ANOMALY DETECTION FOR SPARSE AND IRREGULAR MULTIVARIATE TIME SERIES WITH LATENT SDES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series anomaly detection (MTSAD) is critical for a wide range of application areas, such as industrial monitoring, cybersecurity, or healthcare. Real-world data is often sparse, irregularly sampled or partially observed, yet existing methods assume uniformly sampled time series. We propose Latent SDE Anomaly Detection (LSD), a generative approach that models the observed time series through a continuous-time stochastic dynamical system in a low-dimensional latent space, either Euclidean or hyperspherical. This formulation directly handles missing observations and irregular sampling, without imputation or resampling. On six established benchmarks, LSD achieves the best overall rank among classical and state-of-the-art baselines, and on an industrial benchmark with repetitive dynamics, it reaches roughly twice the AUPRC of the strongest baseline. Under severe sparsity, with as little as 1 % of the data retained on an industrial benchmark, LSD remains the best-performing method, whereas the ranking of the baselines changes substantially with the amount of available data. These results highlight latent SDEs as a natural inductive bias for anomaly detection in multivariate time series, especially in the presence of real-world irregularities.

1 INTRODUCTION

Multivariate time series are ubiquitous in a variety of real-world applications, from industrial monitoring to cybersecurity, finance, or healthcare. Monitoring for anomalous patterns is critical for predictive maintenance in industry, fraud detection in finance, intrusion detection in cybersecurity, or early warning systems in healthcare and has been actively researched since the 1960s.

With the increasing availability of data, the task of Multivariate Time Series Anomaly Detection (MTSAD) has gained attention in machine learning research (Blázquez-García et al., 2021; Li & Jung, 2023; Zamanzadeh Darban et al., 2024). Concerning the learning methodology, all paradigms from unsupervised to supervised, including semi- and self-supervised methods can be found in the literature. However, for many real-world applications, we lack labeled data, putting a focus on unsupervised methods. In this regime, one trains a model exclusively on benign data and anomalous patterns are then identified as out-of-distribution samples under the learned model by thresholding an anomaly score. Figure 1 illustrates this paradigm on three representative features from the Quanser Aero 2 Pick-and-Place Dataset (QAPPD) benchmark (Nosrati et al., 2027), where the thresholded score yields a binary prediction classifying each sample as either benign or anomalous.

Furthermore, in many real-world applications, time series data is often noisy (Cook et al., 2020). More critical, however, it is often also highly irregular: different variables are sampled unevenly, asynchronously or at different frequencies; hence, data might be sparse or partially missing. The prevalent industrial protocol OPC UA implements data sparsity by design; data is only published when values deviate sufficiently from the last recorded value (cf. OPC Foundation, 2023). The assumption of equidistant sampling is therefore often unrealistic in practice (cf. Silva et al., 2012; Weerakody et al., 2021), yet state-of-the-art methods for MTSAD typically rely on this assumption.

To still apply these methods, the data needs to be preprocessed, e.g., by imputation and resampling. However, this may introduce bias, alter temporal structure, and affect downstream detection. In line with Belay et al. (2023), who argue that such preprocessing should be avoided altogether, we contend that MTSAD methods should natively accommodate these data characteristics without relying on imputation or resampling (we refer to this requirement as **R1**).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Beyond sampling irregularity, time series are often high-dimensional, with multiple measured variables reflecting different aspects of a shared underlying dynamical process, such as a physical process in industrial control or a physiological process in healthcare. In machine and factory automation, for instance, repeated pick-and-place tasks or injection molding operations lead to roughly cyclic trajectories in the observed time series (Schindler et al., 2025), and abnormal behavior may manifest not in a single variable but in the interplay between several. This motivates models that explicitly represent the shared temporal structure across variables (we refer to this requirement as **R2**). Even during normal operation, however, the underlying process need not evolve identically across repetitions. Small, unmeasured fluctuations in load or friction, or other environmental conditions such as the ambient temperature can alter the evolution of the system state, e.g., by changing the speed or timing of a robotic movement, and thereby affect subsequent measurements. A model should therefore accommodate variability in the underlying dynamics (we refer to this requirement as **R3**), separately from measurement noise that affects how the system state is recorded.

Benchmarking progress in MTSAD is further complicated by the limited quality and diversity of available datasets. Wu & Keogh (2023) show that many widely used benchmarks are trivially solvable, in the sense that near-perfect detection can be achieved with a single line of code, and that some contain mislabeled anomalies. Moreover, industrial data is frequently subject to privacy constraints and proprietary restrictions that prevent its public release (Souza et al., 2020). Available benchmarks therefore originate primarily from three sources: internet companies addressing their own operational monitoring problems (Su et al., 2019; Abdulaal & Lancewicki, 2021), public organizations in narrow, domain-specific applications (Hundman et al., 2018), and research institutes collecting data in controlled testbeds (Goh et al., 2017; Ahmed et al., 2017; Nosrati et al., 2027). In these settings, data is typically logged at fixed rates with centralized acquisition, which partly explains why irregular sampling and natural sparsity are largely absent from existing benchmarks.

Against this background, we make the following **contributions**: **(1) A novel MTSAD method**, Latent SDE Anomaly Detection (LSD), that takes learning the latent stochastic dynamics as a first principle. This approach inherently accommodates irregularly sampled and sparse multivariate time series, without requiring imputation or resampling as a preprocessing step. We consider two latent manifolds, one of which is the hypersphere $\mathbb{S}^n$, motivated by the repetitive behavior of many industrial processes.

(2) Comprehensive benchmarking on six established benchmarks and one industrial benchmark with repetitive dynamics, showing that LSD performs consistently well against classical and state-of-the-art baselines, with the best overall rank on the established benchmarks and roughly twice the AUPRC of the strongest baseline on the industrial benchmark.

(3) Robustness under extreme sparsity. In a dedicated sparsity experiment simulating the limited data availability of industrial settings, we show that LSD ranks first at all levels of data availability, from full data down to $1\,\%$. In contrast, the ranking of the baselines changes substantially with sparsity: methods that perform well on full data are among the weakest under sparse observations.

2 Related work

Multivariate Time Series Anomaly Detection. MTSAD (Blázquez-García et al., 2021; Li & Jung, 2023; Zamanzadeh Darban et al., 2024) is a complex and challenging problem, with substantial progress and numerous methods proposed in recent years. Existing work broadly falls into two categories: *forecasting-based* (Hundman et al., 2018; Munir et al., 2019) and *reconstruction-based* (Ruff et al., 2018; Audibert et al., 2020; Tuli et al., 2022) approaches. Forecasting-based methods learn the temporal dynamics of a time series and use predictions over a given context window to anticipate future observations. The deviation between the predicted and true measured values serves as an anomaly score. However, such approaches typically model each variate independently, ignoring spatial dependencies between features (Hundman et al., 2018; Munir et al., 2019). This need not be a drawback: Pinet et al. (2026) show that most MTSAD benchmarks can be solved by treating a multivariate time series as a collection of univariate streams, an approach that frequently matches or outperforms complex multivariate models in both forecasting and anomaly detection. Moreover, forecasting-based methods assume that future observations can be predicted sufficiently well from available data. This assumption is questionable in applications where multiple legitimate continuations exist. An example is an industrial machine that switches between modes of operation depending

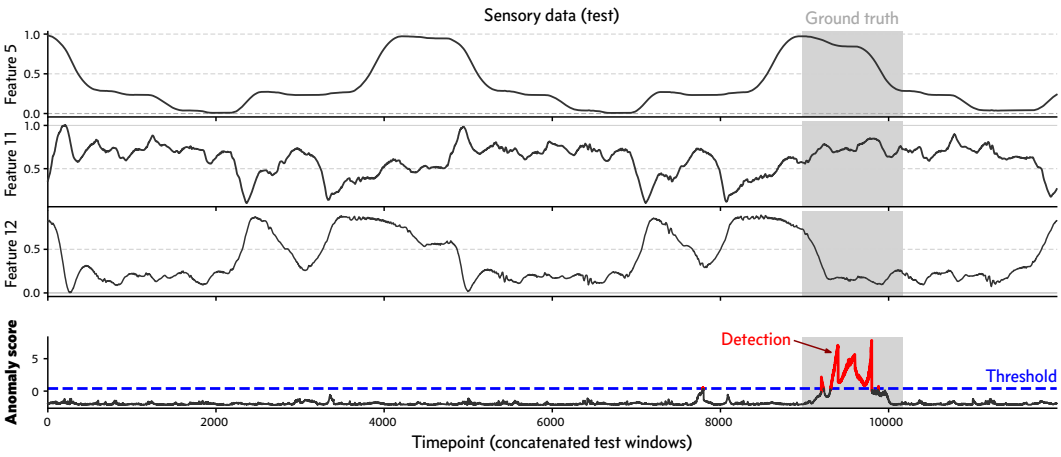


Figure 1: LSD applied to the QAPPD (Nosrati et al., 2027) benchmark. **Top:** Three representative features from trace 1, with ground-truth anomalous regions shaded in gray. **Bottom:** Corresponding anomaly score of LSD ($\mathbb{R}^n$), given by the negative log-likelihood of the observations; detected anomalies are marked in red.

on unobserved conditions (Schindler et al., 2025), e.g., by adjusting its actuators to compensate for a rise in ambient temperature that is not itself measured. Reconstruction-based methods, by contrast, learn a latent representation from which the input is regenerated, as in LSTM-based autoencoders (Malhotra et al., 2015) and OmniAnomaly (Su et al., 2019). However, such models may also reconstruct unseen anomalous patterns well, which reduces their sensitivity to genuine anomalies (Song et al., 2023; Kim & Kim, 2025). Graph-based methods have also been adopted for MTSAD (Deng & Hooi, 2021) to explicitly capture spatial inter-variate dependencies. However, methods that rely on a graph structure fixed after training tend to be less effective when the data distribution shifts over time (Cai et al., 2026).

