State-Derivative-Aware Neural Controlled Differential Equations for Multivariate Time Series Anomaly Detection and Diagnosis

Xin Sun[1], **Heng Zhou**[1], **Yuhao Wu**[1], **Chao Li**[12*]

[1]International Business School, Zhejiang University, Hangzhou, China
[2]National Engineering Laboratory for Industrial Control System Security Technology, Zhejiang University, Hangzhou, China
xinsun1@zju.edu.cn, hengzhou@zju.edu.cn, wyh18811306637@zju.edu.cn, chaoli@zju.edu.cn

Abstract

Multivariate time series anomaly detection is a crucial factor in real-world applications but a challenging task due to the complex temporal dependencies and system dynamics. Reconstruction-based methods have made great improvements in recent years. However, we observe an issue these methods are suffering, that they primarily measure deviations in the time points themselves when performing anomaly detection but ignore changes in the dynamic properties of the system. In these cases, they are unable to produce sufficient reconstruction errors to detect anomalies, so some potential abnormal time points caused by the dynamic evolution of the system are missing. To address this problem, we propose a novel method, SDA^{2}D, which models system dynamics by the derivative of the NCDE-derived state vector with respect to time, enabling the learning of reconstruction deviation and system evolution jointly. Our experimental results show that SDA^{2}D achieves noticeable improvements in four benchmark datasets, and the visualization also provides further instructions for anomaly diagnosis, which helps locate the sources of these anomalies.

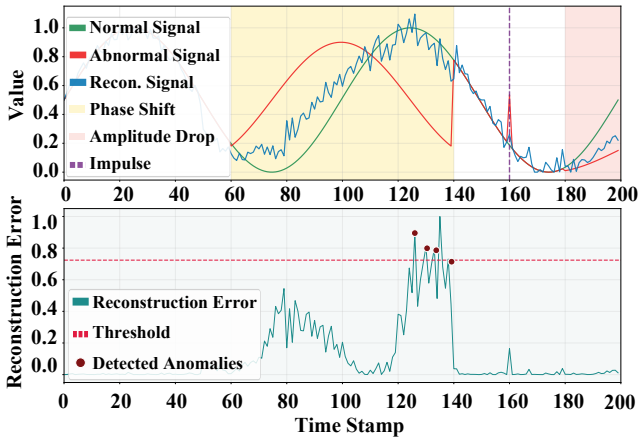


Figure 1: Challenges for reconstruction-based models. They mainly focus on the deviations from data, while generating insufficient anomaly scores for detecting anomaly events.

1 Introduction

The development of Internet of Things (IoT) devices and advanced monitoring systems has driven an exponential growth of time series data in various domains (Almeida et al. 2023; Zhang et al. 2025), from industrial automation (Kamm et al. 2023), energy systems (Huang et al. 2024; Zhou et al. 2025) to healthcare (Di Martino and Delmastro 2023) and smart infrastructure (Farahani et al. 2023). In such data-rich environments, effective anomaly detection, which aims to identify observations that significantly deviate from expected temporal patterns, plays a crucial role in preventing system failures, mitigating operational risks, and enabling proactive decision-making in real-world applications (Zamanzadeh Darban et al. 2024).

In recent years, reconstruction-based methods have become competitive candidates for **M**ultivariate **T**ime **S**eries **A**nomaly **D**etection (MTS AD) (Sun, Zhou, and Li 2025). A commonly used principle is that all samples for training are normal, so larger reconstruction errors can be generated for abnormal samples during inference. These reconstruction-based models include specially designed AD models such as Darban et al. (2025); Xu et al. (2022), together with time series foundation models such as FITS (Xu, Zeng, and Xu 2024) and TimesNet (Wu et al. 2023). Following the findings of the latest study (Liu and Paparrizos 2024), the standard principle can be concluded as follows. First, we assume that all time points in the training dataset are normal, and these models learn to rebuild them as accurately as possible; then, during inference with the testing dataset, these models would generate larger reconstruction errors for abnormal time points than normal ones because they do not learn how to reconstruct abnormal data. Acquiring the reconstruction errors from the testing data for each time point, we normalize them and apply the statistical principle $\mu + 3\sigma$ to determine the time points with larger errors as anomalies.

Although significant improvements have been made, we have observed a limitation of these methods. Reconstruction errors primarily measure deviations in the time points themselves, but may ignore changes in the dynamic properties of the system. For example, if the current state of a given system is abnormal but the amplitudes of the time points are still in the normal range, e.g., a phase shift in

*The Corresponding Author.

periodic fluctuations, reconstruction errors may not be detected at these time points. These are widely present in MTS analysis, where conventional reconstruction metrics fail to capture anomalies caused by dynamic changes in a system, such as phase drifts in sensor signals, frequency deviations in mechanical vibrations, or morphological distortions in biological signals. To make this more concrete, we use Fig.1 to show these cases. We can notice that when an abnormal phase shift with valid amplitudes occurs, although the reconstructed time series can reflect the normal data pattern to some extent and generate larger errors, these errors are not enough to support that they are anomalies, as shown in Fig.1. They focus on time-point reconstruction, but overlook the fact that abnormal events such as phase shifts and impulses are more severe dynamic anomalies than errors themselves.

