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

Krzysztof Kotowski[1,*] **Christoph Haskamp**[2,*] **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 [2]Airbus Defence and Space [3]Opole University of Technology
[4]Silesian University of Technology [5]CGI [6]European Space Agency
[7]Solenix

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 Benchmark for Anomaly Detection (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 (zenodo.org/records/15237121) 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, being further explored through a dedicated Kaggle competition (kaggle.com/competitions/esa-adb-challenge).

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 [1]. However, more sophisticated anomalies are usually detected manually, which is a very expensive and error-prone task [2]. 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 [3–6], USA [7], Canada [8], Korea [9], and Japan [10], and multiple private companies [11–13]. It is also a prioritized domain of the Artificial Intelligence for Automation Roadmap of the European Space Agency (ESA) [14], and there is a growing trend in applying such systems directly onboard spacecraft for faster alarming and autonomous operations [15]. However, spacecraft telemetry is an especially complex example of multivariate time series data of high dimensionality and volume (years of recordings from up to

*Corresponding authors: kkotowski@kplabs.pl, christoph.haskamp@airbus.com, gabriele.decanio@esa.int

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

thousands of channels per spacecraft [16]), 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. [17]) 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 [2] 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 many publicly available datasets, benchmarks, and metrics for TSAD are flawed and cannot be used for an unbiased evaluation of emerging machine learning (ML) techniques, especially in complex real-world settings [18–21]. Specifically, the most popular NASA SMAP and MSL datasets of spacecraft telemetry [7] are too trivial and have unrealistic anomaly density, inconsistent ground truth, and run-to-failure bias [20]. There are a few TSAD benchmarks that avoid these flaws, but they are either univariate [20], artificial [22], or do not represent complexities of real systems (varying sampling rates, different channel types, or real-time processing). See Appendix 2.6 for detailed analysis of related datasets and benchmarks.

Contributions. The proposed **European Space Agency Benchmark for Anomaly Detection (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 [23] and adapted to be feasible for real-world time series data.

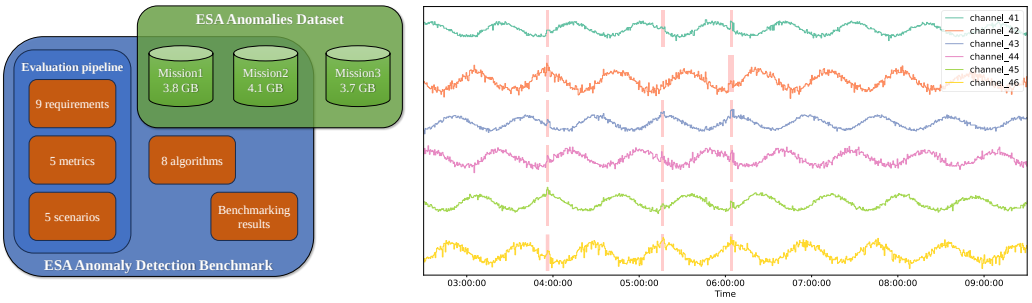

Figure 1: **Content of the ESA-ADB.** Left: Main elements of the proposed benchmark. Right: Example annotated event id_155 from Mission1 (highlighted with light red boxes).

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 (zenodo.org/records/15237121). The nomenclature (e.g., channel types, telecommands, and event categories) is explained in the first section of the Appendix.

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 2), 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) [24] and the annotations were iteratively refined with assistance of unsupervised and semi-supervised algorithms. For detailed description of the annotation process, see Appendix 2.3 and our previous related works [25, 26].

Design choices. Our dataset has several features distinguishing it from other related datasets (Appendix 2.6). 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 [27], CATS [22], and TELCO [28]. 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 1). 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 2.5).

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 2.4). 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 2.3) – 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 2). 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 [20].

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. [29]. Most annotations are categorized as multivariate global subsequences, but there is also a diverse set of other types of

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.

anomalies (Appendix 3.1), including some especially challenging ones (Appendix 2.2). 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 2.

