Towards Adaptive and Robust Unsupervised Anomaly Detection in Satellite Telemetry

Lorenzo Brancato[1], Alessandro Lucchetti[2], Marco Giglio[3], and Francesco Cadini[4]

[1,2,3,4] *Politecnico di Milano, Department of Mechanical Engineering, Via La Masa 1, Milan, 20156, Italy*
lorenzo.brancato@polimi.it
alessandro.lucchetti@polimi.it
marco.giglio@polimi.it
francesco.cadini@polimi.it

ABSTRACT

Satellites and space systems play a vital role in space exploration, where reliability is critical due to the high cost of missions and the inability to intervene directly in case of faults. Operating in harsh and remote environments, these systems are prone to unexpected anomalies that can jeopardize mission objectives. Detecting such anomalies through telemetry data is essential but complicated by the data's complexity and the limited options for corrective action. Traditional anomaly detection methods often rely heavily on hyperparameter tuning and require large volumes of training data, making them less adaptable, data-dependent, and slower to deploy. To address these challenges, we propose a novel, plug-and-play time-series anomaly detection (TSAD) framework. Designed for robustness and fast integration, it minimizes number of hyperparameters and tuning requirements, leverages an adaptive and robust thresholding mechanism, and operates effectively with only a statistically significant volume of unlabeled data—enabling accurate, scalable anomaly detection across diverse mission scenarios.

1. INTRODUCTION

Prognostics and Health Management (PHM) is a monitoring paradigm that seeks to maximize availability and safety by rapidly detecting incipient faults, tracing their underlying causes, and forecasting the remaining useful life (RUL) of systems and components—ranging from power plants to aircraft (Lee et al., 2014; Hu, Xu, Lin, & Pecht, 2020). In typical deployments, permanently installed sensor networks stream measurements to algorithms that process data online to inform maintenance actions and operational decisions. The recent surge in data-driven approaches has been propelled by the proliferation of Internet of Things (IoT) sensors and advances in artificial intelligence (AI), enabling richer models and faster inference.

Within this landscape, machine-learning (ML)–enabled PHM is especially compelling for space missions. ML methods can digest large, heterogeneous data streams and surface actionable insights quickly—key advantages in an environment where repair is infeasible and mission costs are high. Prior work has investigated PHM for reusable liquid rocket engines (J. Wu, 2005; Tsutsumi et al., 2021) and for satellite subsystems (Fuertes et al., 2016; Ruszczak, Kotowski, Evans, & Nalepa, 2025), demonstrating the potential to extend mission lifetimes and mitigate catastrophic failures.

Satellite telemetry forms a demanding class of multivariate time series. Its distinctive properties include: (i) very high dimensionality and volume, often encompassing multi-year records across thousands of channels per spacecraft (Lakey & Schlippe, 2024); (ii) complex interdependencies among channels; (iii) heterogeneous data characteristics—such as nonuniform sampling across time and channels, gaps from idle modes or communications outages, slow degradation trends, and concept drift caused by evolving operations and mission phases; (iv) diverse variable types, spanning physical measurements, categorical states, counters, and binary telecommands; and (v) noise and measurement artifacts induced by the harsh space environment.

In routine flight operations, spacecraft operations engineers (SOEs) monitor telemetry to detect anomalies and maintain safe, continuous service for science, communications, Earth-observation, and navigation missions. Current automated support systems are often limited to thresholding or pattern matching against known signatures (Martinez, 2012; Hundman, Constantinou, Laporte, Colwell, & Soderstrom, 2018). As a result, subtle or novel anomalies frequently require manual discovery—an expensive, time-consuming process that is vulnerable to human error (Lutz & Mikulski, 2004). Consequently, major space organizations—including

the European Space Agency (ESA) (Evans et al., 2017; Martinez & Donati, 2018), the National Aeronautics and Space Administration (NASA) (Hundman et al., 2018), Centre National d'Études Spatiales (CNES) (Fuertes, Pilastre, & D'Escrivan, 2018), the German Aerospace Center (DLR) (OMeara, Schlag, Faltenbacher, & Wickler, 2016), and the Japan Aerospace Exploration Agency (JAXA) (Yairi et al., 2017)—together with industry partners such as Airbus, have intensified efforts to research, build, and validate advanced automated anomaly-detection capabilities (Pilastre, Boussouf, d'Escrivan, & Tourneret, 2020; Ruszczak et al., 2023; Tariq et al., 2019; C. Zhang et al., 2019).

To situate our work among recent advances, we group ML time-series anomaly detection (TSAD) methods into three prevalent families. (1) *Recurrent/prediction–reconstruction models* (e.g., LSTM/GRU, sequence-to-sequence and autoencoder variants) detect departures from learned dynamics or reconstruction fidelity; contemporary studies refine encoder–decoder capacity and frequency-aware modules for improved sensitivity in multivariate settings (Iqbal & Amin, 2024; Neloy & Turgeon, 2024). (2) *Transformer-based architectures* leverage self-attention to model long-range temporal and cross-channel dependencies, with variants tailored for multivariate telemetry (e.g., channel-wise attention, variable-temporal attention, stacked/forecasting Transformers) demonstrating state-of-the-art performance and improved anomaly localization (Wen et al., 2023; Kang & Kang, 2024; Kim, Kang, & Kang, 2023; Shimillas, Malialis, Fokianos, & Polycarpou, 2025). (3) *Probabilistic generative approaches* (e.g., VAEs, normalizing flows, diffusion-style models) aim to learn calibrated distributions of normal behavior for principled scoring; recent work advances autoregressive and Koopman-inspired VAEs, and broader surveys emphasize their role in uncertainty-aware detection (Zamanzadeh Darban, Webb, Pan, Aggarwal, & Salehi, 2024; Leushuis, Matthijsse, Ophelders, Aarts, & Pechenizkiy, 2025; Koochali, Bauer, Giglio, Kugler, & Görner, 2025; X. Wu et al., 2025). Comprehensive 2024–2025 surveys synthesize these trends and propose taxonomies spanning learning paradigms, architectures, and evaluation practices, underscoring the need for domain-specific benchmarks for fair comparison (Zamanzadeh Darban et al., 2024; Wang et al., 2025; Correia, Goos, Klein, Bäck, & Kononova, 2024; Kim, Park, Lee, & Cho, 2025).

Despite a vast design space for TSAD—for instance, 158 algorithms are cataloged in (Schmidl, Wenig, & Papenbrock, 2022)—the space sector faces a persistent challenge: robust, objective evaluation. Operational missions rarely offer large numbers of labeled anomalies, and comprehensive multi-source datasets remain scarce, complicating fair performance assessment. Moreover, recent analyses have exposed critical shortcomings in public datasets, benchmarks, metrics, and evaluation protocols, highlighting the need for improved, bias-resistant methodologies to assess emerging ML techniques (El Amine Sehili & Zhang, 2023; Wagner et al., 2023; R. Wu & Keogh, 2021; Herrmann, Bieber, Verhagen, Cosson, & Santos, 2024). Addressing this gap, ESA has released an open-source repository of diverse, anonymized satellite telemetry together with a standardized benchmark intended to enable fair comparison of TSAD methods on telemetry data (Kotowski et al., 2024; KP Labs, 2024).