Recently, **N**eural **C**ontrolled **D**ifferential **E**quations (NCDEs) have been proposed to effectively model time series (Kidger et al. 2020). By integrating the dynamic modeling capabilities of differential equations with neural networks, NCDEs effectively capture complex temporal dependencies. This motivates us to explore their potential in addressing the limitations of existing methods in detecting anomalies caused by dynamic state evolution, a capability that remains to be explored. Specifically, once an informative feature vector for a given time window is derived from NCDEs, we can compute its time derivative to quantify the state evolution of this system. Although this derivative is more semantically implicit and high-dimensional, this aligns with the math definition that the derivative of a function $f(x)$, which is denoted $f'(x)$ or $\frac{df(x)}{dx}$, characterizes the instantaneous rate of change. These models should also learn how to reconstruct the derivative to evaluate the dynamic state of the system. This property naturally helps us to model the dynamics of the system effectively.

To further address the limitation in existing methods for reconstruction, we propose SDA^{2}D, the **S**tate-**D**erivative-**A**ware Neural Controlled Differential Equations for multivariate time series **A**nomaly **D**etection, a novel method leveraging the time derivative from NCDEs to enhance the ability to capture system state evolution, instead of focusing solely on reconstruction errors at individual time points. SDA^{2}D mainly contains four key components: (1) NCDE Solver: An NCDE solver is integrated to model the continuous-time dynamics of sequential data, enabling adaptive learning of temporal dependencies and latent state transitions; (2) Temporal-Spatial Interaction Module (TSIM): We develop a strategy to capture the inherent temporal dependencies and feature correlations in an interactive process; (3) System State Derivative Aware Module (SSDAM): Based on the state vector of the system, the time derivative can be calculated to describe the state evolution according to the NCDE solution mechanism and the chain derivation rule; and (4) Joint Reconstruction Module (JRM): SDA^{2}D learns jointly to rebuild the time series and the system state derivative for different datasets.

Our main contributions are listed as follows: (1) We observe the limitation on reconstruction-based methods caused by dynamic changes of system states, showing that insufficient support for anomaly detection relies only on reconstruction errors; (2) We propose a novel method, SDA^{2}D, to model errors at time points and differences from the evolution of the system state; (3) Our experimental results show the effectiveness of SDA^{2}D in four real-world datasets. We also perform anomaly diagnosis and visualization to demonstrate the potential and explainability of SDA^{2}D.

2 Related Work

Multivariate Time Series Anomaly Detection

Modern methods integrate deep learning through density estimation, such as MPPCACAD (Yairi et al. 2017) and DAGMM (Zong et al. 2018), or clustering mechanism such as Fuzzy C-Means (Li et al. 2021). Meanwhile, contrastive learning models, including CARLA (Darban et al. 2025), TFMAE (Fang et al. 2024), DCdetector (Yang et al. 2023), TS-TCC (Eldele et al. 2023), and CoST (Woo et al. 2022), further improve representation learning for multivariate time series anomaly detection tasks.

The development of reconstruction-based methods advanced to sophisticated models like OmniAnomaly (Su et al. 2019), MTAD-GAT (Zhao et al. 2020), Anomaly Transformer (Xu et al. 2022), CATCH (Wu et al. 2024) and SARAD (Dai et al. 2024). In parallel, time series foundation models including OFA (Zhou et al. 2023), FITS (Xu, Zeng, and Xu 2023), and TimesNet (Wu et al. 2023) have demonstrated strong anomaly detection capabilities via reconstruction. The recent DADA (Shentu et al. 2024) further extends this area as a generalizable detector.

Despite noticeable advances, we find that reconstruction-based methods mainly focus on point-wise deviations, potentially ignoring changes in dynamic properties of the system. We address this by jointly learning reconstruction biases and system evolution.

Neural Differential Equations

Neural Differential Equations (NDEs) provide a theoretical framework for modeling continuous-time dynamical systems with the expressive function approximation of deep neural networks (Kidger 2022). They exhibit distinct advantages in modeling complex continuous dynamical processes, particularly by overcoming the resolution constraints and discontinuity artifacts inherent in conventional discrete-time methods (Dupont, Doucet, and Teh 2019; Kidger et al. 2020).

One of the most common NDEs is a NODE, proposed in (Chen et al. 2018), which can be formulated as:

$$\mathbf{z}(t_T) = \mathbf{z}(t_0) + \int_{t_0}^{t_T} f(\mathbf{z}(t), t; \theta_f) dt, \tag{1}$$

where f_θ is the ODE function implemented by a neural network to approximate the continuous dynamics of the state vectors of the system, that is, the time derivative $\frac{d\mathbf{z}(t)}{dt}$.

From Eq.1, it can be observed that one limitation of NODEs is that the solution of an ODE is determined by the initial condition at $\mathbf{z}(t_0)$, which provides no inherent mechanism to integrate observed data from subsequent time steps, ultimately constraining their representation learning