Challenges in Industrial Time Series. Industrial time series are often sparse, irregularly sampled, and noisy (Colombi et al., 2024). Sparsity and irregular sampling are frequently even intended by design, e.g., to reduce the communication load (OPC Foundation, 2023). Existing MTSAD pipelines address this in two places. At the data level, observations are resampled onto regular grids, missing values are imputed, and signals are denoised (Wang et al., 2025); while this enables standard models, it can introduce bias and discard information about the sampling process itself (Belay et al., 2023). At the model level, robustness to noise must be provided by the architecture: transformer-based detectors (Xu et al., 2022; Tuli et al., 2022), for instance, tend to fit noisy observations and require additional regularization or noise-robust training (Kim & Kim, 2025). Both levels thus treat sparsity and noise as nuisances to be removed, rather than as properties of the data-generating process to be modeled, which is the perspective taken by LSD.

Neural Differential Equations for Time Series Modeling. A small body of work leverages Neural Ordinary Differential Equations (ODEs) (Chen et al., 2018; Rubanova et al., 2019), Neural Stochastic Differential Equations (SDEs) (Li et al., 2020a), and, more broadly, Neural Differential Equations (Oh et al., 2025) for time series anomaly detection (Sun et al., 2026; Glyn-Davies & Girolami, 2022). In principle, such models are well suited for irregularly sampled and sparse observations, since their continuous-time formulation accommodates asynchronous measurements and avoids resampling onto a fixed grid. The existing approaches, however, do not exploit these properties in either their modeling design or their evaluation. Sun et al. (2026) model the latent state with a deterministic Neural CDE and argue that it handles irregular and incomplete data without imputation, yet construct the driving path by cubic spline interpolation of the observations and evaluate exclusively on regularly sampled benchmarks. Glyn-Davies & Girolami (2022) learn the drift of an SDE with Gaussian processes directly in observation space for streaming anomaly detection, which captures stochasticity but neither learns a latent representation nor targets high-dimensional MTSAD benchmarks. In contrast, LSD combines stochastic latent dynamics with a mask-based treatment of missing observations and is explicitly evaluated under sparse, irregular sampling.

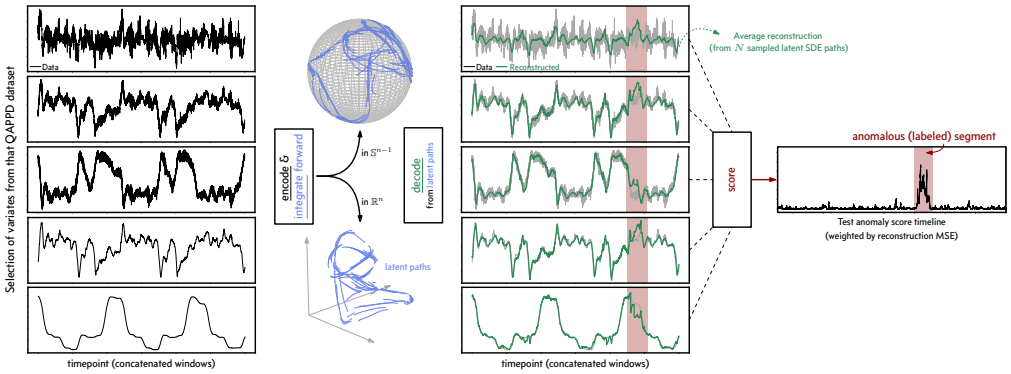


Figure 2: Overview of the proposed framework. During inference, a context window from the QAPPD dataset (five of 10 variates shown) is encoded into an initial state in a low-dimensional latent space, from which the learned SDE is integrated forward to the desired evaluation times. The resulting latent states are decoded into the observation space and scored to obtain point-wise anomaly scores.

3 METHOD

Motivated by requirements **R1**–**R3** (see Section 1), we investigate *structured, continuous-time latent stochastic dynamics* for reconstruction-based MTSAD. In the following, we present arguments for our modeling choices and how they specifically address these requirements. A schematic illustration of our approach is shown in Figure 2.

First, to represent shared temporal structure (**R2**), we posit a compact (low-dimensional) latent dynamical state underlying the potentially high-dimensional and redundant measurements. A separate observation likelihood relates this state to the measurements and represents measurement noise. *Second*, continuous-time evolution accommodates irregular observation times; together with inference and likelihood evaluation that account for partially observed inputs, this addresses **R1**. *Third*, to represent variability in normal evolution (**R3**), we allow stochastic perturbations throughout the latent trajectory, beyond uncertainty in its initial state. These perturbations can approximate unresolved influences within normal operation, such as small fluctuations in load or friction, that affect the subsequent motion. We retain explicit constraints on the form of this stochastic evolution by adopting the structured SDE family of Zeng et al. (2023), based on a prescribed matrix Lie group action (e.g., $\mathrm{SO}(n)$ acting on points on the sphere $\mathbb{S}^{n-1}$). This choice is motivated by the requirements of reconstruction-based detection: fitting normal observations well is insufficient if the model also reconstructs anomalous inputs accurately, see Section 2. *We therefore favor a parsimonious description of latent dynamics, with explicit structural constraints intended to limit such generalization while retaining the variability of normal behavior.*

The choice of latent geometry controls the extent of this "parsimony". In this work, we not only explore (1) a spherical latent space (which has an intuitive motivation for capturing roughly cyclic underlying processes) but also (2) latent dynamics in Euclidean space $\mathbb{R}^n$ with $\mathrm{GL}(n)$ action. To prevent over-generalization to anomalous inputs, we further employ a deliberately weak decoder, which encourages the SDE to capture underlying dynamics rather than overfit to individual observations.

3.1 PROBLEM SETUP

We consider a d-variate time series as a function $\mathbf{x} : [0, T] \to \mathbb{R}^d$, observed at potentially unevenly spaced time points t_j. To represent partial observations, each t_j is assigned a binary mask $\mathbf{m}(t_j) \in \{0, 1\}^d$, where $m_i(t_j) = 1$ if the i-th feature is observed. This yields a sequence $\big((t_1, \mathbf{x}(t_1), \mathbf{m}(t_1)), \ldots, (t_n, \mathbf{x}(t_n), \mathbf{m}(t_n))\big)$. An anomaly detection model is trained unsupervised on such sequences containing *only* benign, non-anomalous data. At test time, it maps a sequence of the same form to anomaly scores $s(t_j) \in \mathbb{R}$, one per observation time, where larger values indicate a higher *level of abnormality*. As illustrated in Figure 1, thresholding at some $r \in \mathbb{R}$ yields a binary decision: $\mathbb{1}(s(t_j) > r)$ marks the observation at t_j as anomalous under the learned model of normal behavior. In practice, a long time series is split into N windows of w time steps, resulting in a dataset

of windows $\mathbf{X}_1, \ldots, \mathbf{X}_N$ with $\mathbf{X}_n \in \mathbb{R}^{d \times w}$. Multi-trace datasets consist of K independent traces $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$: a separate model is trained and evaluated on each train/test pair, and metrics are aggregated across all traces.

3.2 Modeling Normal Behavior via Latent SDEs

To perform MTSAD, we rely on the work by Zeng et al. (2023) and model a time series $\mathbf{x}$ in terms of a latent time series $\mathbf{z}$ in a low-dimensional latent space using a variational autoencoder framework (Kingma & Welling, 2014). Latent time series are considered as draws from the law of a stochastic process Z solving an SDE on a homogeneous space H, i.e., on a space on which a matrix Lie group $\mathcal{G}$ acts transitively, meaning that for any $p, q \in H$ there is $g \in \mathcal{G}$ such that $g \cdot p = q$.

Specifically, following Zeng et al. (2023), we consider stochastic processes in H of the form $Z_t = G_t \cdot Z_0$, where Z_0 is the initial state distribution, and G_t solves an SDE in $\mathcal{G}$ with $G_0 = \mathbf{Id}$. The prior process $p(Z)$ is the Brownian motion in H acting on some prior initial state distribution, and the approximate posterior $q_\theta(Z|X) \approx p_\theta(Z|X)$ is given by an SDE that shares the same diffusion term but uses a drift term that is polynomial in time. The Lie-algebra-valued coefficients of the polynomial and the approximate posterior's initial state distribution are input-dependent via a recognition network with learnable weights. Sampling from the prior and approximate posterior then means numerically solving the SDE, producing discrete representations of a desired number of sample paths, each as a sequence of states $z(t_j)$ at predefined but in principle arbitrary values of t_j. The observation likelihood $p_\theta(X|Z)$ given a sample path z is then produced by a decoder network $g_\theta : H \subset \mathbb{R}^D \to \mathbb{R}^d$ that acts on all states $z(t_j)$ in the same way, mapping them to the means of the conditional marginals $p_\theta(\mathbf{x}(t_j)|\mathbf{z}(t_j)) = \mathcal{N}(g_\theta(z(t_j)), \sigma_\theta^2 \mathbf{I})$.

In practice, this means that the architecture consists of three key components: (1) a recognition network f_θ operates on a sequence $\big((t_1, \mathbf{x}(t_1), \mathbf{m}(t_1)), \ldots, (t_n, \mathbf{x}(t_n), \mathbf{m}(t_n))\big)$ representing a potentially irregularly sampled time series and maps it to the parameters of the latent SDE describing the approximate posterior process $q_\theta(Z|X)$, i.e., to the parameters of its initial distribution and the coefficients of the drift polynomial, (2) a numerical SDE solver that produces sample paths from the prior and approximate posterior, and (3) a decoder network g_θ that maps the sample paths to the parameters of the observation likelihood. We use mTAND (Shukla & Marlin, 2021) as the recognition network, a simple two-layer MLP as the decoder network, and a geometric Euler–Maruyama solver (Piggott & Solo, 2016). Both networks are trained jointly by maximizing the evidence lower bound (ELBO) $\mathbb{E}_{q_\theta(Z|X)}\big[\log p_\theta(X|Z)\big] - \mathrm{KL}\left(q_\theta(Z|X) \| p(Z)\right)$, or rather an empirical estimate thereof. For each sample path $\mathbf{z} \sim q_\theta(Z|X)$ from the approximate posterior, we sum the coordinate-wise log-likelihood $\log p_\theta(x_i(t)|\mathbf{z}(t))$ for all available coordinate-time pairs, i.e., for all (i, t) with $t \in \{t_1, \ldots, t_n\}$ and $m_i(t) = 1$, and take the mean over all sample paths. The KL term involves an integral over $q_\theta(Z|X)$ originating from Girsanov's theorem, which we estimate from the same sample paths.