Although interest in data-driven PHM for spacecraft telemetry is accelerating (Fuertes et al., 2016; Ruszczak et al., 2025), most existing ML solutions are tailored to specific components or subsystems and do not readily scale to the full breadth of telemetry. In addition, many approaches require substantial supervised or self-supervised training, which can be impractical for new missions, changing configurations, or drifting telemetry. These constraints motivate the pursuit of generalized anomaly-detection approaches that efficiently analyze large telemetry archives while remaining agnostic to individual satellite subsystems (Kotowski et al., 2024).

In this work, we respond to these challenges by introducing a generalized, efficient, simple, and robust anomaly-detection framework. The proposed algorithm features clearly interpretable hyperparameters that are straightforward to tune, does not require a training phase, and—owing to its computational efficiency—enables rapid processing of large-scale telemetry. Initial evaluations on the ESA Anomaly Detection Benchmark (ESA-ADB) show promising performance, pointing toward safer and more reliable anomaly-detection practices in spacecraft operations (Kotowski et al., 2024).

2. METHODOLOGY

In this section, we define the necessary terms and concepts to describe the TSAD algorithm used in Section 3.

2.1. Distance profile

We define a *time series* T of length n as an ordered sequence of real numbers, $T[i]$, in which $T = (T[1], T[2], ..., T[n])$. A *subsequence*, $T_{i,L}$, is a continuous segment of length L from a time series T starting from position i. $T_{i,L} = (T[i], T[i+1], ..., T[i+L-1])$, where $1 \leq i \leq n - L + 1$. A *query*, Q, is a time series of length m, which is searched within a time series T of length $n >> m$.

Given a time series T and a query Q, the *distance profile* is another sequence D of length $n - m + 1$ such that $D[i] = dist(Q, T_{i,m})$. Here, $dist$ is a generic distance function that defines the distance between two equally length time series. Typical distance functions include Euclidean distance, Pearson's correlation coefficient, cosine similarity, and angular distance. Some more advanced distance functions can compare two unequal-length time series, such as dynamic time warping (DTW) (Sakoe & Chiba, 2003; Z. Zhang et al.,

Algorithm 1: Brute-Force Distance Profile

Input: Time series $T[1 \dots n]$, query $Q[1 \dots m]$ with $m \leq n$
Output: Distance profile $D[1 \dots n-m+1]$

```
// Initialize distance profile array
D[1...n-m+1] ← 0
// Compute Euclidean distances between Q
   and each subsequence of T
for j ← 1 to n − m + 1 do
    s ← 0
    for i ← 1 to m do
        d ← Q[i] − T[i + j − 1]
        s ← s + d^2
    D[j] ← √s
return D
```

2017), longest common subsequence (LCSS) (Vlachos, Kollios, & Gunopulos, 2002; Vlachos, Hadjieleftheriou, Gunopulos, & Keogh, 2003), and move-split-merge (MSM) (Stefan, Athitsos, & Das, 2012). One can consider more variations of the distance profile by allowing the distance function more flexibility, such as by removing the end-point constraint (Silva, Yeh, Batista, & Keogh, 2016). However, the definition of distance profile is not limited to how the distance function operates on the pair of time series.

In this paper, we consider the Euclidean distance as our distance measure without any discontinuity. The Euclidean distance between two time series Q and $T_{i,m}$ of length m is defined in Eq. (1).

$$dist(Q, T_{i,m}) = \sqrt{\sum_{i=1}^{m} (Q[i] - T_{i,m}[i])^2} \tag{1}$$

The Algorithm 1 describes the brute-force way to compute the distance profile, considering the Euclidean distance as the metric. The algorithm scans the time series T once. At each position, the algorithm computes the distance to the query, Q. The algorithm saves all distances in the array D.

The brute-force approach for computing each element of D would involve explicitly evaluating Eq. (1) for every sliding position j. This computation has computational complexity of order $O((n-m+1)m) \sim O(nm)$, which quickly becomes impractical for large n and m. Therefore, in the next section, we describe the $O(n \log n)$ algorithm to compute the distance profile under Euclidean distance. This approach is faster than the baseline approach (Algorithm 1) that has time complexity of $O(nm)$ since $m > \log n$ holds for most real-world applications.

2.2. MASS: Mueen's algorithm for similarity search

To achieve higher computational efficiency, it is beneficial to expand Eq. (1) when applied in Algorithm 1 as follows:

$$\sum_{i=1}^{m} \sqrt{(Q[i] - T[i+j-1])^2} = \tag{2}$$
$$\sqrt{\sum_{i=1}^{m} Q[i]^2 + \sum_{i=1}^{m} T[i+j-1]^2 - 2\sum_{i=1}^{m} Q[i]T[i+j-1]}$$

In this expanded form, we can identify three distinct terms:

- The first term, $\sum_{i=1}^{m} Q[i]^2$, depends only on the query sequence Q and remains constant across all sliding windows. Hence, its computational complexity is $O(m)$.
- The second term, $\sum_{i=1}^{m} T[i+j-1]^2$, is the sum of squares of a sliding subsequence of the time series T. This sum can be efficiently computed for all the sliding subsequences of T in a single pass through the reference time series T, using a prefix sum (or cumulative sum) approach. Computing this term for all windows thus requires only $O(n)$ time.
- The third term, $\sum_{i=1}^{m} Q[i]T[i+j-1]$, is the dot product between the query sequence Q and each subsequence of the reference time series T. Clearly, computing this term for all windows would again have a time complexity $O(nm)$. However, as we will show next, this is exactly the step where we can achieve a significant computational improvement.

The core idea of the MASS algorithm is to use a convolution operation between the query Q, reversed in time, and the reference time series T:

$$\sum_{i=1}^{m} Q[i]T[i+j-1] = (T * Q_{\text{rev}}) \tag{3}$$

where $Q_{\text{rev}} = (Q[m], Q[m-1], ..., Q[1])$, and $*$ denotes the linear convolution operator. Linear convolution can be computed efficiently using the Fast Fourier Transform (FFT) according to the convolution theorem, which states that convolution in the time domain equals multiplication in the frequency domain:

$$y * x_{\text{rev}} = \mathcal{F}^{-1}\left(\mathcal{F}(y) \cdot \mathcal{F}(x_{\text{rev}})\right) \tag{4}$$

Here, $\mathcal{F}(\cdot)$ denotes the FFT operation, and $\mathcal{F}^{-1}(\cdot)$ denotes the inverse FFT. The computational complexity of computing this convolution using the fast FFT algorithm is $O(L \log L)$ (Cormen, Leiserson, Rivest, & Stein, 2001), where L is the length of the sequences after appropriate zero-padding $\left(\text{typically } L = 2^{\lceil \log_2(n+m-1) \rceil} \approx O(n)\right)$, to ensure the element-wise product operation yields only the necessary cross-product terms. Thus, the complexity for computing the

third term using FFT-based convolution reduces dramatically to $O(n \log n)$.