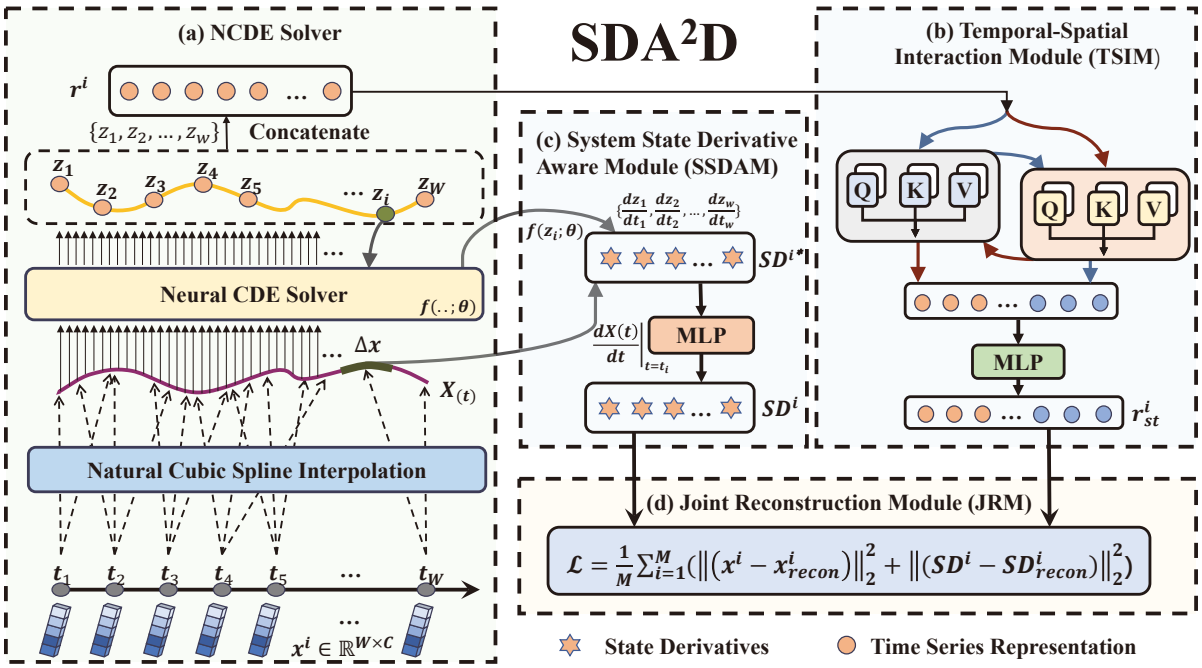


Figure 2: The overall architecture of SDA^{2}D. It consists of four key compotents: (1) NCDE Solver for getting the status vector for a given time window x^i; (2) TSIM module for building temporal-spatial features; (3) SSDAM module for calculating dynamic derivation to describe system evolution; and (4) JRM module for joint learning with enhanced masking strategy.

capabilities. To overcome these, NCDEs that utilize the Riemann-stieltjes integral are proposed in (Kidger et al. 2020), which can be formulated as:

$$\begin{aligned}\mathbf{z}(t_T) &= \mathbf{z}(t_0) + \int_{t_0}^{t_T} f(\mathbf{z}(t);\theta_f)dX(t), \\ &= \mathbf{z}(t_0) + \int_{t_0}^{t_T} f(\mathbf{z}(t);\theta_f)\frac{dX(t)}{dt}dt,\end{aligned} \tag{2}$$

where f_θ is the CDE function and $X(t)$ is a continuous time series path interpolated from the input time series data. Under the effect of $X(t)$, NCDEs can learn continuous-time dynamics with explicit dependence on external signals.

Meanwhile, although NCDEs have been selected for certain time series tasks, such as representation learning (Liu et al. 2024), imputation (Wi, Shin, and Park 2024), and forecasting (Jhin et al. 2024; Li, Zhu et al. 2023), it also remains to be explored how to perform MTS AD and improve the ability of models to capture the state evolution of a given system under the mechanism of NCDEs.

3 Method

Problem Settings

We define an MTS dataset as $\mathcal{D} = \{x_1, x_2, x_3, ..., x_N\}$ and each time point $x_i \in \mathbb{R}^C$, where N represents the total number of time instances and C represents the number of variables in MTS. In practice, we define a window size W, selecting every W time points as a time interval, that is, $\mathcal{X} = \{x^1, x^2, x^3, ..., x^M\}$, where each $x^i \in \mathbb{R}^{W\times C}$ for $i = 1$ to M, with M the number of time intervals. Then we send every time interval x^i individually to SDA^{2}D for anomaly detection. This can be viewed as a classification task that determines whether each time point is normal or abnormal. The architecture of SDA^{2}D is shown in Fig.2.

NCDE Solver

Here, we select an NCDE solver (Kidger et al. 2020) as our time series encoder. Functioning as a continuous-time counterpart to RNNs (Yu et al. 2019), NCDEs inherently model the temporal evolution of time series through hidden states of the continuously evolving system. This contrasts with Transformer-based approaches (Vaswani et al. 2017) that discretize temporal values into tokenized sequences, a paradigm originally designed for NLP that conflicts with the continuous nature of real-world temporal processes. The reasons for the selection of an NCDE solver as our encoder can be concluded as two key advantages. First, its continuous dynamics formulation provides theoretical guarantees for capturing dynamic changes in a system, which is critical to enhancing the sensitivity of AD models to capture state evolution. Second, its mathematical foundation robustly handles irregularly sampled or incomplete time series without requiring artificial value imputation, such as zero-filling (Yue et al. 2022) or linear interpolation (Zerveas et al. 2021), which could distort data distributions.