Anomaly Detection. Once the latent SDE model is trained, we draw L sample paths $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)} \sim q_\theta(Z \mid \mathbf{X}_n)$ for each window $\mathbf{X}_n$ and compute the point-wise, per-variate negative log-likelihood $-\frac{1}{L}\sum_{l=1}^{L} \log p_\theta\big(\mathbf{X}_n \mid \mathbf{z}^{(l)}\big)$ at all observed entries ($m_i(t_j) = 1$), i.e., a Monte Carlo estimate of the expected log-likelihood term of the ELBO. For the Gaussian emission model, it equals, up to constants, the squared deviation from the mean reconstruction plus the variance across sample paths, so that the uncertainty captured by the latent SDE enters the score. Optionally, scores are min-max normalized on the training data. As a time step may appear in several overlapping windows, its scores are averaged across these windows. Finally, the per-variate scores $s_i(t_j)$ are aggregated into a univariate anomaly score $s(t_j)$, either (1) as a weighted sum $s(t_j) = \sum_{i=1}^{d} \alpha_i \, s_i(t_j)$ with weights $\alpha_i = \exp(1/\mathrm{MSE}_i) / \sum_{k=1}^{d} \exp(1/\mathrm{MSE}_k)$, where MSE_i is the mean squared training reconstruction error of variate i, so that well-reconstructed variates receive larger weights, or (2) as the ℓ_1-norm $s(t_j) = \sum_{i=1}^{d} |s_i(t_j)|$.

Implementation. Implementation details are provided in Section B.

4 Empirical Evaluation

Benchmark Datasets. We evaluate LSD and all baselines on seven benchmarks spanning multiple domains and anomaly types: Secure Water Treatment (SWaT) (Goh et al., 2017), Water Distribution (WaDi) (Ahmed et al., 2017), and Pooled Server Metrics (PSM) (Abdulaal & Lancewicki, 2021) each consist of a single continuous trace, whereas Mars Science Laboratory (MSL), Soil Moisture

Table 1: Performance comparison on **single-trace benchmarks**. The best, second-best, and third-best results per metric are highlighted. Within each section, methods are sorted by mean rank.

		SWaT			WaDi			PSM		
Rank ↓	Model	AUC ↑	AUPRC ↑	F1 ↑	AUC ↑	AUPRC ↑	F1 ↑	AUC ↑	AUPRC ↑	F1 ↑
5.56	COPOD	87.04 ± 0.00	76.59 ± 0.00	75.22 ± 0.00	78.78 ± 0.00	22.22 ± 0.00	35.98 ± 0.00	66.87 ± 0.00	43.24 ± 0.00	49.01 ± 0.00
5.56	iForest	84.93 ± 0.51	73.76 ± 0.63	74.49 ± 0.01	73.90 ± 0.52	17.53 ± 1.07	29.63 ± 2.10	70.01 ± 1.18	47.06 ± 1.61	50.59 ± 1.13
6.33	OCSVM	82.76 ± 0.00	72.65 ± 0.00	77.45 ± 0.00	52.27 ± 0.00	27.69 ± 0.00	13.04 ± 0.00	69.83 ± 0.00	51.29 ± 0.00	49.63 ± 0.00
7.22	kNN	82.54 ± 0.00	72.49 ± 0.00	78.83 ± 0.00	47.10 ± 0.00	5.00 ± 0.00	12.62 ± 0.00	73.91 ± 0.00	54.33 ± 0.00	54.86 ± 0.00
8.11	PCA	82.99 ± 0.00	73.34 ± 0.00	76.07 ± 0.00	49.95 ± 0.00	5.21 ± 0.00	13.53 ± 0.00	65.74 ± 0.00	47.44 ± 0.00	46.59 ± 0.00
11.78	LOF	48.20 ± 0.00	15.63 ± 0.00	25.31 ± 0.00	46.66 ± 0.00	4.94 ± 0.00	11.93 ± 0.00	71.57 ± 0.00	43.61 ± 0.00	54.48 ± 0.00
7.89	TcnED	83.91 ± 0.22	74.43 ± 0.23	76.50 ± 0.30	53.14 ± 0.73	11.04 ± 1.50	14.14 ± 1.67	61.56 ± 0.31	37.51 ± 0.28	46.05 ± 0.40
8.78	TranAD	83.07 ± 0.25	73.29 ± 0.17	76.07 ± 0.14	49.13 ± 0.11	5.14 ± 0.01	12.67 ± 0.08	66.34 ± 0.47	46.86 ± 0.73	48.43 ± 0.36
8.78	AnomalyTrans.	82.82 ± 0.21	72.67 ± 0.32	76.08 ± 0.34	46.36 ± 0.59	4.85 ± 0.04	12.84 ± 0.09	70.04 ± 1.44	48.50 ± 1.07	50.82 ± 1.75
9.56	USAD	82.85 ± 0.12	73.38 ± 0.06	76.81 ± 0.01	48.99 ± 0.16	5.06 ± 0.01	12.96 ± 0.26	61.78 ± 2.14	40.18 ± 2.26	46.32 ± 1.14
10.00	DeepIF	82.94 ± 0.35	72.01 ± 0.48	75.30 ± 0.43	48.04 ± 0.94	5.00 ± 0.09	12.60 ± 0.25	69.47 ± 0.67	46.11 ± 0.75	49.87 ± 0.86
10.11	COUTA	78.30 ± 10.09	50.85 ± 25.57	59.54 ± 18.64	44.46 ± 3.67	6.90 ± 4.46	13.30 ± 3.47	70.72 ± 2.72	45.98 ± 4.17	51.92 ± 2.54
11.22	DeepSVDD	46.79 ± 2.08	13.58 ± 0.99	27.75 ± 0.89	39.90 ± 1.43	4.40 ± 0.09	11.07 ± 0.12	77.03 ± 2.01	50.74 ± 3.54	57.99 ± 3.49
13.89	TimesNet	24.45 ± 0.54	8.72 ± 0.09	21.38 ± 0.00	50.47 ± 0.36	6.04 ± 0.09	11.72 ± 0.39	59.58 ± 0.15	39.24 ± 0.31	43.54 ± 0.05
5.44	LSD on $\mathbb{R}^n$ (ours)	82.26 ± 0.05	70.80 ± 0.05	74.86 ± 0.06	54.84 ± 0.25	20.63 ± 0.03	26.14 ± 0.03	77.23 ± 1.56	53.92 ± 1.17	59.87 ± 1.85
5.78	LSD on $\mathbb{S}^n$ (ours)	82.26 ± 0.05	70.80 ± 0.05	74.86 ± 0.06	54.95 ± 0.14	20.64 ± 0.01	26.14 ± 0.00	75.64 ± 1.29	52.37 ± 2.04	56.13 ± 1.77

Active and Passive (SMAP) (Hundman et al., 2018), and Server Machine Data (SMD) (Su et al., 2019) comprise multiple independent traces (e.g., SMD provides separate train and test sets per machine). In addition, QAPPD (Nosrati et al., 2027) is a multi-trace benchmark of trajectories from a real cyber-physical system performing a repetitive task. Descriptions, statistics, and licenses of all datasets are given in Section A.

All datasets are normalized with min-max scaling. The few missing values in WaDi ($< 0.05\,\%$) are imputed with a centered moving average over a window of 10 samples.

Baselines. We compare LSD with classical and deep learning baselines. Classical methods include Principal Component Analysis (PCA) (Shyu et al., 2006), k-Nearest Neighbors (kNN) (Ramaswamy et al., 2000), One-Class Support Vector Machine (OCSVM) (Schölkopf et al., 2001), Local Outlier Factor (LOF) (Breunig et al., 2000), Isolation Forest (iForest) (Liu et al., 2008), and Copula-Based Outlier Detection (COPOD) (Li et al., 2020b). Deep learning baselines include DeepSVDD (Ruff et al., 2018), USAD (Audibert et al., 2020), TranAD (Tuli et al., 2022), TcnED (Garg et al., 2022), AnomalyTransformer (Xu et al., 2022), DeepIF (Xu et al., 2023), TimesNet (Wu et al., 2023), and COUTA (Xu et al., 2024). We evaluate all methods under the same protocol, sharing hyperparameters where applicable.

Metrics. We evaluate all methods with three widely adopted anomaly detection metrics: the Area Under the Receiver Operating Characteristic curve (AUC), the Area Under the Precision-Recall Curve (AUPRC), and the F1-score. AUC and AUPRC are threshold-free and summarize performance across all operating points. AUPRC is particularly informative under the strong class imbalance typical of anomaly detection (see Table 6), as its chance level equals the anomaly ratio rather than 0.5. The F1-score requires a decision threshold r, which we select on the test set following Sarfraz et al. (2024): let $s \in \mathbb{R}^{N_{\text{test}}}$ denote the anomaly scores over all N_{test} test time steps and $l \in \{0,1\}^{N_{\text{test}}}$ the corresponding ground-truth labels. The candidate set R consists of the midpoints between consecutive sorted distinct values of s, so that every non-trivial binary labeling $\mathbb{1}_{s>r}$ is obtained by exactly one candidate. We report $\max_{r\in R} \mathrm{F1}(l, \mathbb{1}_{s>r})$, which corresponds to the best achievable F1-score and is therefore an optimistic estimate of performance at a fixed operating point. All results are reported as mean $\pm$ standard deviation over five independent runs with different random seeds.

4.1 Benchmark Results

Both our LSD variants, i.e., LSD with $\mathbb{R}^n$ and $\mathbb{S}^n$ as latent manifold, incorporate a data subsampling step during training, which randomly masks a subset of input samples at each iteration, acting as a regularizer that mitigates overfitting of the learned SDE. Since the mask is sampled independently at every update step during stochastic optimization, the model is exposed to the complete dataset over the course of training.

For the six widely used benchmarks, we follow the protocol of Cai et al. (2026) and use a window length of 100 samples with a stride of 100, i.e., non-overlapping windows. Results are reported in

Table 2: Performance comparison on **multi-trace benchmarks**. The best, second-best, and third-best results per metric are highlighted. Within each section, methods are sorted by mean rank.