The total computational complexity of this FFT-based approach, known in the literature as the MASS (Mueen's Algorithm for Similarity Search) algorithm (Zhong & Mueen, 2024), is dominated by the FFT-based convolution step, giving an overall complexity of $O(n \log n)$. The Algorithm 2 describes the FFT-based way to compute the distance profile, considering the Euclidean distance as the distance metric.

Algorithm 2: MASS: Distance Profile via FFT

Input: Time series $T[1 \dots n]$, query $Q[1 \dots m]$ with $m \leq n$
Output: Distance profile $D[1 \dots n-m+1]$
// Initialize distance profile array
$D[1 : n-m+1] \leftarrow 0$
// Precompute constant term from Q
$S_{QQ} \leftarrow \sum_{i=1}^{m} Q[i]^2$
// Compute prefix sum of T^2
$\Sigma_{T^2}[1] \leftarrow 0$;
for $i \leftarrow 1$ *to* n **do**
  $\Sigma_{T^2}[i+1] \leftarrow \Sigma_{T^2}[i] + T[i]^2$
// Compute sliding sums of T^2
for $j \leftarrow 1$ *to* $n-m+1$ **do**
  $S_{TT}[j+1] \leftarrow \Sigma_{T^2}[j+m-1] - \Sigma_{T^2}[j]$
// Reverse Q and zero-pad to length n
for $i \leftarrow 1$ **to** n **do**
  if $i \leq m$ **then**
    $Q_{\text{rev}}[i] \leftarrow Q[m+1-i]$
  else
    $Q_{\text{rev}}[i] \leftarrow 0$
// Compute sliding dot products using FFT-based convolution (Eq.(4))
$S_{QT} \leftarrow \texttt{FFTConv}(Q_{\text{rev}}, T)$
// Compute distance profile (Eq.(2))
for $j \leftarrow 1$ *to* $n-m+1$ **do**
  $D[j] \leftarrow \sqrt{S_{QQ} + S_{TT}[j] - 2S_{QT}[j+m-1]}$
return D
Function $\texttt{FFTConv}(x, y)$**:**
  $X \leftarrow \mathcal{F}(x)$
  $Y \leftarrow \mathcal{F}(y)$
  $P \leftarrow X \odot Y$; // element-wise product
  $C \leftarrow \mathcal{F}^{-1}(P)$
  return C

2.3. Sliding Minimum Euclidean Distance (SMED)

Given a time series T of length n, we define W_{R} and W_{Q} as the size of two consecutive windows of fixed length belonging to T, namely the *reference* window and the *query* window, such that, at a given iteration step i, we have:

$$T_{\text{ref}} = T_{i,W_{\text{R}}}$$
$$T_{\text{que}} = T_{W_{\text{R}},W_{\text{R}}+W_{\text{Q}}}$$

The SMED algorithm computes a similarity score by finding the best match between the query and the reference at each iteration, then advances both windows in time. When working with sliding windows, it would, in principle, be possible to slide forward every time a new sample is available in the time series; in that case, the sliding occurs every time step. In our framework, we introduce an additional parameter, the stride L_{W}, which represents the fixed number of samples stored in local memory before sliding forward both the reference window and the current window. In other words, introducing this parameter allows the time series dynamics to be decoupled from the algorithm dynamics. The Sliding Minimum Euclidean Distance (SMED) algorithm that we introduce in this work is described in Algorithm 3.

Algorithm 3: SMED: Sliding Minimum Euclidean Distance

Input: Time series $T[1 \dots n]$
Output: $S_T[1 \dots K]$ the anomaly scores of T
// Initialize S array
$S_T \leftarrow 0$
// Max allowable number of steps
$K = \lfloor \frac{n-(W_{\text{R}}+W_{\text{Q}})}{L_{\text{W}}} \rfloor + 1$
// Compute minimum Euclidean distances
for $i = 1$; $i \leq K$; $i \leftarrow i + L_{\text{w}}$ **do**
  // Update reference and query
  $T_{\text{ref}} \leftarrow T_{i,W_{\text{R}}}$
  $T_{\text{que}} \leftarrow T_{W_{\text{R}},W_{\text{R}}+W_{\text{Q}}}$
  // Compute distance profile
  $D \leftarrow \texttt{MASS}(T_{\text{que}}, T_{\text{ref}})$
  $S_T \leftarrow \texttt{append}\big(\min(D)\big)$; // best match
return S_T

The idea behind this algorithm is to leverage the MASS algorithm to compute at each algorithm step k the minimum Euclidean distance value between the query window and the reference window, aiming at finding the best similarity match between the latest time series values and the past time series values. Intuitively, when this value is high it means that an unseen pattern, given the most recent history, is being scanned. A visual representation of the SMED algorithm is also provided in Figure 1 for clarity.

Suppose that:

$$T = m + \varepsilon$$

where m is a slow-varying trend and ε is a (zero-mean) stationary noise. Consider two subsegments of T, and denote those windows as the reference window T_{ref} and the query window T_{que}:

$$T_{\text{ref}} = T_{i,W_{\text{R}}} = m_{i,W_{\text{R}}} + \varepsilon_{i,W_{\text{R}}}$$
$$T_{\text{que}} = T_{W_{\text{R}},W_{\text{R}}+W_{\text{Q}}} = m_{W_{\text{R}},W_{\text{R}}+W_{\text{Q}}} + \varepsilon_{W_{\text{R}},W_{\text{R}}+W_{\text{Q}}}$$

When computing the distance profile of T_{que} in T_{ref}, the j-th

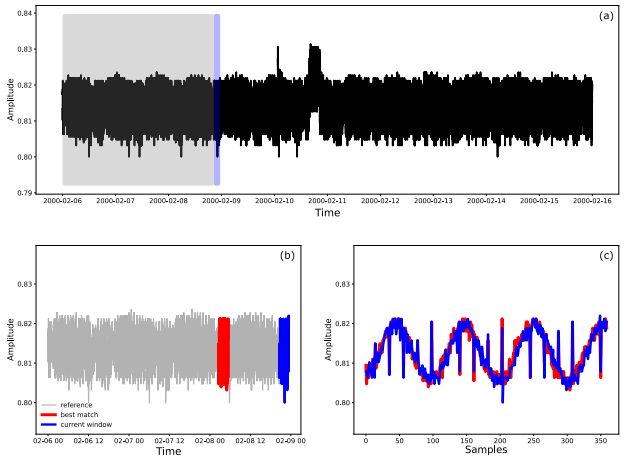

Figure 1. (a) Illustration of the reference window (light-gray area) and the consecutive query window (light-blue area) overlapped on a segment of time series T (black line). (b) Reference series (in light-gray) with outlined best match (in red) and query window (in blue). (c) Zoom on the overlapped match and the query window.