Concretely, first, we employ the nature cubic spline to interpolate discrete time series $\mathcal{D}$ into a continuous path $X \in \mathbb{R}^{N\times C}$ with each $X(t) \in \mathbb{R}^C$, which is the preprocessing of the NCDEs. Second, we split the whole time series into different time intervals. For every x^i, we use an

initialization layer with parameter θ_1 and average operation to transform it from time domain into hidden state domain and get $\mathbf{z}(t_0)$ for the following solver steps, which can be denoted as:

$$\mathbf{z}(t_0)^* = \Phi_{\theta_1}(x^i),\ \mathbf{z}(t_0)^* \in \mathbb{R}^{W\times dim}, \tag{3}$$

$$\mathbf{z}(t_0) = \frac{1}{W}\sum_{i=1}^{W}\mathbf{z}(t_0)^*_{i,:},\ \mathbf{z}(t_0) \in \mathbb{R}^{dim}, \tag{4}$$

where dim is the dimension of latent space in the NCDE solver. For the time interval x^i with time stamps range in $[t_1, t_W]$, we can get the system state vector by:

$$\mathbf{z}(t_i) = \mathbf{z}(t_0) + \int_{t_0}^{t_i} f(\mathbf{z}(t);\theta_2)\frac{dX(t)}{dt}dt,\ i \in [1, W], \tag{5}$$

where θ_2 is the learnable parameters defined in the NCDE solver. The solutions generated by the NCDE solver exhibit continuous temporal dynamics, reflecting the gradual transformation of the system. Consequently, we concatenate all representations at each time observation to form the system state vector for x^i by:

$$r^i = \text{Concat}(\mathbf{z}(t_1), \mathbf{z}(t_2), \mathbf{z}(t_3), ..., \mathbf{z}(t_W)),\ r^i \in \mathbb{R}^{W\times dim}. \tag{6}$$

Temporal-Spatial Interaction Module (TSIM)

Capturing both temporal dynamics and spatial correlations in multivariate time series data is critical to modeling complex systems. Temporal patterns reveal evolving trends and latent state transitions, while spatial correlations encode cross-variable interactions and structural dependencies. Unlike existing methods that tend to build temporal and spatial features in separate branches (Deng et al. 2021; Li, Feng, and Belgiu 2024; Shao et al. 2022), we develop a time-spatial interaction strategy to generate more informative temporal and spatial features.

To achieve these, we incorporate the attention mechanism to extract these features. Here, we select $\mathbf{t}$ and $\mathbf{s}$ to represent the temporal and spatial branches. In TSIM, there are two submodules, $h_{\mathbf{t}}$ and $h_{\mathbf{s}}$, for the extraction of temporal features and spatial correlations, respectively. Meanwhile, to control the number of trainable parameters of SDA^{2}D, we build each of them with a single multihead self-attention encoder layer. We define the number of attention heads to be $N_{h_{\mathbf{t}}}$ and $N_{h_{\mathbf{s}}}$, respectively. For one of these two submodules, the calculation process can be described as:

$$h_{\mathbf{p}}(r^i) = \text{Concat}(\text{head}_1, \ldots, \text{head}_{N_{h_{\mathbf{p}}}})\mathbf{W}^O,\ \mathbf{p} \in \{\mathbf{t}, \mathbf{s}\}, \tag{7}$$

$$\text{head}_j = \text{Attention}(r^i\mathbf{W}_j^Q, r^i\mathbf{W}_j^K, r^i\mathbf{W}_j^V),\ j \in [1, N_{h_{\mathbf{P}}}], \tag{8}$$

$$\text{Attention}(\mathbf{Q}_j, \mathbf{K}_j, \mathbf{V}_j) = \text{Softmax}\left(\frac{\mathbf{Q}_j\mathbf{K}_j^\top}{\sqrt{d_k}}\right)\mathbf{V}_j, \tag{9}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ and $\mathbf{W}^O$ are learnable parameter matrices, and d_k is the projection dimension of $\mathbf{K}$ in each attention head. To better fusion the relationships between temporal and spatial, we select a interaction strategy as:

$$r^i_{\mathbf{s}\circ\mathbf{t}} = h_{\mathbf{s}}(h_{\mathbf{t}}(r^i)), r^i_{\mathbf{t}\circ\mathbf{s}} = h_{\mathbf{t}}(h_{\mathbf{s}}(r^i)), \tag{10}$$

where $r^i_{\mathbf{s}\circ\mathbf{t}}, r^i_{\mathbf{t}\circ\mathbf{s}} \in \mathbb{R}^{W\times dim}$. In this case, the extraction of both features is guided and interacted with by the other part, aiming to further enhance the information of the features. Then we combine $r^i_{\mathbf{s}\circ\mathbf{t}}$ and $r^i_{\mathbf{t}\circ\mathbf{s}}$ to get the temporal-spatial feature $r^i_{\mathbf{st}}$ by:

$$r^i_{\mathbf{st}} = \text{Concat}(r^i_{\mathbf{s}\circ\mathbf{t}}, r^i_{\mathbf{t}\circ\mathbf{s}}),\ r^i_{\mathbf{st}} \in \mathbb{R}^{W\times(2\times dim)}, \tag{11}$$

which is used for time series reconstruction in the following part to evaluate biases on a time-point level.

System-State Derivative-Aware Module

As we have discussed in Sect.1 and shown in Fig.1, these existing reconstruction-based methods mainly focus on reconstruction errors between the input time series and the rebuild time series, while may fail to capture the anomalies caused by the state evolution, leading to suboptimal performance when conducting AD. It inspires us that it is not sufficient enough to support these dynamic anomalies solely on the basis of reconstruction errors from time series. NCDEs are powerful tools for modeling complex systems, and we use the time derivative, which aligns with the effect in math, to better capture dynamic properties in the system.

