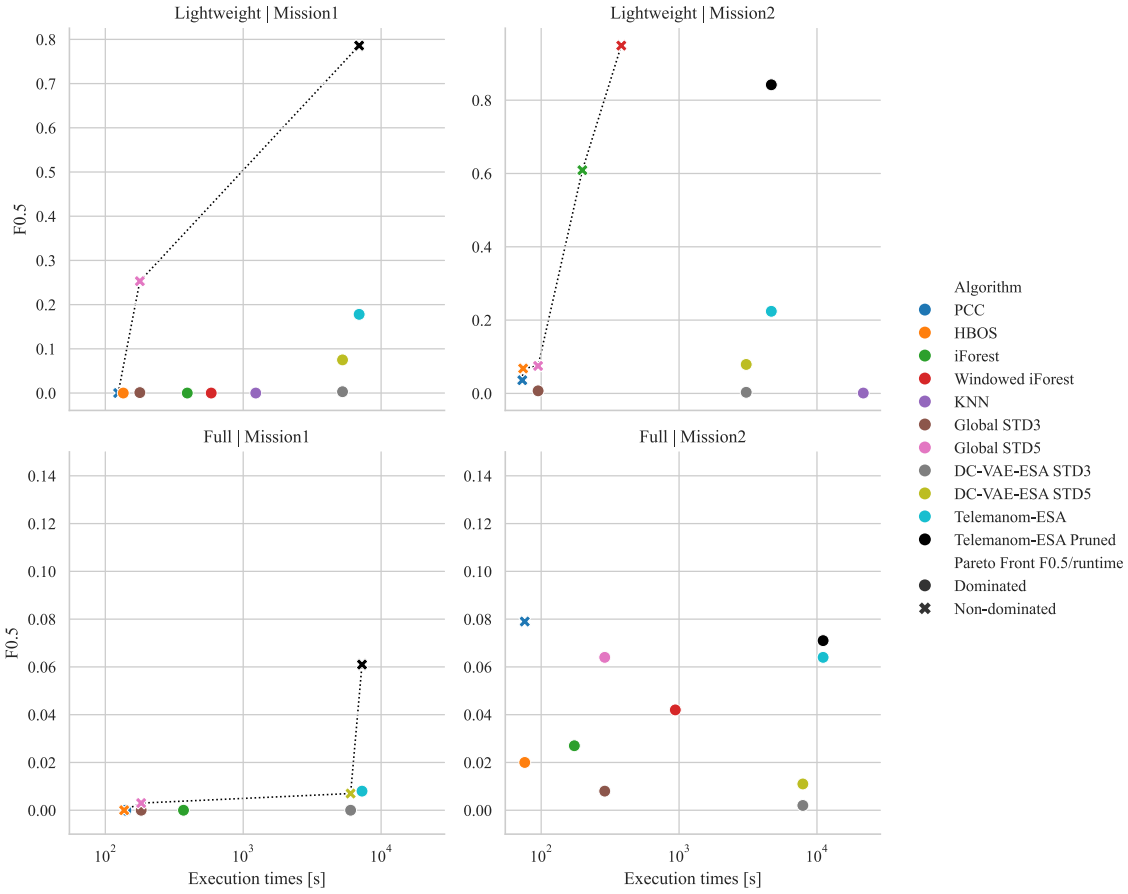

Figure 27: Pareto front analysis of the $F_{0.5_e}$-scores versus algorithm execution times for different missions and channel subsets.