Euclidean distance value is:

$$D[j] = \sum_{i=1}^{W_\mathrm{Q}} \left(T_\mathrm{que}[i] - T_\mathrm{ref}[i+j-1]\right)^2$$
$$= \sum_{i=1}^{W_\mathrm{Q}} \left(m_{W_\mathrm{R},W_\mathrm{R}+W_\mathrm{Q}} + \varepsilon_{W_\mathrm{R},W_\mathrm{R}+W_\mathrm{Q}} - m^j_{i,W_\mathrm{R}} - \varepsilon^j_{i,W_\mathrm{R}}\right)^2$$

Let

$$\Delta m^j_i = \left(m_{W_\mathrm{R},W_\mathrm{R}+W_\mathrm{Q}} - m^j_{i,W_\mathrm{R}}\right)$$
$$\Delta \varepsilon^j_i = \left(\varepsilon_{W_\mathrm{R},W_\mathrm{R}+W_\mathrm{Q}} - \varepsilon^j_{i,W_\mathrm{R}}\right)$$

Then

$$D[j] = \sum_{i=1}^{W_\mathrm{Q}} \left(\Delta m^j_i + \Delta \varepsilon^j_i\right)^2$$

Expanding the square:

$$D[j] = \sum_{i=1}^{W_\mathrm{Q}} \left(\Delta^2 m^j_i + 2\Delta m^j_i \Delta \varepsilon^j_i + \Delta^2 \varepsilon^j_i\right)$$
$$= W_\mathrm{Q}(\Delta^2 m^j + 2\Delta m^j \Delta \varepsilon^j + \Delta^2 \varepsilon^j)$$

and taking expectation on D with respect to noise, yields:

$$\mathbb{E}\big[D\big] = W_\mathrm{Q}(\Delta^2 m + 2\sigma^2)$$

being $\mathbb{E}[\Delta\varepsilon] = 0$ and $\mathbb{E}[\Delta^2\varepsilon] = 2\sigma^2$.

Therefore, the distance profile is roughly proportional to:

$$W_\mathrm{Q}(\Delta^2 m) + \text{noise}$$

With the SMED algorithm, we minimize the quantity above over j within the reference window, and hence pick the reference window slice whose trend is the $closest$ to the query window's trend. If the drift across the span of the reference window is small relative to noise variance, the added trend-difference penalty is minor. Therefore, the anomaly score, S_T, computed on T using Algorithm 3 behave almost as if the process were stationary, even though T is not. Only when abrupt changes are observed in T, S_T will not behave so. This allows the SMED algorithm to adapt to slow time-varying trend time series, and to highlight the occurrence of sudden changes in the time series, which are likely due to the presence of anomalies.

2.4. Thresholding

In every anomaly detection task, establishing a proper threshold is a critical step. In this subsection, we introduce the logic behind the threshold definition in our TSAD algorithm. In statistics, given a generic distribution, the so-called measure of statistical dispersion, also known as the measure of spread, of the distribution is typically performed using the InterQuartile Range (IQR), which is defined as the difference between the 75th and 25th percentiles of the data. A more general definition in this sense, is the InterPercentile Range (IPR). Given a general percentile value, p, the latter is defined as:

$$\mathrm{IPR}_p = Q_{1-p} - Q_p, \quad 0 \le p \le 0.5 \tag{5}$$

Note that, when $p = 0.25$ $\mathrm{IPR}_p = \mathrm{IQR}$. The above quantities are usually employed in statistics to define the so-called *box-plot*, which is a method for demonstrating graphically the locality, spread, and skewness of groups of numerical data through their percentiles. In this representation (Figure 2), there can be lines (which are called *whiskers*) extending from the box indicating variability outside the upper and lower percentiles. A popular choice for the whiskers' boundaries is based on the value of 1.5 IPR_p. From above the upper percentile (Q_{1-p}), a distance of 1.5 times the IPR_p is measured out, and a whisker is drawn up to the largest observed data point from the dataset that falls within this distance. Similarly, a distance of 1.5 times the IPR_p is measured out below the lower percentile (Q_p), and a whisker is drawn down to the lowest observed data point from the dataset that falls within this distance. Outliers that differ significantly from the rest of the data may be plotted as individual points beyond the whiskers on the box-plot. Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution.

Given that, by definition of Euclidean distance, the computed anomaly score values result in a distribution of strictly positive numbers, i.e., $S_T \in [0, +\infty)$. Here we set the threshold using scores computed on the *nominal* behavior of the time

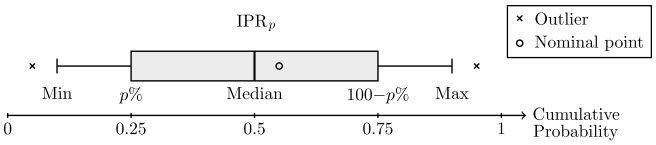

Figure 2. Visual representation of the interpercentile range.

series T. Let S_T denote the anomaly-score distribution obtained from a baseline nominal segment of T, we define the initial threshold τ as the upper whisker of the box-plot for S_T:

$$\tau = Q_{1-p}(S_T) + 1.5 \cdot \text{IPR}_p(S_T) \tag{6}$$

2.5. Anomaly detection

In this subsection, we describe the TSAD algorithm established in this work for the task of effectively detecting anomalous events in satellite telemetry data. First, we define a set of time series $\mathbf{T} = \{T^1, T^2, ..., T^N\}$, an array of N time series of length n. The TSAD algorithms consists of two main phases: (i) the *warm-up* phase where we take the first n_{wu} samples of $\mathbf{T}$ and we apply Algorithm 3 to compute the anomaly score distribution for each time series in the set, next we compute the N initial threshold values for each time series using Eq. (6) ; (ii) the *deployment* phase, where Algorithm 3 is applied online on the remaining samples of $\mathbf{T}$ to determine the anomaly score values at each step k. In every iteration, the latter values are compared against their corresponding thresholds to assess whether an anomalous pattern is encountered or not. The corresponding predictions are collected. Optionally, it is possible to update the threshold value employing a calendar strategy, where we update the threshold values based on the newly computed anomaly scores. These steps are summarized in Algorithm 4. For the sake of clarity, a visual representation showing how SMED and the thresholding mechanism interact is illustrated in Figure 3.