Global Cubic Spline Coefficient Computation When we interpolate discrete time series $\mathcal{D}$ into a continuous path $X \in \mathbb{R}^{N\times C}$, as described in Sect.3, we select the nature cubic spline as the basic method. After interpolation, the coefficients of the cubic spline interpolation are first computed globally throughout the time range $[t_1, t_N]$. For every two adjacent time stamps t_k and t_{k+1}, the spline can be described as the following for $t \in [t_k, t_{k+1})$:

$$X(t) = \mathbf{a}_k + \mathbf{b}_k(t-t_k) + \mathbf{c}_k(t-t_k)^2 + \mathbf{d}_k(t-t_k)^3, \tag{12}$$

where $\mathbf{a}_k$, $\mathbf{b}_k$, $\mathbf{c}_k$, $\mathbf{d}_k \in \mathbb{R}^C$ are coefficients for the k-th spline ranging in $[t_k, t_{k+1})$, ensuring continuity of $X(t)$, $X'(t)$ and $X''(t)$. These coefficients are derived by solving a system of linear equations based on data points and boundary conditions.

Window Coefficient Extraction After obtaining the global coefficients $\Psi = \{\psi_k\}_{k=1}^{N-1} = \{\mathbf{a}_k, \mathbf{b}_k, \mathbf{c}_k, \mathbf{d}_k\}_{k=1}^{N-1}$, a local time window of size W is split. For this window x^i with time stamps ranging in $[t_1, t_W]$, the relevant coefficients correspond to the intervals within $[t_1, t_W]$ can be represented as $\text{Coeffs}^i \in \mathbb{R}^{(W-1)\times 4\times C}$:

$$\text{Coeffs}^i = \{\psi_{t_1}, \psi_{t_2}, \psi_{t_3}, ..., \psi_{t_{W-1}}\}, \tag{13}$$

and it is mathematically stipulated that the last group parameters in Coeffs^i, i.e., $\psi_{t_{W-1}}$, is also for calculating with the right endpoint of the interval $[t_1, t_W]$, that is t_W.

Spline Derivative Calculation Given the spline $X(t)$ and the time stamps $[t_1, t_W]$ for x^i, the spline derivative $\frac{dX(t)}{dt}$ can be calculated piecewise. Concretely, for a given time stamp t in $[t_1, t_W]$ and t is located in the spline $[t_k, t_{k+1})$ with $1 \le k \le W-1$, its corresponding spline derivative can be acquired by:

$$\frac{dX(t)}{dt} = \mathbf{b}_k + 2\mathbf{c}_k(t-t_k) + 3\mathbf{d}_k(t-t_k)^2, \tag{14}$$

where $\mathbf{a}_k$, $\mathbf{b}_k$, $\mathbf{c}_k$, and $\mathbf{d}_k$ are parameters in ψ_k. Here, we can extract the values of the spline derivative of integer moments in $[t_1, t_W]$ and then get $\frac{dX(t)}{dt} \in \mathbb{R}^{W\times C}$, which will be used to calculate the final state derivative to describe system evolution in terms of the chain rule.

System State Derivative Calculation As shown in Eq.5 and Eq.6, r^i contains meaningful information about the state of the system. To further describe the state evolution that occurred in the time interval x^i and enhance the ability of SDA^{2}D to capture the system dynamics, we calculate the state derivative of the system in the following steps. Specifically, for a calculated $\mathbf{z}(t_i)$ by Eq.5, we can compute the final state derivative of current time stamp t_i as:

$$\begin{aligned}\frac{d\mathbf{z}(t_i)}{dt_i} &= \underbrace{\frac{d\mathbf{z}(t_0)}{dt_i}}_{\mathbf{z}(\mathbf{t_0}) \text{ is a constant}} + \underbrace{\frac{d(\int_{t_0}^{t_i} f(\mathbf{z}(t);\theta_2)\frac{dX(t)}{dt}dt)}{dt_i}}_{\text{Leibniz's Rule and Chain Rule}}, \\ &= f(\mathbf{z}(t_i);\theta_2)\frac{dX(t)}{dt}\Big|_{t=t_i},\end{aligned} \tag{15}$$

which means that we can acquire the state derivative of time stamp t_i by feeding it back to $f(\cdot)$, and multy it with the corresponding spline derivative. We concatenate state derivatives from t_1 to t_W to described the dynamic evolution of x^i, denoted as $\text{SD}^{i^*} \in \mathbb{R}^{W\times dim}$:

$$\text{SD}^{i^*} = \text{Concat}(\frac{d\mathbf{z}(t_1)}{dt_1}, \frac{d\mathbf{z}(t_2)}{dt_2}, \frac{d\mathbf{z}(t_3)}{dt_3}, ..., \frac{d\mathbf{z}(t_W)}{dt_W}). \tag{16}$$

Then, we employ an MLP-based structure Φ with parameters θ_3 to map the state derivative from hidden state space into time domain:

$$\text{SD}^i = \Phi_{\theta_3}(\text{SD}^{i^*}),\ \text{SD}^i \in \mathbb{R}^{W\times C}. \tag{17}$$

Joint Reconstruction Module (JRM)