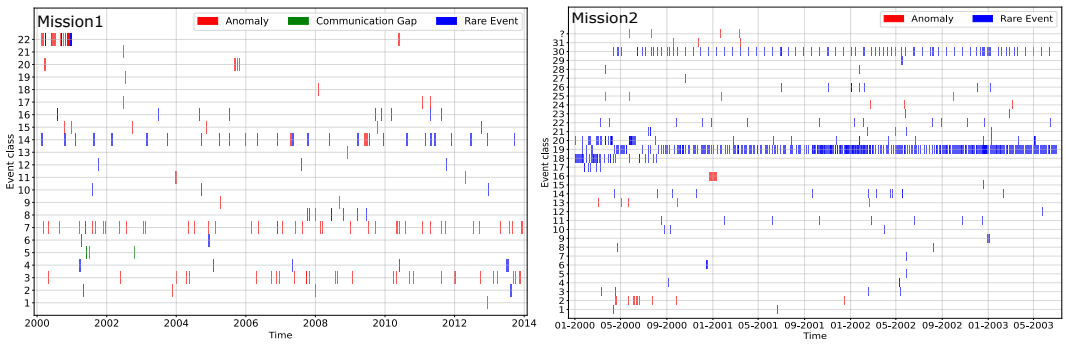

Figure 2: **Distributions of different classes and categories of events across timelines of Mission1 (left) and Mission2 (right).** The bar width corresponds to the event length, but for better visualization, the minimum width was limited to 10 and 2.5 days for Mission1 and Mission2, respectively. The question mark represents anomalies of unknown class.

2.3 Dataset split for the benchmark

Mission3 was excluded from the benchmark because of the small number and triviality of anomalies (according to Definition 1 from Wu & Keogh [20]), and many communication gaps and invalid segments (Table 1). Remaining missions 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 it is 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 the later (mature) phases of missions, where a substantial amount of telemetry data is already available

for training. However, it is also important to deploy anomaly detection systems early in the mission lifecycle. To address this, additional scenarios with shorter training periods are explored in Appendix 4.4. 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 or hundreds of channels is a very challenging task which takes a lot of computing power, so for initial experiments, familiarization, 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. This selection is subjective, but our main goal was to provide channels that are challenging for algorithms, interesting for SOEs, relatively easy to visualize and analyze manually, and not strongly dependent on other channels or subsystems. Selected channels from these subsets are presented in Figure 1 and Appendix 4.1.

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 [17, 19, 30–32]. Algorithms for our benchmark have been preselected based on the work by Schmidl et al. [17] and its corresponding TimeEval framework [23] because of the largest number of implemented algorithms (more than 70). This framework also includes the most widely used deep learning algorithm for anomaly detection in spacecraft telemetry – Telemanom by NASA [7] – 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. Based no the detailed analysis in Appendix 3.4.5, 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) [33], k-nearest neighbors (KNN) [34], principal components classifiers (PCC) [35], and isolation forest (iForest) [36] with its windowed version. The more advanced semi-supervised LSTM-based Telemanom algorithm [7] has been significantly adapted (Appendix 3.4.1) 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 3.4.2) and the Dilated Convolutional Variational AutoEncoder (DC-VAE) [28] with a series of adaptations (DC-VAE-ESA in Appendix 3.4.3), 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 [37]) 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

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.
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 4.5)	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.

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 [24, 38]. 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 3.3).

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 [39]. 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 3.3).

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 [18, 19, 21, 30, 31, 40–47]. 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 3.2.1.

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.

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

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 [7]. 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 3.2.3.

Corrected event-wise $F_{0.5}$-score. The event-wise scoring used for spacecraft telemetry by Hundman et al. [7] 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 3.2). To mitigate this, we use the correction proposed by Sehili and Zhang [18] that penalizes sample-wise (time-wise) false positives in the computation of the event-wise precision Pr_e (Equation 1):

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

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. [7] to additionally penalize false detections.

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 3.4.6). 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 4.2. The corrected event-wise scores for Missions 1 and 2 are presented in Table 4. Results for lower priority metrics are available in Appendix 4. 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 4.6.

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	0.082	0.983	0.100
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	0.014	**0.997**	0.018
Global STD5	0.002	0.761	0.003	**0.203**	0.972	**0.241**
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 4.4). 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 4.1. 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 GlobalSTD5 algorithms turned out to be the best algorithms for the lightweight and full sets, respectively. Overall, unsupervised algorithms perform relatively well for Mission2, sometimes better than the deep learning-based ones. It supports

the claim to always consider simple algorithms as a baseline [20, 48]. 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 2). However, the full set is much more challenging as reflected by much lower corrected event-wise scores. Moreover, metrics for anomalies alone (Appendix 4.2) 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 4.3 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 2.6 gives a summary of how ESA-ADB addresses common flaws listed by Wu and Keogh [20]. 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 [23], 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 [49] and transformer-based models with positional time encoding [50–52]. Beyond TSAD, the dataset also holds potential for research in time series forecasting, telemetry data compression, continual learning, and foundation models.