		SMAP (55)			MSL (27)			SMD (28)		
Rank ↓	Model	AUC ↑	AUPRC ↑	F1 ↑	AUC ↑	AUPRC ↑	F1 ↑	AUC ↑	AUPRC ↑	F1 ↑
5.89	kNN	61.70 ± 0.00	26.71 ± 0.00	36.48 ± 0.00	67.02 ± 0.00	26.43 ± 0.00	39.14 ± 0.00	81.80 ± 0.00	43.13 ± 0.00	48.99 ± 0.00
6.89	PCA	64.89 ± 0.00	27.18 ± 0.00	37.39 ± 0.00	63.28 ± 0.00	20.70 ± 0.00	34.06 ± 0.00	80.61 ± 0.00	43.55 ± 0.00	48.98 ± 0.00
9.22	OCSVM	63.87 ± 0.00	27.13 ± 0.00	34.99 ± 0.00	60.62 ± 0.00	24.18 ± 0.00	31.87 ± 0.00	79.21 ± 0.00	41.63 ± 0.00	40.00 ± 0.00
12.00	COPOD	62.62 ± 0.00	20.28 ± 0.00	33.12 ± 0.00	63.74 ± 0.00	20.24 ± 0.00	33.37 ± 0.00	77.70 ± 0.00	29.57 ± 0.00	33.48 ± 0.00
12.11	LOF	58.32 ± 0.00	25.67 ± 0.00	33.77 ± 0.00	61.48 ± 0.00	20.74 ± 0.00	32.79 ± 0.00	71.20 ± 0.00	30.71 ± 0.00	36.73 ± 0.00
15.44	iForest	57.86 ± 0.21	19.55 ± 0.17	29.33 ± 0.12	57.39 ± 0.07	17.45 ± 0.14	27.06 ± 0.28	77.43 ± 0.15	27.12 ± 0.44	32.43 ± 0.35
5.11	TcnED	65.35 ± 0.48	26.76 ± 0.43	37.46 ± 0.50	63.77 ± 0.38	21.20 ± 0.09	34.44 ± 0.29	81.33 ± 0.17	45.36 ± 0.31	49.72 ± 0.28
5.22	USAD	59.90 ± 0.29	26.05 ± 0.15	37.22 ± 0.19	69.51 ± 0.01	28.67 ± 0.08	39.86 ± 0.03	86.11 ± 0.27	37.93 ± 0.22	49.49 ± 0.39
5.89	DeepIF	63.13 ± 1.29	29.68 ± 0.59	40.08 ± 0.73	61.73 ± 0.40	22.32 ± 0.84	35.14 ± 0.75	87.58 ± 0.16	37.66 ± 0.21	48.66 ± 0.25
6.67	TranAD	62.03 ± 1.82	24.76 ± 0.39	35.29 ± 0.80	64.05 ± 1.21	21.39 ± 0.42	33.75 ± 0.39	82.92 ± 0.41	44.46 ± 0.18	49.47 ± 0.27
8.78	TimesNet	55.12 ± 0.98	21.21 ± 1.03	33.00 ± 0.80	65.24 ± 0.63	24.84 ± 0.27	36.27 ± 0.66	86.34 ± 0.16	39.37 ± 0.63	45.29 ± 0.47
9.44	COUTA	59.68 ± 5.67	24.14 ± 0.76	35.06 ± 0.42	63.52 ± 1.21	20.56 ± 0.84	31.83 ± 0.8	82.07 ± 0.49	43.78 ± 0.40	48.92 ± 0.47
10.33	DeepSVDD	64.78 ± 1.11	22.12 ± 1.07	35.45 ± 0.68	64.05 ± 1.31	20.06 ± 1.05	31.19 ± 1.46	82.52 ± 0.68	25.32 ± 2.01	35.82 ± 1.89
14.22	AnomalyTrans.	57.80 ± 3.88	20.49 ± 1.84	31.51 ± 1.95	57.65 ± 3.68	17.64 ± 1.31	26.99 ± 1.86	80.08 ± 0.33	28.96 ± 0.96	38.53 ± 0.45
4.00	LSD on $\mathbb{R}^n$ (ours)	62.55 ± 0.37	29.97 ± 0.48	38.52 ± 0.29	67.10 ± 0.46	30.62 ± 0.92	42.11 ± 0.51	82.74 ± 0.37	40.94 ± 0.54	46.85 ± 0.53
4.78	LSD on $\mathbb{S}^n$ (ours)	61.25 ± 1.30	29.06 ± 0.69	37.45 ± 0.25	66.31 ± 1.63	28.93 ± 1.24	40.28 ± 1.15	81.89 ± 0.56	42.31 ± 1.08	49.00 ± 0.67

Tables 1 and 2, each organized into three parts: classical machine learning methods, deep learning-based methods, and the two proposed LSD variants on $\mathbb{S}^n$ and $\mathbb{R}^n$. Table 1 covers the single-trace benchmarks SWaT, WaDi, and PSM, and Table 2 the multi-trace benchmarks SMAP, MSL, and SMD, with the number of traces given in parentheses.

To compare methods across benchmarks, we rank each metric within the table and report the mean rank over all metrics and benchmarks, yielding a measure of overall performance. The ranks displayed alongside each method's name in Tables 1 and 2 are computed over the benchmarks of the corresponding table; the joint ranking over all six benchmarks is shown in Table 3. On the single-trace benchmarks, $\mathbb{R}^n$ achieves the best overall rank, with the $\mathbb{S}^n$ variant ranking fourth. On the multi-trace benchmarks, both variants outperform all baselines, with $\mathbb{R}^n$ improving the mean rank by ≈ 1.11 over the next best baseline. Notably, the baselines show complementary strengths, with classical methods performing well on single-trace and deep learning methods on multi-trace benchmarks, whereas LSD ranks ahead of all baselines in both settings (Table 3).

Table 3: Method rankings sorted by overall rank, reported over all benchmarks combined.

Model	Overall	Single Trace	Multi Trace
LSD on $\mathbb{R}^n$ (ours)	4.72	5.44	4.00
LSD on $\mathbb{S}^n$ (ours)	5.28	5.78	4.78
TcnED	6.50	7.89	5.11
kNN	6.56	7.22	5.89
USAD	7.39	9.56	5.22
kNN	7.50	8.11	6.89
TranAD	7.72	8.78	6.67
OCSVM	7.78	6.33	9.22
DeepIF	7.94	10.00	5.89
COPOD	8.78	5.56	12.00
COUTA	9.78	10.11	9.44
iForest	10.50	5.56	15.44
DeepSVDD	10.78	11.22	10.33
TimesNet	11.33	13.89	8.78
AnomalyTrans.	11.50	8.78	14.22
LOF	11.94	11.78	12.11

We further evaluate the proposed method on real-world industrial data from the QAPPD (Nosrati et al., 2027) benchmark. Unlike for the other benchmarks, the underlying dynamics are known from the data acquisition process: the cyber-physical system performs a repetitive pick-and-place task with a robot, returning to approximately the same initial state after each iteration. Each of the 10 traces is evenly subsampled to 10 % of its original length, resulting in a sampling rate of 10 Hz. One repetition spans about 500 samples on average, with varying duration across repetitions.

This knowledge informs the choice of context window. Ideally, a window would cover approximately one *full* repetition, allowing a model to capture the complete temporal structure of the task; for LSD, this aligns with the inductive bias of a continuous latent trajectory, which could then represent the repeated dynamics explicitly within a single window. However, the average repetition length of QAPPD exceeds the window length that LSD can process at reasonable computational cost. We therefore set the window length to 200 samples, i.e., roughly 40 % of an average repetition, and compensate with a stride of 20, so that consecutive windows overlap by 90 % and each part of a repetition is observed in multiple contexts during training. The same window length and stride are used for all baselines and both LSD variants. At test time, each time step is covered by several windows, and its final anomaly score is the average over all windows containing it.

The repetitive structure also bears on the choice of latent geometry. Since each repetition of the pick-and-place task returns the system to approximately the same state, the latent trajectory is expected to

trace out approximately closed loops. A compact latent space such as $\mathbb{S}^n$ represents such recurrent dynamics without drift, as trajectories are bounded and revisit the same region by construction, whereas in $\mathbb{R}^n$ this recurrence must be learned from data alone. More generally, a latent space whose geometry matches that of the data-generating process has been argued to facilitate learning (Falorsi et al., 2018), and hyperspherical latent spaces have been shown to outperform Euclidean ones in low dimensions (Davidson et al., 2018). Moreover, the constrained geometry of $\mathbb{S}^n$ acts as a regularizer on the latent dynamics, which we expect to be most beneficial when observations are scarce. We therefore hypothesize that $\mathbb{S}^n$ is advantageous on QAPPD, in particular under sparse observations.

The results for the QAPPD benchmark are reported in Table 4. Both LSD variants outperform all baselines on this benchmark, with $\mathbb{R}^n$ ranking first (avg. rank 1.0) and $\mathbb{S}^n$ second (2.0). The strongest baseline, LOF (3.0), lies within one standard deviation of $\mathbb{S}^n$ on AUC, but is clearly outperformed by both $\mathbb{R}^n$ and $\mathbb{S}^n$ in terms of AUPRC and F1-score. The best deep-learning baseline, DeepSVDD, reaches an average rank of 4.0. Both results stand in stark contrast to the performance of these methods on the remaining benchmarks (see Table 3): LOF ranks last among all methods there (avg. rank 11.94), and DeepSVDD ranks among the lowest (avg. rank 10.78).

Table 4: Performance comparison on the QAPPD benchmark. The best, second-best, and third-best results per metric are highlighted.