Now, we have obtained the temporal-spatial feature $r^i_{\mathbf{st}}$ and the system state derivative SD^i for the time interval x^i. The principle we addressed here can be concluded that SDA^{2}D should not only learn how to reconstruct time points in normal data patterns, but also be aware of the dynamic evolution of the system state. Given these, first, we introduce two submodules with parameters θ_4 and θ_5 to learn the normal data patterns and the evolution of the system in JRM, which can be denoted as Φ_{θ_4} and Φ_{θ_5}, respectively. These operations can be shown as:

$$\text{SD}^i_{\text{recon}} = \Phi_{\theta_4}(\text{SD}^i),\ x^i_{\text{recon}} = \Phi_{\theta_5}(r^i_{\mathbf{st}}), \tag{18}$$

then we can design the loss function to learn both of them jointly:

$$\mathcal{L} = \frac{1}{M}\sum_{i=1}^{M}(\|x^i - x^i_{\text{recon}}\|_2^2 + \|\text{SD}^i - \text{SD}^i_{\text{recon}}\|_2^2). \tag{19}$$

4 Experiment

Experimental Setting

Dataset In our experiments, we select four redesigned real-world datasets to train and evaluate SDA^{2}D: MSL and SMAP from (Hundman et al. 2018), SWaT from (Goh et al. 2017) and TAO from (Liu and Paparrizos 2024).

Baseline We have selected 15 models with different structures as our baselines in our experiments: CATCH (Wu et al. 2024), DADA (Shentu et al. 2024), FITS (Xu, Zeng, and Xu 2023), TFMAE (Fang et al. 2024), TimesNet (Wu et al. 2023), OFA (Zhou et al. 2023), DCdetector (Yang et al. 2023), AnomalyTransformer (Xu et al. 2022), CoST (Woo et al. 2022), CAE-M (Zhang et al. 2021), TS-TCC (Eldele et al. 2023), USAD (Audibert et al. 2020), OmniAnomaly (Su et al. 2019), MSCRED (Zhang et al. 2019) and DAGMM (Zong et al. 2018).

Implementation Detail The latest benchmark study (Liu and Paparrizos 2024) has redesigned well-organized datasets and searched for optimal hyperparameters in the optimizer, learning rate, and weights of existing loss functions for most of the baselines included in this study. For some of our selected base models, which are not temporarily imported, we set their hyperparameters in their original papers or repositories as optimal ones. Then these models and our proposed SDA^{2}D are also integrated into this proposed pipeline to run in a universal data flow. We program our codes with Python 3.8.13, PyTorch 1.13.0, CUDA 11.7 and Ubuntu 18.04 on a single NVIDIA RTX 3090 24GB GPU. We set the learning rate and the weight decay of the optimizer to 1e-4 and 1e-5, respectively. In addition, a scheduler for the adjustment of the learning rate is included with the multiplicative factor of the learning rate decay defined as 0.5. Our codes are available at https://github.com/ProEcho1/SDA2D.

Evaluation Measure It has been highlighted that traditionally used MTS AD metrics such as F1, AUC-PR, AUC-ROC and Affiliation-F1 could show potential evaluation issues, while in comparison, VUS-PR emerges as the most robust, accurate and fair evaluation measure (Paparrizos et al. 2022; Liu and Paparrizos 2024; Boniol et al. 2025). Given these, VUS-PR is selected as our key metric. For further detailed information, we also record the results in VUS-ROC as an auxiliary metric.

Following the work pipeline in (Liu and Paparrizos 2024), we use the joint reconstruction errors for MTS AD. Concretely, given the original time series $X = \{x_1, x_2, x_3, ..., x_n\}$ and the reconstructed $\hat{X}=\{\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_n\}$, the ***Joint Reconstruction Errors*** s_t can be calculated with the following strategy:

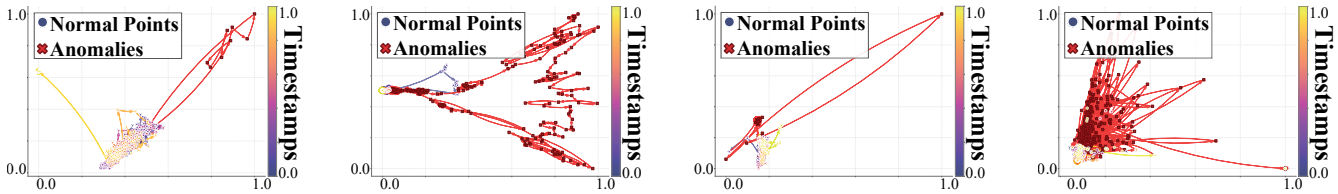
$$s_t = \|x_t - \hat{x_t}\|_2^2 + \|\text{SD}_t - \hat{\text{SD}}_t\|_2^2,\ t = 1, 2, \ldots, n, \tag{20}$$

where $\hat{x_t}$ and $\hat{\text{SD}}_t$ denote the reconstructed time point and system state derivative at this point, respectively. The joint reconstruction errors can be viewed as the anomaly scores for each point because a higher error means that this point is more likely to be abnormal. Then, the anomaly scores for each time point can be represented as s_t^{norm} with normalized s_t using MinMaxScaler. Finally, anomalies are detected using the threshold $\delta = \mu + 3\sigma$, where μ and σ are the mean and standard deviation of s_t^{norm}, which can be shown as:

$$\text{Label}(s_t^{\text{norm}}) = \begin{cases} 1, & \text{if } s_t^{\text{norm}} > \delta \quad \text{(Abnormal)}, \\ 0, & \text{if } s_t^{\text{norm}} \le \delta \quad \text{(Normal)}. \end{cases} \tag{21}$$