Algorithm 4: TSAD: Time series anomaly detection

Input: Time series set $\mathbf{T} = \{T^1, T^2, \ldots, T^N\}$
Output: $P_{\mathbf{T}}$ the predictions computed on $\mathbf{T}$

```
// Warm-up phase
S_T^old ← WarmUp(T, n_wu)
S_T^new ← ∅
τ ← ComputeThres(S_T^old, S_T^new, p)
// Deployment phase
for k ← 1 to N do
    T^k ← T^k[n_wu + 1 : end]
    // Max allowable number of steps
    n_test = n − n_wu
    K = ⌊(n_test − (W_R + W_Q)) / L_W⌋ + 1
    for i = 1; i ≤ K; i ← i + L_w do
        // Update reference and query
        T^k_ref ← T^k_{i,W_R}
        T^k_que ← T^k_{W_R, W_R+W_Q}
        // Compute distance profile
        D^k ← MASS(T^k_que, T^k_ref)
        s_{T^k} ← min(D^k)
        S_{T^k} ← append(s_{T^k})
        if s_{T^k} > τ^k then
            P_{T^k} ← append(1)
        else
            P_{T_k} ← append(0)
        if istrue(update) then
            if mod(i, freq_update) == 0 then
                S^new_{T^k} ← S_{T^k}
                τ^k ← computeThres(S^old_{T^k}, S^new_{T^k}, p)
    S_T ← append(S_{T^k})
    P_T ← append(P_{T^k})
Function WarmUp(T, n_wu):
    for k ← 1 to N do
        T^k ← T^k[1 : n_wu]
        S_{T^k} ← SMED(T^k)
        S_T ← append(S_{T^k})
    return S_T
Function ComputeThres(S^old_T, S^new_T, p):
    for k ← 1 to N do
        S_{T^k} ← concatenate(S^old_{T^k}, S^new_{T^k})
        τ^k ← Q_{1−p}(S_{T^k}) + 1.5 · IPR_p(S_{T^k})
        τ ← append(τ^k)
    return τ
return P
```

3. RESULTS

In this section, we report the results obtained by applying the TSAD algorithm presented above on a time series set that is extracted from the ESA-ADB datasets. The ESA-ADB datasets includes 76 channels from Mission 1 and 100 channels from Mission 2; however, only 58 and 47 channels, respectively, can be monitored for anomalies (*target* channels), while the rest are meant to support the detection process (*non-target* channels). Channels are grouped into 6 subsystems – 4 in Mission 1 and 5 in Mission 2, with 3 matching subsystems between missions. The anomaly density, in terms of annotated data points, is between 1.80% (Mission 1) and 0.57% (Mission 2), which addresses the flaw of unrealistic anomaly density reported for many popular TSAD datasets, as also stated by ESA itself. For further details on this datasets, the interested reader should refer to (Kotowski et al., 2024). To evaluate the performance of our algorithm on the ESA-ADB datasets, we have resorted to the benchmark established by ESA, whose code is available at the link indicated in Ref. (KP Labs, 2024). A summary of the introduced metrics and their hierarchical evaluation is shown in Table 1. Again, for the definition of these metrics the interested reader should refer to (Kotowski et al., 2024). Anomaly detection in tens or hundreds of channels simultaneously may be a very challenging task and it takes a lot of computing power to process such data, so for initial experiments, simpler models, and potential on-board applications, there are also lightweight subsets of channels proposed in ESA-ADB. These are channels 41-

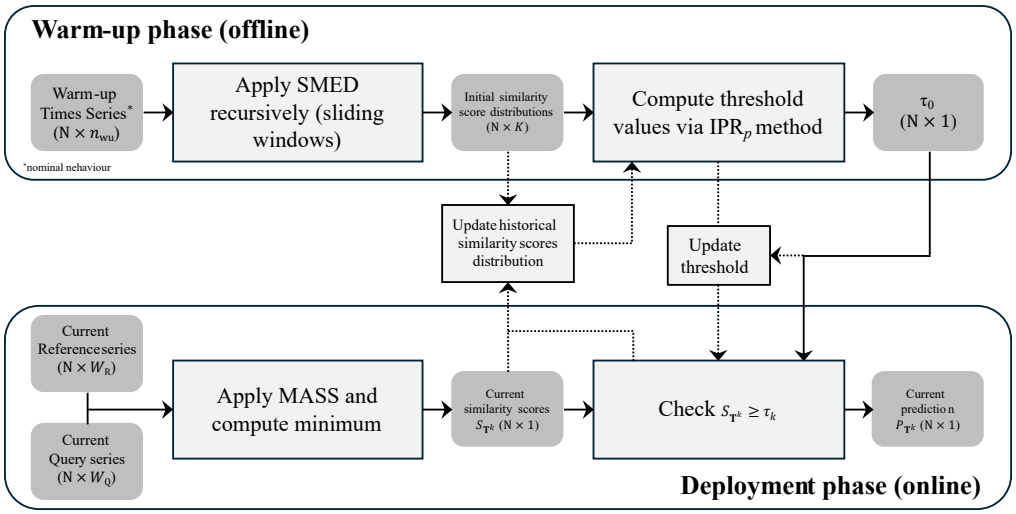

Figure 3. Conceptual diagram illustrating the overall TSAD pipeline. The diagram outlines the interaction between the similarity search and the thresholding mechanism at each prediction step. Dotted lines indicate optional paths which are followed when dynamic thresholding is enabled.

Table 1. 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 — minimise the number of false detections 1b. Anomaly existence — maximise the number of correctly detected anomalies	Corrected event-wise $F_{0.5}$-score
	2a. Subsystems identification — find a list of affected subsystems 2b. Channels identification — find a list of affected channels	Subsystem-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

Table 2. Benchmarking results for detection of **all events** (excluding communication gaps) in lightweight subsets of channels for mission 1 and 2 in ESA-ADB (After ratio of ADTQC which is just a helper value).

Metric		Mission 1		Mission 2	
		SMED	Best-ESA	SMED	Best-ESA
Event-wise	Precision	0.999	0.999	0.979	0.978
	Recall	0.446	0.424	0.940	0.540
	$F_{0.5}$-score	0.801	0.786	0.971	0.842
Channel-aware	Precision	0.446	0.424	0.940	0.465
	Recall	0.438	0.275	0.277	0.384
	$F_{0.5}$-score	0.444	0.362	0.610	0.442
Alarming Precision		0.659	0.875	0.899	0.862
ADTQC	After ratio	0.103	0.143	0.015	0.351
	Score	0.160	0.197	0.075	0.757
Affiliation-based	Precision	0.716	0.711	0.933	0.759
	Recall	0.435	0.423	0.928	0.530
	$F_{0.5}$-score	0.634	0.626	0.932	0.699

46 from subsystem 5 for Mission 1 and channels 18-28 from subsystem 1 for Mission 2.

Here, for the sake of brevity, we focus our performance comparison against the best algorithm tested by ESA, i.e., the so-called *Telemanon-ESA*. For a more comprehensive comparison against other state-of-the-art algorithms tested by ESA on the same datasets, the interested reader can refer to (Kotowski et al., 2024). Additionally, in this preliminary study, we focus on the two lightweight subsets to demonstrate the effectiveness of our algorithm. Notably, the two lightweight subsets were processed in approximately 16 and 13 seconds, respectively, when running the algorithm on a machine equipped with 128 GB of RAM and an Intel Core i9-14900K CPU. ESA, on the other hand, reported a processing time of 1.9h and 1.3h, respectively, for their Telemanon-ESA algorithm when the latter is run on the lightweight subsets on a machine equipped with 32 GB RAM and an Intel i7-10870H CPU (Kotowski et al., 2024). Despite our CPU is roughly 65% more powerful (see `https://cpu.userbenchmark.com/`), our processing times are, respectively, 433% and 359% faster.