		QAPPD (10)		
Rank ↓	**Model**	**AUC** ↑	**AUPRC** ↑	**F1** ↑
3.00	LOF	84.80 ± 0.00	20.08 ± 0.00	28.99 ± 0.00
5.67	KNN	72.87 ± 0.00	15.50 ± 0.00	21.98 ± 0.00
8.00	COPOD	62.02 ± 0.00	6.76 ± 0.00	12.27 ± 0.00
9.00	OCSVM	61.83 ± 0.00	5.25 ± 0.00	9.69 ± 0.00
9.33	IForest	63.58 ± 2.47	5.07 ± 0.44	9.54 ± 0.30
14.00	PCA	56.93 ± 0.00	4.22 ± 0.00	8.98 ± 0.00
4.00	DeepSVDD	82.95 ± 0.86	19.28 ± 0.98	27.69 ± 0.82
5.33	DeepIF	77.39 ± 1.89	15.05 ± 0.49	22.62 ± 0.85
10.00	USAD	49.75 ± 0.68	7.74 ± 0.42	13.32 ± 0.69
10.67	AnomalyTrans.	59.05 ± 3.32	5.17 ± 1.00	9.28 ± 1.23
12.00	TimesNet	59.49 ± 1.01	4.98 ± 0.16	8.78 ± 0.45
13.33	COUTA	58.72 ± 3.45	4.89 ± 0.56	8.73 ± 1.09
13.67	TcnED	50.50 ± 1.86	4.88 ± 0.23	9.10 ± 0.34
15.00	TranAD	51.99 ± 0.85	4.45 ± 0.14	8.62 ± 0.18
1.00	LSD on $\mathbb{R}^n$ (ours)	88.77 ± 0.53	39.35 ± 4.87	50.75 ± 2.37
2.00	LSD on $\mathbb{S}^n$ (ours)	86.70 ± 2.58	33.86 ± 3.54	46.53 ± 3.50

The results also reveal an unexpected finding regarding the two LSD variants: contrary to our hypothesis that $\mathbb{S}^n$ is particularly suited for cyclic data, the $\mathbb{R}^n$ variant outperforms $\mathbb{S}^n$ on QAPPD, suggesting that the additional flexibility of the Euclidean latent space is beneficial for this benchmark.

Figure 2 includes a qualitative view of the learned geometry: it shows the latent trajectories of LSD on the first trace of QAPPD, visualized directly in the latent spaces $\mathbb{S}^2$ and $\mathbb{R}^3$ without any further projection. The learned trajectories form recurring, approximately closed loops, reflecting the repetitive nature of the underlying pick-and-place task.

4.2 Sparsity Evaluation

To evaluate the robustness of the proposed method under varying levels of data sparsity, we compare against all classical machine learning baselines and the three best-performing deep-learning methods, selected by mean rank from Table 4. All methods use the same fixed hyperparameter configuration as in Section 4.1; only the sparsity level is varied. To simulate data sparsity and the resulting irregular sampling, we generate bursty masks that specify which observations are available; the generation procedure is described in Appendix B.5, and example masks are shown in Figure 3. Masking is applied uniformly across the training, validation, and test splits.

Unlike LSD, the baselines cannot natively handle sparse and irregularly sampled inputs. We therefore apply two standard preprocessing steps: first, the data is *resampled* to the original frequency, and second, missing values on each variate are imputed between the nearest preceding and following observations. Values at the beginning or end of a sequence are filled by carrying the first or last available observation backward or forward, respectively. As imputation strategies, we employ linear interpolation, cubic spline interpolation, and forward filling.

We report results for *sparse burst subsampling* with $1\,\%$ of the original data retained, averaged over 5 runs (Table 5). Under this level of sparsity, both LSD variants outperform all tested baselines, with average ranks of 1.0 for $\mathbb{S}^n$ and 2.67 for $\mathbb{R}^n$. Notably, the best-ranked method of each baseline group under full data availability drops to the bottom within its group: DeepSVDD reaches an average rank of 8.67 and LOF of 10.67.

A further observation concerns the two LSD variants. Whereas $\mathbb{R}^n$ performs best in Table 4, $\mathbb{S}^n$ ranks first on all metrics under *sparse observations*. This is consistent with our hypothesis from Section 4.1 that the compact hyperspherical latent space provides a beneficial inductive bias under extreme observational scarcity.

Further results, including a sparsity level of $5\,\%$ as well as additional imputation strategies, are given in Appendix C.2.

Runtime and ablations. LSD incurs a substantially higher training cost than the baselines due to the numerical integration of the latent SDE, while its inference cost is comparable to that of the deep-learning baselines (Appendix C.1). An ablation on PSM, replacing the stochastic latent dynamics by a deterministic latent Neural ODE and the weak decoder by an over-parametrized one, is reported in Appendix C.3.

Table 5: Sparsity evaluation on the QAPPD benchmark with $1\,\%$ of the data retained. Baselines receive linearly interpolated inputs; LSD operates on the sparse data directly. Ranks are computed only over the results shown in this table.

		QAPPD (1%)		
Rank ↓	Model	AUC ↑	AUPRC ↑	F1 ↑
4.33	COPOD (linear)	55.25 ± 2.25	4.46 ± 0.70	9.95 ± 1.09
6.67	OCSVM (linear)	53.99 ± 3.61	4.02 ± 0.92	9.77 ± 2.26
7.33	KNN (linear)	55.17 ± 3.48	3.40 ± 0.42	9.46 ± 1.32
7.67	IForest (linear)	54.35 ± 2.54	3.83 ± 0.34	9.06 ± 1.01
8.33	PCA (linear)	52.91 ± 3.63	3.72 ± 0.80	9.13 ± 1.84
10.67	LOF (linear)	51.70 ± 4.62	2.91 ± 0.28	7.95 ± 0.85
5.00	DeepIF (linear)	52.64 ± 5.08	4.97 ± 1.75	10.29 ± 1.97
5.67	USAD (linear)	51.46 ± 4.26	4.54 ± 1.38	10.48 ± 1.50
6.67	DeepSVDD (linear)	54.53 ± 5.64	3.42 ± 0.37	9.83 ± 1.38
1.00	LSD on $\mathbb{S}^n$ (ours)	58.87 ± 2.48	6.97 ± 0.96	10.74 ± 1.11
2.67	LSD on $\mathbb{R}^n$ (ours)	56.46 ± 2.74	6.07 ± 1.24	10.13 ± 1.34

5 Discussion and Limitations

We have presented LSD, an SDE-based framework for MTSAD that models normal behavior through latent dynamics on a homogeneous space, here $\mathbb{R}^n$ or $\mathbb{S}^n$, and uses the likelihood of the decoded observations as the anomaly score. Although the framework also supports forecasting and generative modeling, we do not exploit these capabilities. LSD achieves the best overall rank across the established benchmarks, with a more pronounced advantage on multi-trace than on single-trace benchmarks, and outperforms all baselines on QAPPD. Under bursty subsampling, a sparsity pattern common in industrial control systems, LSD ranks first at every tested level of data availability, whereas the relative performance of the baselines changes substantially with the amount of available data.

The results on QAPPD further indicate that the appropriate latent geometry depends on the characteristics of the data. Although the inductive bias of $\mathbb{S}^n$ suggests an advantage for cyclic data, $\mathbb{R}^n$ performs better under full data availability, presumably owing to its greater flexibility. Under severe sparsity, in contrast, $\mathbb{S}^n$ prevails, which we attribute to the regularizing effect of the constrained, compact geometry. The choice of latent geometry is thus non-trivial and should be informed by both the structure of the underlying dynamics and the data availability.

Limitations. The primary limitation of LSD relative to all baseline methods is its substantially higher training cost, inherent to the complexity of fitting a latent SDE with a numerical solver. Additionally, the sparsity experiment is constrained by the use of artificially introduced "missingness", which may not fully reflect the irregular and heterogeneous sparsity patterns encountered in real industrial deployments. Finally, unlike some prior works, we deliberately refrained from extensive hyperparameter tuning, including the selection of regularization strategies and of score and data normalization techniques, which leaves open the possibility that further tuning could yield additional performance gains.

Future Work. The sparsity experiments on QAPPD highlight the need for benchmarks that exhibit cyclic dynamics alongside natural sparsity and true irregular sampling, i.e., characteristics that are prevalent in real industrial deployments but remain underrepresented in current MTSAD benchmarks. Developing and releasing such benchmarks represents an important direction for future work. Beyond benchmarking, the LSD framework is a natural candidate for extension to related tasks such as *root cause analysis*, where the per-variate log-likelihood may serve as a principled indicator for isolating the source of anomalous behavior. Finally, the results of this work raise the more general question of how to select the latent geometry for a given type of data. A principled criterion, informed by the structure of the underlying dynamics, the data availability, or both, would turn this choice from an empirical design decision into a data-driven one, for cyclic and non-cyclic data alike.

ETHICS STATEMENT

This work does not involve human subjects or personal data. All datasets used are publicly available, some upon registration, and are used as released by their respective authors. Potential societal impacts are discussed in section D. The authors declare no conflicts of interest related to this work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing our method, specifically for writing parts of the source code, and as a discussion partner providing feedback on the methodology and the experimental design. All AI-assisted code was reviewed and tested by the authors, who retained full control over the implementation, and all methodological and experimental decisions were made by the authors. We have not used generative AI tools to generate data sets, develop the conceptual framework, formulate or prove mathematical claims, propose hypotheses, or clean or reformat datasets; translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper for grammar, spelling, and readability. All text originates from the authors, and every suggested edit was reviewed before being adopted. Generative AI tools were not used for literature search or reference management. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The problem formalization is given in Section 3.2, and the latent SDE model, training objective, and point-wise anomaly score are described in Section 3.3 and Appendix B. The complete source code is included in the supplementary material and will be released publicly upon the paper beeing published; it covers data preparation, training, evaluation, and the sparsity experiments, and fixes all random seeds, including those for generating the sparsity masks. All benchmarks are publicly available; Appendix A lists their sources, licenses, and statistics, and Section 4 and Appendix B document all preprocessing steps and data splits. The hyperparameters of LSD are reported in Table 7, and all baselines use the default configurations of PyOD and DeepOD (Appendix B). Metrics and the F1 threshold selection are specified in Section 4, and the sparsity mask generation in Section 4.2. All results are averaged over five independent runs. Appendix B describes the computing infrastructure, and Appendix C reports training runtimes.