Model	Venue	Dataset (The best results and second results are marked with **bold fonts** and underlines, respectively)							
		MSL		SMAP		SWaT		TAO	
		VUS-PR†	VUS-ROC*	VUS-PR†	VUS-ROC*	VUS-PR†	VUS-ROC*	VUS-PR†	VUS-ROC*
CATCH	ICLR, 2025	0.0331	0.7806	0.2882	0.5912	0.1472	0.4235	0.0636	0.4573
DADA	ICLR, 2025	0.0093	0.5049	0.2619	0.6933	0.1201	0.4211	0.0579	0.4306
FITS	ICLR, 2024	0.0087	0.5105	0.2704	0.7845	0.1448	0.4782	0.0784	0.5456
TFMAE	ICDE, 2024	0.0078	0.4089	0.0890	0.2044	0.1233	0.4620	0.0492	0.2476
TimesNet	ICLR, 2023	0.0072	0.4316	0.2678	0.7194	0.1302	0.4121	0.0732	0.4339
OFA	NeurIPS, 2023	0.0083	0.5077	0.2929	0.7897	0.1221	0.4157	0.0570	0.3706
DCDetector	KDD, 2023	0.0096	0.5130	0.1508	0.5208	0.1280	0.4817	0.0491	0.4162
A.T.	ICLR, 2022	0.0063	0.5068	0.2397	0.6326	0.1413	0.4925	0.0513	0.4920
CoST	ICLR, 2022	0.0071	0.4822	0.1477	0.5154	0.1369	0.4998	0.0498	0.3402
CAE-M	TKDE, 2021	0.0043	0.2067	0.0736	0.0203	0.1804	0.6096	0.0784	0.5435
TS-TCC	IJCAI, 2021	0.0088	0.5255	0.1491	0.5214	0.1574	0.5339	0.0484	0.3236
USAD	KDD, 2020	0.0072	0.2702	0.0741	0.0388	0.1768	0.6011	0.0647	0.4641
OmniAnomaly	KDD, 2019	0.0052	0.5025	0.0780	**0.9111**	0.1768	0.6012	0.0647	0.4640
MSCRED	AAAI, 2019	0.0111	0.6883	0.0958	0.3914	0.1804	0.6096	0.0784	0.5434
DAGMM	ICLR, 2018	0.0042	0.2630	0.0748	0.0597	0.1804	0.6095	0.0783	0.5432
SDA2D (Ours)		**0.2829**	**0.9643**	**0.4981**	0.8828	**0.2095**	**0.6395**	**0.1232**	**0.6228**

Table 1: Results of SDA^{2}D and baseline models. All experiments are performed under five random seeds, and the mean values are reported. † denotes as the core metric, $*$ means the auxiliary metric.



(a) MSL (b) SMAP (c) SWaT (d) TAO

Figure 3: The visualization of the dynamic evolution of different systems. In the four subfigures, the red points represent for abnormal time points, and other points represent for normal time points marked with gradient color. It can be found that abnromal time points and normal time points tend to distribute in different subspaces.

Result

Comparison with Different AD Models As indicated in Tab.1, SDA^{2}D has made noticeable improvements on four real-world datasets. Concretely, for the core metric VUS-PR, SDA^{2}D achieves the best results during baseline models on datasets MSL (0.2829), SMAP (0.4981), SWaT (0.2095) and TAO (0.1232). The MSL dataset exhibits complex dynamic evolutions due to normal changes in spacecraft working conditions. As we noted in Sect.1 and Fig.1, such data shifts can hinder reconstruction-based models by generating insufficient reconstruction errors. In contrast, SDA^{2}D learns the state derivative of this system, enabling it to better track these dynamic changes and perform more robust anomaly detection. In addition, as for our selected auxiliary metric VUS-ROC, SDA^{2}D also achieves the best on three real-world datasets, MSL (0.9643), SWaT (0.6395), and TAO (0.6228), with the second results on dataset SMAP (0.8828).

Dynamic Evolution of System To further demonstrate the effectiveness and explainability of SDA^{2}D, for each dataset, we have selected time-continuous subsets of time points of the same length. We guarantee that each subset contains both normal time points and abnormal time points. After getting these subsets from different datasets, we pass them through SDA^{2}D to obtain the high-dimension representations for each time point. Then, PCA is employed to map these representations into a two-dimensional space. For each subset consisting of time points from the same dataset, we connect the time points in chronological order using straight lines with arrows. The results of visualization can be found in Fig.3. The red points represent abnormal time points, and the other points represent normal time points marked with gradient color. It can be observed that normal time points

Experiment	$\overline{\text{VUS-PR}}$
Ours	**0.2784**
Train with only time feature	0.2633
Train with only spatial feature	0.2611
Train without time-spatial interaction	0.2707
AD with only time errors	0.2532
AD with only derivative errors	0.2609

Table 2: Results of ablation study with different AD strategies on four selected datasets.

and abnormal time points are located in different subspaces, as shown from Fig.3(a) to Fig.3(d). These show that SDA^{2}D learns meaningful information to describe the evolution of the system.