The SMED algorithm has four main hyperparameters that must be properly tuned in order to run it. These are the reference length size, W_R, the query length size, W_Q, the stride, L_W, and the percentile value, p, for setting the threshold. Since the threshold is defined considering a portion of time series that represents its nominal behavior, it is reasonable to assume a very small value of p (not zero, to avoid in-

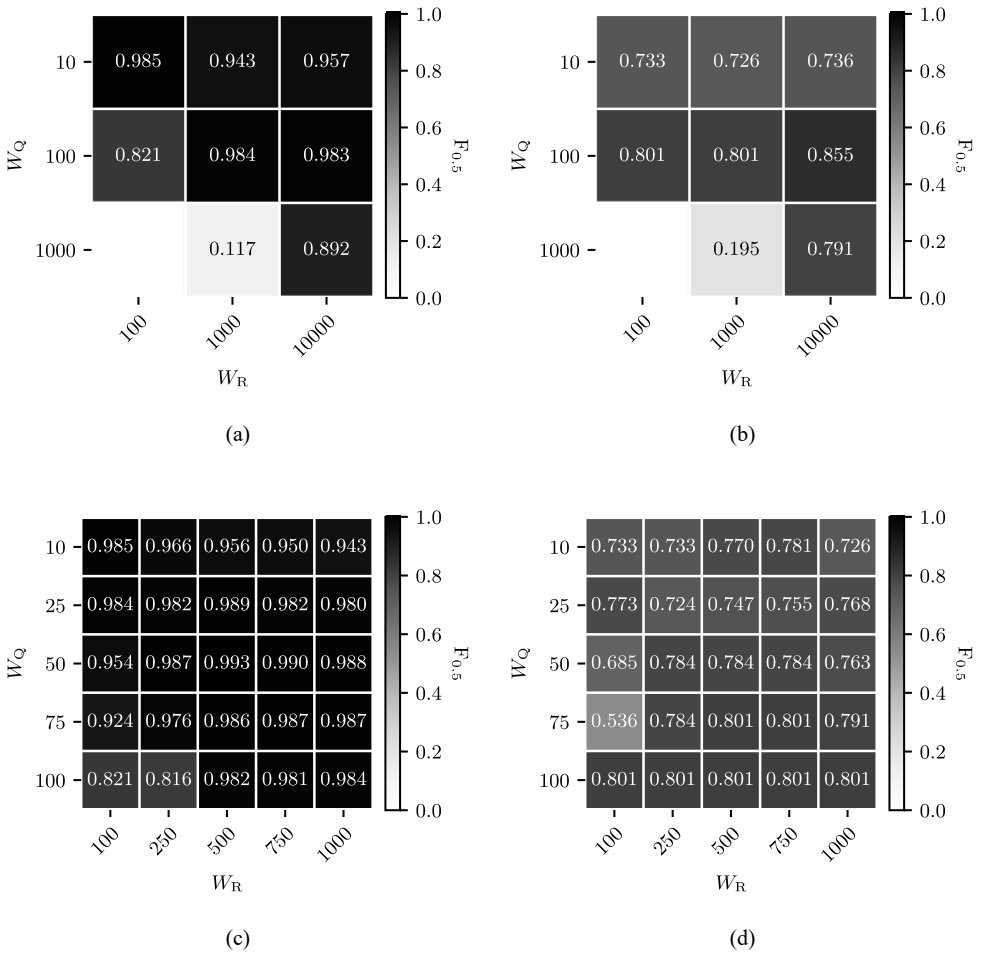

Figure 4. (a) Coarse order-of-magnitude sensitivity analysis of SMED to query and reference window size for Mission 1. (b) Coarse order-of-magnitude sensitivity analysis of SMED to query and reference window size for Mission 2. (c) Refined sensitivity analysis of SMED to query and reference window size for Mission 1. (d) Refined sensitivity analysis of SMED to query and reference window size for Mission 2.

Table 3. Benchmarking results for detection of **anomalies alone** in lightweight subsets of channels for mission 1 and 2 in ESA-ADB (After ratio of ADTQC which is just a helper value).

Metric		Mission 1		Mission 2	
		SMED	Best-ESA	SMED	Best-ESA
Event-wise	Precision	0.806	0.999	0.000	0.000
	Recall	0.862	0.862	0.000	0.000
	$F_{0.5}$-score	0.816	0.968	0.000	0.000
Channel-aware	Precision	0.862	0.529	0.000	0.000
	Recall	0.862	0.862	0.000	0.000
	$F_{0.5}$-score	0.862	0.722	0.000	0.000
Alarming Precision		0.676	0.862	0.000	0.000
ADTQC	After ratio	0.000	0.040	-	-
	Score	0.130	0.159	-	-
Affiliation-based	Precision	0.914	0.927	0.554	0.500
	Recall	0.853	0.859	0.771	0.000
	$F_{0.5}$-score	0.901	0.912	0.587	0.000

cluding possible physiological outliers in $S_{T_{\text{nom}}}$ due to non-anomalous patterns to make the threshold setting more stable).Therefore, we fix $p = 0.001$. The stride, L_{W}, determines the time granularity of the algorithm, i.e., of the prediction step. In principle, $L_{\text{W}} = 1$ would allow the algorithm to closely follow the dynamic of the time series which is being scanned, however, here, for the sake of computational time (given the large amount of data points and time series considered), we fix $L_{\text{W}} = 5$, which is a good trade-off value, especially when considering a small query size (note, in fact, that it should always be satisfied by the condition $L_{\text{W}} \geq W_{\text{Q}}$ to avoid time gaps). W_{R}, W_{Q} are not trivial to tune and significantly affect the performance of the algorithm. Hence, we perform a sensitivity analysis on these two parameters. We use the event-wise correct F_β score (with $\beta = 0.5$) as the reference global performance metric, since according to the ESA this metric summarizes the most important feature that a TSAD should have.

Figure 4(a) and Figure 4(b) show the results of a preliminary coarse sensitivity analysis investigating the effect on performance of different relative orders of magnitude between the query and the reference sizes, sweeping W_{Q} from $10^1 - 10^3$ and W_{R} from $10^2 - 10^4$. The results show that, as a rule of thumb, a difference of one order of magnitude leads to the best performance. Conversely, significant performance degradation occurs when the query size is too close to or too small compared to the reference size. Another very important aspect, is the computational time needed to run the algorithm, which rely on the MASS algorithm to compute the minimum value of the distance profile between the reference and the query at each prediction step, whose time-complexity is $O(n \log n)$. Therefore, smaller reference window sizes are recommended when yielding to similar performance results. We therefore conduct a more refined sensitivity analysis, shown in Figure 4(c) and Figure 4(d), considering the range $10 - 10^2$ for W_{Q} and the range $10^2 - 10^3$ for W_{R}. Surprisingly, considering these ranges (excluding the trivial case $W_{\text{Q}} = W_{\text{R}}$) the algorithm global performance are not strongly affected, showcasing a rather good robustness when the latter parameters are properly tuned.

The detection performance results are summarized in Table 2 (for all events), and Table 3 (for anomalies alone), and show competitive performance compared to the baseline approaches when considering these two datasets. These results were obtained using the following hyperparameter settings:

$$\begin{aligned} W_{\text{R}} &= 750 \\ W_{\text{Q}} &= 100 \\ L_{\text{W}} &= 5 \\ p &= 0.001 \end{aligned}$$

which represent a preliminary yet effective configuration for the algorithm, obtained from the sensitivity analysis.