REFERENCES

Ahmed Abdulaal and Tomer Lancewicki. Real-time synchronization in neural networks for multivariate time series anomaly detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3570–3574, 2021. doi: 10.1109/ICASSP39728.2021.9413847.

Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, CPS Week '17, pp. 25–28. ACM, April 2017. doi: 10.1145/3055366.3055375.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3395–3404, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403392.

Mohammed Ayalew Belay, Sindre Stenen Blakseth, Adil Rasheed, and Pierluigi Salvo Rossi. Unsupervised anomaly detection for iot-based multivariate time series: Existing solutions, performance analysis and future directions. *Sensors*, 23(5), 2023. ISSN 1424-8220. doi: 10.3390/s23052844.

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3), April 2021. ISSN 0360-0300. doi: 10.1145/3444690.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, June 2000.

Jinyu Cai, Yuan Xie, Glynnis Lim, Yifang Yin, Roger Zimmermann, and See-Kiong Ng. Self-perturbed anomaly-aware graph dynamics for multivariate time-series anomaly detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=hJJnwcvE2M.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BysvGP5ee.

Lorenzo Colombi, Michele Vespa, Nicolas Beletti, Brina Matteo, Simon Dahdal, Filippo Tabanelli, Elena Bellodi, Mauro Tortonesi, Cesare Stefanelli, and Massimiliano Vignoli. Multivariate time series anomaly detection in industry 5.0. In *3rd Italian Conference on Big Data and Data Science*, 2024. doi: arXiv:2503.15946.

Andrew A. Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2020. doi: 10.1109/JIOT.2019.2958185.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4027–4035, 2021.

Luca Falorsi, Pim de Haan, Tim R. Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S. Cohen. Explorations in homeomorphic variational auto-encoding. In *ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018. URL https://sites.google.com/view/tadgm.

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, 2022. doi: 10.1109/TNNLS.2021.3105827.

Alex Glyn-Davies and Mark Girolami. Anomaly detection in streaming data with gaussian process based stochastic differential equations. *Pattern Recognition Letters*, 153:254–260, January 2022. ISSN 0167-8655. doi: 10.1016/j.patrec.2021.12.017.

Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In Grigore Havarneanu, Roberto Setola, Hypatia Nassopoulos, and Stephen Wolthusen (eds.), *Critical Information Infrastructures Security*, pp. 88–99, Cham, 2017. Springer International Publishing. ISBN 978-3-319-71368-7.

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, pp. 387–395, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219845.

Gyeongmin Kim and Min-cheol Kim. Multivariate time series anomaly detection using tranad: Assessing the effectiveness of noise generalization techniques, 2025. URL http://dx.doi.org/10.2139/ssrn.5238491.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

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

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3870–3882. PMLR, 26–28 Aug 2020a.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. COPOD: Copula-Based Outlier Detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1118–1123, Los Alamitos, CA, USA, November 2020b. IEEE Computer Society. doi: 10.1109/ICDM50108.2020.00135.

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, pp. 413–422, USA, 2008. IEEE Computer Society. ISBN 9780769535029. doi: 10.1109/ICDM.2008.17.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *23rd European Symposium on Artificial Neural Networks, ESANN 2015, Bruges, Belgium, April 22-24, 2015*, 2015. URL https://www.esann.org/sites/default/files/proceedings/legacy/es2015-56.pdf.

Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2019. doi: 10.1109/ACCESS.2018.2886457.

Khayyam Nosrati, Martin Uray, Saverio Messineo, Olaf Sassnick, and Stefan Huber. Federated learning for multivariate time series anomaly detection in industrial automation. In Manuela Geiß, Ulrich Göhner, Sebnem Gül-Ficicir, Bernhard Moser, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil (eds.), *Database and Expert Systems Applications - DEXA 2026 Workshops*, pp. 90–101, Cham, 2027. Springer Nature Switzerland. ISBN 978-3-032-34202-7.

YongKyung Oh, Seungsu Kam, Jonghun Lee, Dong-Young Lim, Sungil Kim, and Alex Bui. Comprehensive review of neural differential equations for time series analysis, 2025. URL https://arxiv.org/abs/2502.09885.

OPC Foundation. OPC Unified Architecture - Part 1: Overview and Concepts. https://opcfoundation.org, 2023. Version 1.05.

Marc Piggott and Victor Solo. Geometric euler-maruyama schemes for stochastic differential equations in so(n) and se(n). *SIAM Journal on Numerical Analysis*, 54(4):2490–2516, 2016. ISSN 00361429. URL http://www.jstor.org/stable/44018130.

Marc Pinet, Julien Cumin, Samuel Berlemont, and Dominique Vaufreydaz. Anomalies in multivariate time series benchmarks are mostly univariate, 2026. URL https://arxiv.org/abs/2606.02670.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pp. 427–438, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132174. doi: 10.1145/342009.335437.

Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 10–15 Jul 2018.

M. Saquib Sarfraz, Mei-Yen Chen, Lukas Layer, Kunyu Peng, and Marios Koulakis. Position: quo vadis, unsupervised time series anomaly detection? In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Simon Schindler, Elias Steffen Reich, Saverio Messineo, and Stefan Huber. Topology-driven identification of repetitions in multi-variate time series. In *Proceedings of the 6th Interdisciplinary Data Science Conference (iDSC'25)*, 05 2025. URL https://arxiv.org/abs/2505.10004.

Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001. ISSN 0899-7667. doi: 10.1162/089976601750264965.

Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=4c0J6lwQ4_.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and Li Wu Chang. *Principal Component-based Anomaly Detection Scheme*, pp. 311–329. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-31229-1. doi: 10.1007/11539827_18.

Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pp. 245–248, 2012.

Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. Memto: memory-guided transformer for multivariate time series anomaly detection. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Vinicius M. A. Souza, Denis M. dos Reis, André G. Maletzke, and Gustavo E. A. P. A. Batista. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, 34(6):1805–1858, July 2020. ISSN 1573-756X. doi: 10.1007/s10618-020-00698-5. URL http://dx.doi.org/10.1007/s10618-020-00698-5.

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, pp. 2828–2837, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330672.

Xin Sun, Heng Zhou, Yuhao Wu, and Chao Li. State-derivative-aware neural controlled differential equations for multivariate time series anomaly detection and diagnosis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(30):25727–25735, March 2026. ISSN 2159-5399. doi: 10.1609/aaai.v40i30.39770.

Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. Tranad: deep transformer networks for anomaly detection in multivariate time series data. *Proc. VLDB Endow.*, 15(6):1201–1214, February 2022. ISSN 2150-8097. doi: 10.14778/3514061.3514067.

Jun Wang, Wenjie Du, Yiyuan Yang, Linglong Qian, Wei Cao, Keli Zhang, Wenjia Wang, Yuxuan Liang, and Qingsong Wen. Deep learning for multivariate time series imputation: a survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/1187.

Philip B. Weerakody, Kok Wai Wong, Guanjin Wang, and Wendell Ela. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.02.046.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Renjie Wu and Eamonn J. 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, 2023. doi: 10.1109/TKDE.2021.3112126.

Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023. doi: 10.1109/TKDE.2023.3270293.

Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Trans. on Knowl. and Data Eng.*, 36(11):5723–5736, November 2024. ISSN 1041-4347. doi: 10.1109/TKDE.2024.3393996.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Comput. Surv.*, 57(1), October 2024. ISSN 0360-0300. doi: 10.1145/3691338.

Sebastian Zeng, Florian Graf, and Roland Kwitt. Latent sdes on homogeneous spaces. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. URL http://jmlr.org/papers/v20/19-011.html.

Supplementary Material

This supplementary material provides additional details supporting the main paper. Section A describes all benchmark datasets and reports their statistics. Section B documents the full experimental setup, including data preparation, hyperparameter configurations, baseline implementations, the generation of the sparsity masks, and the computing infrastructure. Additional experimental results, including runtime measurements, extended sparsity results, and an ablation study, are given in Section C. Potential societal impacts are discussed in Section D.

A Benchmark Datasets

This section describes all benchmark datasets together with their sources and licenses. Table 6 summarizes their statistics, including the number of traces, the total number of training and test samples, and the anomaly ratio of the test split. All statistics refer to the datasets as released, i.e., before any preprocessing.

- **SWaT** (Goh et al., 2017) and **WaDi** (Ahmed et al., 2017) contain sensor and actuator data from a water treatment and a water distribution testbed, respectively, with anomalies caused by cyber-physical attacks. Both are published by iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design; access requires registration.[1]
- **PSM** (Abdulaal & Lancewicki, 2021) contains multivariate telemetry from internal server infrastructure at eBay Inc., labeled as normal or anomalous, and is released under the CC BY 4.0 license.[2]
- **MSL** and **SMAP** (Hundman et al., 2018) contain telemetry from NASA's Curiosity rover and the SMAP satellite, with anomalies corresponding to reported telemetry faults. The data is made available by the authors without an explicit license.[3]
- **SMD** (Su et al., 2019) contains server machine telemetry from a large internet company, with anomalies corresponding to abnormal operating states, and is released under the MIT License.[4]
- **QAPPD** (Nosrati et al., 2027) comprises trajectories from a real cyber-physical system performing a repetitive task, i.e., the system repeatedly returns to approximately the same region of its state space without strict temporal periodicity. It is released under the CC BY-SA 4.0 license.[5]

B Experimental Setup

B.1 Data Preparation and Splits

We follow a protocol similar to that proposed by Cai et al. (2026). Each benchmark dataset consists of a training set and a test set. The training set contains only benign samples and is used for model fitting, whereas the test set contains both benign and anomalous samples and is used to evaluate anomaly detection performance. For methods that require a validation set, we randomly reserve $10\,\%$

[1] https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/
[2] https://github.com/eBay/RANSynCoders
[3] https://pds-atmospheres.nmsu.edu/data_and_services/atmospheres_data/Mars/Mars.html, https://nsidc.org/data/smap
[4] https://github.com/NetManAIOps/OmniAnomaly
[5] https://github.com/JRC-ISIA/industrial-federated-learning

Table 6: Statistics of the used benchmarks.