5 Acknowledgments

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

[1] Jose Martinez and Alessandro Donati. Novelty Detection with Deep Learning. In *2018 SpaceOps Conference*, Marseille, May 2018. American Institute of Aeronautics and Astronautics.

[2] 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. Conference Name: IEEE Transactions on Software Engineering.

[3] 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.

[4] 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.

[5] 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.

[6] 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. Publisher: Nature Publishing Group.

[7] 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.

[8] 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.

[9] 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.

[10] 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. Conference Name: IEEE Transactions on Aerospace and Electronic Systems.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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, 2023. Springer Nature Switzerland.

[16] 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. ISSN: 1095-323X.

[17] 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.

[18] 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. arXiv:2308.13068 [cs, stat].

[19] 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.

[20] 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.

[21] 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.

[22] Patrick Fleith. Controlled Anomalies Time Series (CATS) Dataset, February 2023.

[23] 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.

[24] 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 2023. Publisher: Elsevier.

[25] 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.

[26] Krzysztof Kotowski, Christoph Haskamp, Jacek Andrzejewski, Bogdan Ruszczak, Jakub Nalepa, Daniel Lakey, Peter Collins, Aybike Kolmas, Mauro Bartesaghi, Jose Martinez-Heras, and Gabriele De Canio. (In Press) The Making of the European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry. In *2025 SpaceOps Conference*, Montreal, Canada, 2025. Canadian Space Agency.

[27] 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.

[28] 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. Conference Name: IEEE Transactions on Network and Service Management.

[29] 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.

[30] 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. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[31] 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.

[32] 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.

[33] 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.

[34] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, 29(2):427–438, 2000.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] M. Shanker, M. Y. Hu, and M. S. Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, August 1996.

[40] 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 2022.

[41] 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.

[42] 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.

[43] 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.

[44] 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 2022. Number: 7.

[45] 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 2022. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[46] 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.

[47] 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.

[48] 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. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[49] 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 2023.

[50] 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.

[51] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. October 2021.

[52] 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. ISSN: 2640-3498.

A Technical Appendices and Supplementary Material

The Technical Appendix is available as a separate PDF.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: There is a dedicated paragraph in the introduction about contributions of the paper, where we did our best to reflect the scope as accurately as possible. The abstract is a concise summary of the main claims included in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a dedicated paragraph in the conclusions that discusses limitations and future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not introduce any new theorems or proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The existing TimeEval framework is used to run anomaly detection algorithms and all algorithms in the benchmark are properly referenced and/or described in the paper. All experimental procedures (preprocessing, data splits, evaluation, computational resources, and algorithms' parametrization) are described in the main text or Appendix. All data and code are publicly available under zenodo.org/records/15237121 and github.com/kplabs-pl/ESA-ADB. The GitHub repository contains all necessary instructions on how to reproduce all steps of the benchmark, from data preprocessing to final metrics' calculation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All data and code are publicly available under zenodo.org/records/15237121 and github.com/kplabs-pl/ESA-ADB. The GitHub repository contains all necessary instructions on how to reproduce all steps of the benchmark, from data preprocessing to final metrics' calculation.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: There are separate paragraphs in the paper to discuss data splits and evaluation procedures of the benchmark. Algorithm selection and parametrization is thouroughly described in the main text and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: There is a single metric value calculated on a single large continuous test set (time series) for each algorithm in the benchmark. Metrics are not calculated separately for each annotated event, so it is impossible to generate error bars or run statistical tests for them. Due to the dataset volume, it would be very computationally expensive to perform several runs of each experiment, e.g., with different seeds, to generate error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details about the computational resources and calculation times are given in the Appendix 4.5 and 4.6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research does not involve human subjects or personally identifiable and sensitive information. We do not see any potential harmful consequences of the research. Datasets are properly documented and have well-specified open licenses.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We do not see any potential malicious or unintended uses of the dataset that would have a specific societal impact. The technology supported by the benchmark has no direct impact on human subjects.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The dataset introduced in the paper is distributed under CC BY 3.0 IGO license and is an original work based on the internal data from the European Space Agency. The code of the benchmark is available under MIT license and it properly references its sources, in particular, the TimeEval framework.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper itself can be considered a documentation of the introduced assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.