4. CONCLUSION

This work introduced a simple, plug-and-play framework for unsupervised time series anomaly detection in satellite telemetry. The core SMED method leverages a MASS-based distance profile to compare current behavior against recent patterns, while an adaptive IPR-based thresholding strategy starting from a healthy baseline limits the need for hyperparameter tuning and labeled data while ensuring low false-positive rates. Preliminary experiments on the ESA-ADB lightweight subsets indicate that the proposed method obtains event-wise performance competitive and in some cases superior to the performance obtained by the best algorithm tested by ESA in their benchmark. These results were obtained on the resampled data, which are those processed by the majority of the algorithms tested by ESA, including the best one. These results suggest that simple, efficient similarity search coupled with adaptive thresholding is a strong baseline for operational PHM pipelines in space missions. Future work will: (i) broaden validation to the full ESA-ADB benchmark set; and (ii) test the methodology on raw, non-resampled signals. This will allow for a full assessment of the methodology's superior robustness compared to state-of-the-art TSAD algorithms for telemetry data.

ACKNOWLEDGMENT

This work is funded by the Project EXTESA: EXTESA - EXplainable deep learning" per la definizione di un sisTEma diagnostico e prognostico (PHM) automatico in futuri Satelliti per missioni lunAri. The project is funded by ASI - Agenzia Spaziale Italiana, as ADDENDUM n. 2021-41-HH.1-2025 and under the "ACCORDO ATTUATIVO n. 2022-4-Q.0".

REFERENCES

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms. 2001.* MIT press: Cambridge, US.

Correia, L., Goos, J.-C., Klein, P., Bäck, T., & Kononova, A. V. (2024). Online model-based anomaly detection in multivariate time series: Taxonomy, survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence*, *133*, 109323. Retrieved from https://doi.org/10.1016/j.engappai.2024.109323 doi: 10.1016/j.engappai.2024.109323

El Amine Sehili, M., & Zhang, Z. (2023). Multivariate time series anomaly detection: Fancy algorithms and flawed evaluation methodology. In *Technology conference on performance evaluation and benchmarking* (pp. 1–17).

Evans, D., Martinez, J., Korte-Stapff, M., Brighenti, A., Brighenti, C., & Biancat, J. (2017). Data mining to drastically improve spacecraft telemetry checking. In *Space operations: Contributions from the global community* (pp. 87–113). Springer.

Fuertes, S., Picart, G., Tourneret, J.-Y., Chaari, L., Ferrari, A., & Richard, C. (2016). Improving spacecraft health monitoring with automatic anomaly detection techniques. In *14th international conference on space operations* (p. 2430).

Fuertes, S., Pilastre, B., & D'Escrivan, S. (2018). Performance assessment of nostradamus & other machine learning-based telemetry monitoring systems on a spacecraft anomalies database. In *2018 spaceops conference* (p. 2559).

Herrmann, L., Bieber, M., Verhagen, W. J., Cosson, F., & Santos, B. F. (2024). Unmasking overestimation: a re-evaluation of deep anomaly detection in spacecraft telemetry. *CEAS Space Journal*, *16*(2), 225–237.

Hu, X., Xu, L., Lin, X., & Pecht, M. (2020). Battery lifetime prognostics. *Joule*, *4*(2), 310–346.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 387–395).

Iqbal, A., & Amin, R. (2024). Time series forecasting and anomaly detection using deep learning. *Computers & Chemical Engineering*, *182*, 108560. Retrieved from https://www.sciencedirect.com/science/article/pii/S0098135423004301 doi: 10.1016/j.compchemeng.2023.108560

Kang, H., & Kang, P. (2024). Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism. *Knowledge-Based Systems*, *290*, 111507. Retrieved from https://doi.org/10.1016/j.knosys.2024.111507 doi: 10.1016/j.knosys.2024.111507

Kim, J., Kang, H., & Kang, P. (2023). Time-series anomaly detection with stacked transformer representations and 1d convolutional network. *Engineering Applications of Artificial Intelligence*, *120*, 105964. Retrieved from https://doi.org/10.1016/j.engappai.2023.105964 doi: 10.1016/j.engappai.2023.105964

Kim, J., Park, S., Lee, M., & Cho, S. (2025). A comprehensive survey of deep learning for time series: Architectures, challenges, and emerging directions. *Artificial Intelligence Review*. Retrieved from https://doi.org/10.1007/s10462-025-11223-9 (Online first 2025) doi: 10.1007/s10462-025-11223-9

Koochali, A., Bauer, D., Giglio, A., Kugler, S., & Görner, L. (2025). Vaeneu: A new avenue for VAE application on probabilistic forecasting. *Applied Intelligence*. Retrieved from https://doi.org/10.1007/s10489-024-06203-5 (Online first 2025) doi: 10.1007/s10489-024-06203-5

Kotowski, K., Haskamp, C., Andrzejewski, J., Ruszczak, B., Nalepa, J., Lakey, D., ... Martinez-Heras, J. (2024). European space agency benchmark for anomaly detection in satellite telemetry. *arXiv preprint arXiv:2406.17826*.

KP Labs. (2024). *Esa anomaly detection benchmark.* https://github.com/kplabs-pl/ESA-ADB. (GitHub repository; accessed 2025-07-16)

Lakey, D., & Schlippe, T. (2024). A comparison of deep learning architectures for spacecraft anomaly detection. In *2024 ieee aerospace conference* (pp. 1–11).

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical systems and signal processing*, *42*(1-2), 314–334.

Leushuis, R. M., Matthijsse, P., Ophelders, T., Aarts, E., & Pechenizkiy, M. (2025). Probabilistic forecasting with var-vae: Advancing time series forecasting with vector autoregression in the latent space. *Information Sciences*, *692*, 122184. Retrieved from https://doi.org/10.1016/j.ins.2025.122184 doi: 10.1016/j.ins.2025.122184

Lutz, R. R., & Mikulski, I. C. (2004). Empirical analysis of safety-critical anomalies during operations. *IEEE Transactions on Software Engineering*, *30*(3), 172–180.

Martinez, J. (2012). Drmust-a data mining approach for anomaly investigation. In *Spaceops 2012* (p. 1275109).

Martinez, J., & Donati, A. (2018). Novelty detection with deep learning. In *2018 spaceops conference* (p. 2560).

Neloy, A. A., & Turgeon, M. (2024). A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs. *Machine Learning*

with Applications, *17*, 100572. Retrieved from https://www.sciencedirect.com/science/article/pii/S2666827024000148 doi: 10.1016/j.mlwa.2024.100572

OMeara, C., Schlag, L., Faltenbacher, L., & Wickler, M. (2016). Athmos: automated telemetry health monitoring system at gsoc using outlier detection and supervised machine learning. In *14th international conference on space operations* (p. 2347).