Dataset	Traces	Features	Length		Anomalies (%)
			Train	Test	
MSL	27	55	58,317	73,729	10.48
PSM	1	25	132,481	87,841	27.76
QAPPD	10	13	1,820,010	1,820,000	2.66
SMAP	55	25	140,825	444,035	12.83
SMD	28	38	708,405	708,420	4.16
SWaT	1	51	496,800	449,919	11.97
WaDi	1	130	784,570	172,803	5.77

of the training data for validation and use the remaining $90\,\%$ for training. For methods that do not require a validation set, e.g., PCA, we use the full training set for model fitting.

Following Cai et al. (2026), all methods that require windowed inputs use a window size of 100 and a stride of 100, resulting in non-overlapping windows, except for QAPPD, where we use a window size of 200 and a stride of 20 (see Section 4.1). For the validation set, 10% of the full windows are sampled.

B.2 Hyperparameter Settings

For the baselines, we do not tune hyperparameters manually but use the default settings of the respective implementations.

The hyperparameters of LSD are listed in Table 7. The maximum number of epochs is lower for multi-trace than for single-trace benchmarks to limit the total training time, since a separate model is trained per trace. The latent dimension of $\mathbf{z}$ (for both $\mathbb{R}^n$ and $\mathbb{S}^n$), the hidden dimension h, and the decoder hidden dimension are set by a rule of thumb based on the number of input features, following the original implementation of Zeng et al. (2023). We tested different weightings of the ELBO terms and different subsampling ratios, but observed no major differences in detection performance; higher subsampling ratios (> 0.8) mainly slowed down the decrease of the training loss. The choices with a noticeable effect were the score normalization and the score aggregation.

Table 7: Hyperparameters of LSD for each benchmark.

Parameter	MSL	PSM	QAPPD	SMAP	SMD	SWaT	WaDi
Max. Epochs	690	2000	1500	690	690	2000	2000
Batch Size	512	512	256	512	512	256	128
Learning Rate	0.001	0.1	0.01	0.001	0.001	0.05	0.05
Latent Dim. (z)	16	8	4	16	12	16	14
Hidden Dim. (h)	26	14	20	26	18	24	56
Polynomial Degree K (drift)	12	6	5	12	8	12	16
Decoder Hidden Dim.	24	12	11	24	16	22	64
KL_0 Weight	0.0001	0.0001	0.001	0.0001	0.0001	0.0001	0.0001
KL_p Weight	0.001	0.001	0.01	0.001	0.001	0.001	0.001
Likelihood Weight	1	1	1	1	1	1	100.0
σ	0.05	0.2	0.2	0.05	0.075	0.2	0.2
Subsample Ratio	0.4	0.5	0.1	0.4	0.4	0.5	0.5
Normalize Score	✓	✓	✓	✓	✓	×	×
Score Aggregation	ℓ_1	ℓ_1	mse-weighted	ℓ_1	ℓ_1	ℓ_1	ℓ_1

B.3 Baseline Implementation

Classical baseline methods (kNN, LOF, iForest, OCSVM, COPOD, and PCA) are implemented using the PyOD (Zhao et al., 2019) library[6]. Deep learning-based baselines (DeepSVDD, USAD,

[6] https://pyod.readthedocs.io/en/latest/index.html

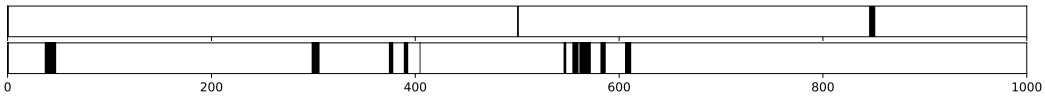


Figure 3: Exemplary masks at sparsity levels 1 % and 5 %. Black indicates available observations.

TcnED, TranAD, AnomalyTransformer, DeepIF, TimesNet, and COUTA) are implemented using the DeepOD (Xu et al., 2023) library[7].

B.4 Method Implementation

LSD builds on the implementation of Zeng et al. (2023), publicly available online[8]. In line with Section 3.2, the architecture consists of three components:

- The **recognition network** maps the time-indexed sequence of observations to the parameters of the approximate posterior of the latent SDE, i.e., its initial state distribution and the drift coefficients, and is implemented as a multi-time attention network (mTAND) (Shukla & Marlin, 2021).
- The **latent SDE** defines the path distribution on the homogeneous space and is implemented in two variants, `GLnPathDistributionEncoder` for $\mathbb{R}^n$ and `SOnPathDistributionEncoder` for $\mathbb{S}^n$. Both are parametrized by the latent dimension of $\mathbf{z}$ and the degree K of the Chebyshev polynomials used to represent the drift. Sample paths are obtained with a geometric Euler–Maruyama solver (Piggott & Solo, 2016).
- The **decoder** is a two-layer MLP with a ReLU activation on the hidden layer, whose dimensionality is dataset dependent.

All parameters are optimized using Adam with cosine annealing learning rate scheduling. The training objective is the ELBO, with individually weighted terms for the reconstruction likelihood, the Kullback-Leibler (KL) divergence of the initial latent state KL_0, and the divergence of the latent path samples KL_p. The likelihood is evaluated under a normal distribution with a fixed standard deviation σ, treated as a hyperparameter.

B.5 Generation of Burst Mask

To simulate data sparsity and the consequent irregular sampling, we generate masks that specify the available data points as follows. For each feature, we sample lengths of consecutive zeros, i.e., of missing observations, in the mask from a geometric distribution with parameter sp and lengths of consecutive ones from a geometric distribution with parameter $s(1-p)$. This corresponds to a discrete Markov chain $(M_t)_t$ with transition probabilities $\mathbb{P}[M_{t+1} = 1 \mid M_t = 0] = sp$ and $\mathbb{P}[M_{t+1} = 0 \mid M_t = 1] = s(1-p)$, which yields a mask with an expected fraction p of available observations. Compared to a uniformly sampled random mask, this results in longer intervals of missing or available information, and therefore a more difficult problem, but also a realistic setting. The lengths of such intervals are further controlled by a parameter $s \in (0, 1]$ (in the experiment we use $s = 0.2$) with smaller s resulting in fewer but longer intervals. The mask is applied to the entire time series, i.e., before dividing it into windows. Two example masks, generated for $p = 0.01$ and $p = 0.05$ with $s = 0.2$, are shown in Figure 3.

B.6 Computing Infrastructure.

All LSD experiments and Neural ODE baseline runs were conducted on a server equipped with an Intel Xeon Silver 4114 CPU (2.20 GHz) and NVIDIA GeForce RTX 2080 Ti GPUs (12 GB VRAM). LSD experiments on QAPPD were run on a server with an Intel Xeon E5-2687W v4 CPU (3.00 GHz) and NVIDIA Titan X (Pascal) GPUs (12 GB VRAM). All baseline experiments were conducted on a server with an Intel Core i9-10940X CPU (3.30 GHz) and NVIDIA GeForce RTX 3090 GPUs (24 GB VRAM).

[7] https://deepod.readthedocs.io/en/latest/
[8] https://github.com/plus-rkwitt/LatentSDEonHS

Table 8: Average training runtimes (in minutes) per run for LSD and all baseline methods. Runtimes are measured on a single GPU where applicable.

Method	SWaT (1)	WaDi (1)	PSM (1)	SMAP (55)	MSL (27)	SMD (28)
COPOD	0.29	0.96	0.05	0.21	0.11	0.35
IForest	0.10	0.29	0.03	0.29	0.15	0.32
KNN	2.74	7.52	0.16	0.18	0.11	0.51
LOF	2.82	7.38	0.16	0.21	0.11	0.50
OCSVM	510.48	2190.78	21.83	0.75	0.33	28.44
PCA	0.07	0.27	0.02	0.13	0.10	0.17
AnomalyTransformer	3.83	4.60	0.96	3.82	0.96	7.76
COUTA	27.67	42.28	7.34	8.16	3.42	35.91
DeepIF	30.11	67.32	5.67	13.81	5.59	42.67
DeepSVDD	1.73	2.72	0.41	1.19	0.55	2.65
TcnED	3.33	3.05	0.60	2.46	0.61	8.67
TimesNet	10.16	11.81	2.50	6.66	1.82	18.54
TranAD	1.21	3.07	0.19	0.74	0.32	1.72
USAD	1.08	5.58	0.15	1.78	1.61	2.07
LSD on $\mathbb{R}^n$	156	116	184	804	411	487
LSD on $\mathbb{S}^n$	196	196	173	1148	371	1066

C Additional Experimental Results

C.1 Experimental Runtimes

Table 8 reports the runtime of a single training run for all baseline methods and our proposed LSD approach, in both $\mathbb{R}^n$ and $\mathbb{S}^n$ variants, across all seven benchmarks. The reported values reflect training time only on a random seed, excluding data loading overhead. For benchmarks comprising multiple traces, runtimes are aggregated over all traces.

The experimental runtimes show that LSD incurs substantially higher training times than all baseline methods. This is expected given the complexity of fitting a latent SDE model and the iterative nature of the numerical solvers required for training. At inference time, the cost is considerably lower, as scoring requires only a forward pass through the recognition network, the SDE solver for a small number of sample paths, and the decoder, without backpropagation or iterative optimization.

The increased cost is thus primarily a training overhead. We consider this trade-off reasonable given the observed performance, particularly in settings with sparse and irregularly sampled data.

C.2 Sparsity Evaluation

This section extends the sparsity evaluation of Section 4.2. Table 9 additionally reports results with $5\,\%$ of the data retained. Moreover, besides feature-wise linear interpolation, we evaluate the baselines with two further strategies for filling missing values: cubic spline interpolation, as used by Sun et al. (2026), and forward filling, which propagates the last valid observation until the next one.

Across both sparsity levels, both LSD variants achieve the best overall ranks. While $\mathbb{R}^n$ leads overall, $\mathbb{S}^n$ is particularly strong at $1\,\%$ data availability.

Regarding the imputation strategies, the simple linear interpolation and forward filling tend to outperform spline interpolation: for both classical and deep learning-based baselines, the spline-imputed variants rank exclusively in the lower half of their respective section of the table. Given the small number of 5 runs, drawing reliable conclusions on the choice of imputation strategy would require a more comprehensive study, which is beyond the scope of this work.