Pilastre, B., Boussouf, L., d'Escrivan, S., & Tourneret, J.-Y. (2020). Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning. *Signal Processing*, *168*, 107320.

Ruszczak, B., Kotowski, K., Andrzejewski, J., Musiał, A., Evans, D., Zelenevskiy, V., ... Nalepa, J. (2023). Machine learning detects anomalies in ops-sat telemetry. In *International conference on computational science* (pp. 295–306).

Ruszczak, B., Kotowski, K., Evans, D., & Nalepa, J. (2025). The ops-sat benchmark for detecting anomalies in satellite telemetry. *Scientific Data*, *12*(1), 710.

Sakoe, H., & Chiba, S. (2003). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, *26*(1), 43–49.

Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, *15*(9), 1779–1797.

Shimillas, C., Malialis, K., Fokianos, K., & Polycarpou, M. M. (2025). *Transformer-based multivariate time series anomaly localization.* arXiv preprint arXiv:2501.08628. Retrieved from https://arxiv.org/abs/2501.08628 doi: 10.48550/arXiv.2501.08628

Silva, D. F., Yeh, C.-C. M., Batista, G. E., & Keogh, E. J. (2016). Simple: Assessing music similarity using subsequences joins. In *Ismir* (pp. 23–29).

Stefan, A., Athitsos, V., & Das, G. (2012). The move-split-merge metric for time series. *IEEE transactions on Knowledge and Data Engineering*, *25*(6), 1425–1438.

Tariq, S., Lee, S., Shin, Y., Lee, M. S., Jung, O., Chung, D., & Woo, S. S. (2019). 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* (pp. 2123–2133).

Tsutsumi, S., Hirabayashi, M., Sato, D., Kawatsu, K., Sato, M., Kimura, T., ... Abe, M. (2021). Data-driven fault detection in a reusable rocket engine using bivariate time-series analysis. *Acta Astronautica*, *179*, 685–694.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining* (pp. 216–225).

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Proceedings 18th international conference on data engineering* (pp. 673–684).

Wagner, D., Michels, T., Schulz, F. C., Nair, A., Rudolph, M., & Kloft, M. (2023). Timesead: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research*.

Wang, F., Jiang, Y., Zhang, R., Wei, A., Xie, J., & Pang, X. (2025). A survey of deep anomaly detection in multivariate time series: Taxonomy, applications, and directions. *Sensors*, *25*(1), 190. Retrieved from https://www.mdpi.com/1424-8220/25/1/190 doi: 10.3390/s25010190

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2023). Transformers in time series: A survey. In *Proceedings of the 32nd international joint conference on artificial intelligence (ijcai-23)* (pp. 5605–5612). IJCAI. Retrieved from https://www.ijcai.org/proceedings/2023/0759.pd doi: 10.24963/ijcai.2023/759

Wu, J. (2005). Liquid-propellant rocket engines health-monitoring—a survey. *Acta Astronautica*, *56*(3), 347–356.

Wu, R., & Keogh, E. J. (2021). 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.

Wu, X., Qiu, X., Gao, H., Hu, J., Yang, B., & Guo, C. (2025). *k^2vae: A koopman-kalman enhanced variational autoencoder for probabilistic time series forecasting.* arXiv preprint arXiv:2505.23017. Retrieved from https://arxiv.org/abs/2505.23017 (Also presented at ICML 2025)

Yairi, T., Takeishi, N., Oda, T., Nakajima, Y., Nishimura, N., & Takata, N. (2017). 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.

Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2024). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, *57*(6), 1–42. Retrieved from https://dl.acm.org/doi/10.1145/3691338 doi: 10.1145/3691338

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., ... Chawla, N. V. (2019). A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 1409–1416).

Zhang, Z., Tavenard, R., Bailly, A., Tang, X., Tang, P., &

Corpetti, T. (2017). Dynamic time warping under limited warping path length. *Information Sciences*, *393*, 91–107.

Zhong, S., & Mueen, A. (2024). Mass: distance profile of a query over a time series. *Data Mining and Knowledge Discovery*, *38*(3), 1466–1492.

BIOGRAPHIES

Lorenzo. BRANCATO was born in Italy on December 12, 1998. He earned his bachelor's degree in mechanical engineering from the Politecnico di Milano in 2020. He then pursued a Master's degree in Mechatronic Engineering at the same institution, completing his studies in 2022. In 2023, he began his Ph.D. in Mechanical Engineering at Politecnico di Milano. His research focuses on developing advanced diagnostic and prognostic approaches for complex dynamic systems subject to degradation and faults. This involves high-fidelity multiphysics modeling and simulation, advanced model-based filtering methods, and advanced data-driven machine learning methods.

Alessandro. LUCCHETTI was born in Sestri Levante, Italy, on 31 August 2000. He received a B.Sc. in Vehicle Engineering from the University of Modena and Reggio Emilia in 2022 and an M.Sc. in Mechanical Engineering from Politecnico di Milano in 2025. He is now a Ph.D. candidate in Mechanical Engineering at Politecnico di Milano, where his research explores surrogate modeling, data-driven, and machine-learning methods for the health monitoring and diagnosis of mechanical and aerospace systems.

Marco. GIGLIO is Full Professor at the Department of Mechanical Engineering, Politecnico di Milano. His main research fields are: (i) Structural integrity evaluation of complex platforms through Structural Health Monitoring methodologies; (ii) Vulnerability assessment of ballistic impact damage on components and structures, in mechanical and aeronautical fields; (iii) Calibration of constitutive laws for metallic materials; (iv) Expected fatigue life and crack propagation behavior on aircraft structures and components; (v) Fatigue design with defects. He has been the coordinator of several European projects: HECTOR, Helicopter Fuselage Crack Monitoring and prognosis through on-board sensor, 2009-2011, ASTYANAX (Aircraft fuselage crack Monitoring System and Prognosis through eXpert on-board sensor networks), 2012-2015, and SAMAS (SHM application to Remotely Piloted Aircraft Systems), 2018-2020. He has been the project leader of the Italian Ministry of Defence project in the National Plan of Military Research, SUMO (Development of a predictive model for the ballistic impact), 2011-2012 and, SUMO 2 (Development of an analytical, numerical and experimental methodology for design of ballistic multilayer protections), 2017-2019. He has published more than 210 papers, h-index 27 (source Scopus) in referred international journals and congresses.

Francesco. CADINI (MSc in Nuclear Engineering, Politecnico di Milano, 2000; MSc in Aerospace Engineering, UCLA, 2003; PhD in Nuclear Engineering, Politecnico di Milano, 2006) is Associate Professor at the Department of Mechanical Engineering, Politecnico di Milano. He has more than 20 years of experience in the assessment of the safety and integrity of complex engineering systems, entailing (i) artificial intelligence (machine learning)-based approaches for classification and regression, (ii) development and application of advanced Monte Carlo algorithms for reliability analysis (failure probability estimation), (iii) diagnosis and prognosis (HUMS) of dynamic, complex systems subject to degradation, (iv) uncertainty and sensitivity analyses, (v) structural reliability analyses.