C.3 Ablation Analysis

We assess the contribution of individual components of LSD by comparing both variants, $\mathbb{R}^n$ and $\mathbb{S}^n$, in their default configuration against two ablations:

Table 9: Sparsity evaluation on the QAPPD benchmark with 1 % and 5 % of the data retained. Baselines are evaluated with linear interpolation, spline interpolation, and forward filling of the missing values; LSD operates on the sparse data directly. The best, second-best, and third-best results per metric are highlighted, and ranks are computed only over the results shown in this table.

		QAPPD (1%)			QAPPD (5%)		
Rank ↓	Model	AUC ↑	AUPRC ↑	F1 ↑	AUC ↑	AUPRC ↑	F1 ↑
5.50	COPOD (linear)	55.25 ± 2.25	4.46 ± 0.70	9.95 ± 1.09	62.64 ± 2.56	6.00 ± 1.66	11.81 ± 1.89
8.17	OCSVM (forward fill)	53.25 ± 6.19	5.15 ± 1.37	11.30 ± 2.81	58.20 ± 3.20	5.56 ± 1.29	9.78 ± 1.88
10.00	PCA (forward fill)	52.72 ± 6.44	5.49 ± 2.31	11.89 ± 4.11	56.45 ± 3.21	5.52 ± 1.37	9.45 ± 1.75
10.83	OCSVM (linear)	53.99 ± 3.61	4.02 ± 0.92	9.77 ± 2.26	62.40 ± 3.81	4.80 ± 0.63	9.71 ± 1.31
11.50	IForest (linear)	54.35 ± 2.54	3.83 ± 0.34	9.06 ± 1.01	63.06 ± 4.30	4.63 ± 0.95	9.91 ± 1.71
12.00	COPOD (forward fill)	52.34 ± 4.67	4.43 ± 0.87	9.80 ± 1.97	58.43 ± 2.69	4.71 ± 0.99	9.93 ± 1.22
13.17	IForest (forward fill)	52.96 ± 7.32	4.22 ± 1.19	9.97 ± 2.21	58.27 ± 2.16	4.35 ± 1.00	8.95 ± 1.45
13.83	KNN (forward fill)	53.15 ± 5.63	4.53 ± 0.72	10.82 ± 2.01	51.17 ± 3.11	4.67 ± 1.01	8.73 ± 1.54
14.83	PCA (linear)	52.91 ± 3.63	3.72 ± 0.80	9.13 ± 1.84	59.40 ± 5.19	4.43 ± 0.65	9.46 ± 1.23
16.67	COPOD (spline)	49.25 ± 4.28	3.17 ± 0.61	8.42 ± 0.95	61.25 ± 4.84	4.49 ± 0.16	10.29 ± 0.55
16.83	KNN (linear)	55.17 ± 3.48	3.40 ± 0.42	9.46 ± 1.32	55.45 ± 2.66	3.96 ± 0.48	8.68 ± 0.98
17.00	PCA (spline)	50.46 ± 3.76	3.23 ± 0.42	9.52 ± 1.05	59.14 ± 5.25	3.90 ± 0.43	9.82 ± 1.70
18.00	IForest (spline)	49.01 ± 4.12	3.31 ± 0.75	9.42 ± 1.64	59.50 ± 5.06	4.20 ± 0.67	9.48 ± 1.11
19.17	LOF (forward fill)	53.98 ± 2.59	4.14 ± 0.84	9.91 ± 1.82	51.78 ± 1.51	3.55 ± 0.48	7.89 ± 0.84
22.67	LOF (spline)	49.32 ± 5.24	3.70 ± 1.39	9.92 ± 3.38	52.54 ± 3.73	3.41 ± 0.22	7.50 ± 0.91
23.50	KNN (spline)	49.39 ± 5.71	2.98 ± 0.45	8.72 ± 1.12	52.92 ± 5.15	3.74 ± 0.66	8.68 ± 1.81
24.33	LOF (linear)	51.70 ± 4.62	2.91 ± 0.28	7.95 ± 0.85	53.54 ± 2.33	3.48 ± 0.40	7.97 ± 1.12
25.50	OCSVM (spline)	49.78 ± 0.87	2.68 ± 0.02	5.14 ± 0.01	53.08 ± 5.20	3.81 ± 0.66	7.89 ± 1.05
6.83	USAD (forward fill)	54.26 ± 4.80	5.49 ± 1.09	12.76 ± 2.61	53.61 ± 2.13	6.11 ± 1.09	10.49 ± 1.12
7.17	DeepIF (linear)	52.64 ± 5.08	4.97 ± 1.75	10.29 ± 1.97	58.80 ± 5.46	6.31 ± 1.24	10.67 ± 2.09
8.50	DeepIF (forward fill)	51.75 ± 3.63	4.73 ± 1.05	10.42 ± 3.26	56.64 ± 2.68	6.79 ± 1.75	10.50 ± 2.34
12.83	USAD (linear)	51.46 ± 4.26	4.54 ± 1.38	10.48 ± 1.50	52.39 ± 2.10	5.38 ± 1.21	10.25 ± 2.28
17.00	DeepSVDD (linear)	54.53 ± 5.64	3.42 ± 0.37	9.83 ± 1.38	54.24 ± 4.66	3.88 ± 0.50	8.08 ± 1.33
20.50	USAD (spline)	49.08 ± 4.80	2.85 ± 0.45	8.67 ± 0.83	53.35 ± 7.11	4.21 ± 1.54	10.30 ± 3.54
21.00	DeepSVDD (spline)	52.24 ± 3.07	3.91 ± 1.15	9.53 ± 1.08	51.04 ± 7.02	3.58 ± 0.56	8.43 ± 1.28
22.33	DeepIF (spline)	48.33 ± 5.70	2.79 ± 0.43	8.40 ± 0.92	55.05 ± 5.48	4.10 ± 1.20	9.56 ± 2.36
25.50	DeepSVDD (forward fill)	50.58 ± 8.87	3.14 ± 0.79	8.09 ± 1.38	49.81 ± 1.74	3.82 ± 0.81	7.36 ± 1.02
4.83	LSD on $\mathbb{R}^n$ (ours)	56.46 ± 2.74	6.07 ± 1.24	10.13 ± 1.34	58.18 ± 1.87	6.66 ± 1.09	11.59 ± 1.02
5.00	LSD on $\mathbb{S}^n$ (ours)	58.87 ± 2.48	6.97 ± 0.96	10.74 ± 1.11	57.85 ± 2.85	6.13 ± 0.76	10.35 ± 0.92

- **Strong decoder.** By default, the decoder is deliberately *weak*, so that the temporal structure has to be captured by the encoder and the latent SDE rather than compensated for by the decoder (Chen et al., 2017). The ablation replaces the default decoder, an MLP with a single hidden layer of 12 units on PSM, by an over-parametrized MLP with four hidden layers of 128 units each.
- **Latent Neural ODE.** We replace the latent SDE by a latent Neural ODE (Rubanova et al., 2019), which models the latent dynamics deterministically and parametrizes the vector field by a neural network instead of Chebyshev polynomials, making it a less constrained model.

The latent Neural ODE shares the recognition network and the Gaussian emission decoder of LSD, but replaces the stochastic path on the homogeneous space by a deterministic trajectory in a Euclidean latent space. The encoder parametrizes a variational posterior $q(\mathbf{z}_0 \mid \mathbf{X}_n)$ over the *initial* state only, and the latent trajectory $\mathbf{z}(t)$ is obtained by integrating an autonomous Neural ODE $\dot{\mathbf{z}} = v_\theta(\mathbf{z})$ over the reference time grid with an adaptive Dormand–Prince solver. The vector field v_θ is a small MLP with three linear layers and hidden dimension 64; all other hyperparameters, such as the latent dimension, are shared with LSD. Accordingly, its ELBO retains the reconstruction and initial-state KL terms but has no path-KL term. As encoder, decoder, anomaly score, and evaluation protocol are identical to LSD, the ablation measures the joint effect of the stochastic path measure, the geometric latent structure, and the constrained parametrization of the dynamics. Results are reported over 3 runs.

The ablation results, shown in Table 10, supports the design choice of a weak decoder: with a stronger decoder network, performance falls short of the proposed configuration. In line with Chen et al. (2017), we attribute this to the more expressive decoder being able to model the training data without relying on the latent representation. The deliberately restricted decoder instead forces the encoder

Table 10: Ablation study on PSM, comparing LSD with a variant using a stronger decoder and with a deterministic latent Neural ODE replacing the latent SDE. Results are reported as mean ± standard deviation over 3 runs. The best and second-best results per metric are highlighted.

		PSM		
Rank ↓	**Model**	**AUC** ↑	**AUPRC** ↑	**F1** ↑
2.67	LSD on $\mathbb{R}^n$ (ours), dec. 2×12	76.22 ± 2.32	47.94 ± 1.39	58.56 ± 3.57
1.00	LSD on $\mathbb{S}^n$ (ours), dec. 2×12	77.75 ± 1.14	50.33 ± 3.37	61.52 ± 1.46
4.00	LSD on $\mathbb{R}^n$ (ours), dec. 4×128	73.52 ± 5.14	48.61 ± 1.79	56.80 ± 3.59
3.00	LSD on $\mathbb{S}^n$ (ours), dec. 4×128	75.24 ± 1.41	47.04 ± 0.94	58.88 ± 2.06
4.33	NeuralODE	74.69 ± 1.13	45.32 ± 0.42	56.90 ± 1.69

and the latent SDE to capture the underlying dynamics, which benefits the downstream detection task.

Both LSD variants also outperform the latent Neural ODE. This indicates that the combination of the stochastic path measure, the geometric latent structure, and the constrained parametrization of the dynamics benefits the downstream detection task, although this ablation does not attribute the gain to any single one of these components.

D Societal Impact

By enabling anomaly detection on noisy, sparse, and irregular multivariate time series, LSD broadens the scope of deployable monitoring systems for industrial applications, where early fault detection directly improves safety, reliability, and sustainability. In line with the EU AI Act, we advocate deploying LSD as a decision-support tool, preserving operator expertise and maintaining human oversight. Adversarial robustness remains an open challenge and a promising direction for future work.