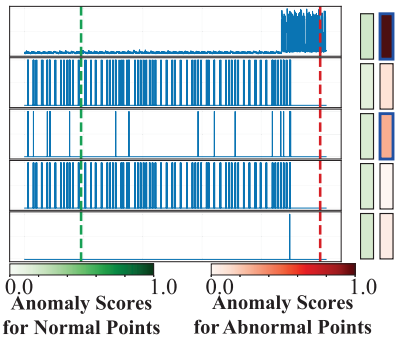
Time Series Anomaly Diagnosis Our method can also be used for anomaly diagnosis. We calculate channel-wise anomaly scores using the normalized Mean Squared Error (MSE) between the input x_i and its reconstruction $\hat{x_i}$. This allows us to identify which specific channels are responsible for a detected anomaly, as demonstrated for the SMAP and SWaT datasets in Fig.4.

First, in Fig.4(a), we can observe that an abnormal event occurred in the first channel due to continuous abnormal mutations and jitters. This aligns with the second column on the right for the anomaly scores of this abnormal time point, that is, the first rectangle on the second column displays a larger reconstruction bias compared to the other channels. Meanwhile, we can also notice that the anomaly score of the third channel is also higher than that of other channels. We attribute this phenomenon to the reason that the model determines that there should be a signal fluctuation here according to the current state of the system, but the actual signal does not observe this fluctuation, thus generating a larger reconstruction error. These provide assistance in finding the devices or sensors that are generating the anomaly in order to fix them. Similarly, in Fig.4(b), the second channel shows an unusual data pattern and a period shift in the data at the selected abnormal point, while at the same time the fifth channel also shows a high signal in an unconventional cycle pattern. Given these, it can be observed that the anomaly scores in the second column also satisfy the analysis, where the second and fifth rectangles show deeper than others.

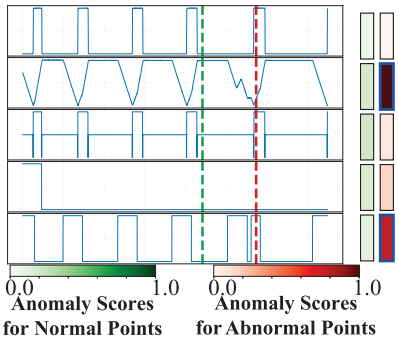
In addition, when paying attention to the selected normal time points, we have found that, compared to abnormal time points, the anomaly scores of each channel do not differ much from each other and the overall values remain at a relatively lower level. This further indicates that SDA^{2}D can benefit to perform anomaly diagnoise, helping locate the source of anomalous events.

Ablation Study

We also explore different strategies for anomaly detection to demonstrate the effectiveness of different components in SDA^{2}D: (1) Train with only time feature; (2) Train with only spatial feature; (3) Train with both time and spatial feature, but without time-spatial interaction; (4) Perform AD with only time reconstruction errors; and (5) Perform AD with only state derivative reconstruction errors. We calculate the average VUS-PR on four selected datasets, and the experimental results can be found in Tab.2. We can observe that both time and spatial features are necessary for more accurate anomaly detection, and SDA^{2}D can benefit from reconstruction errors on both time and derivative.



(a) Anomaly diagnosis on dataset SMAP



(b) Anomaly diagnosis on dataset SWaT

Figure 4: Cases of anomaly diagnosis on different datasets.

Limitation and Future Work

Although improvements can be observed, the calculation of NCDEs is a source-intensive process that may require more time for MTS with more variables. Meanwhile, it is also our future work to establish explicit links between derivative vectors and different working conditions for higher-level interpretability in real-world applications.

5 Conclusion

In this work, we propose SDA^{2}D, a novel method for MTS AD, which can learn the ability to reconstruct from both the time level and the derivative level to further capture the evolution of a given system, addressing the issue of insufficient detection support. Demonstrating its informative features to describe system dynamics, SDA^{2}D not only achieves better performance on four real-world datasets, but also helps locate the source of abnormal events.

Acknowledgments

This work was supported in part by NSFC under grant No.U23A20326, Key R&D Program of Zhejiang Program No.2024C01065, State Key Laboratory of ICT Project No.ICT2025A09, Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China NO.JYB2025XDXM103, and NSFC under Grant No.62293511.

References

Almeida, A.; Brás, S.; Sargento, S.; and Pinto, F. C. 2023. Time series big data: a survey on data stream frameworks, analysis and algorithms. *Journal of Big Data*, 10(1): 83.

Audibert, J.; Michiardi, P.; Guyard, F.; Marti, S.; and Zuluaga, M. A. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 3395–3404.

Boniol, P.; Krishna, A. K.; Bruel, M.; Liu, Q.; Huang, M.; Palpanas, T.; Tsay, R. S.; Elmore, A.; Franklin, M. J.; and Paparrizos, J. 2025. VUS: effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal*, 34(3): 32.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Dai, Z.; He, L.; Yang, S.; and Leeke, M. 2024. SARAD: Spatial association-aware anomaly detection and diagnosis for multivariate time series. *Advances in Neural Information Processing Systems*, 37: 48371–48410.

Darban, Z. Z.; Webb, G. I.; Pan, S.; Aggarwal, C. C.; and Salehi, M. 2025. CARLA: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition*, 157: 110874.

Deng, J.; Chen, X.; Jiang, R.; Song, X.; and Tsang, I. W. 2021. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 269–278.

Di Martino, F.; and Delmastro, F. 2023. Explainable AI for clinical and remote health applications: a survey on tabular and time series data. *Artificial Intelligence Review*, 56(6): 5261–5315.

Dupont, E.; Doucet, A.; and Teh, Y. W. 2019. Augmented neural odes. *Advances in neural information processing systems*, 32.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C.-K.; Li, X.; and Guan, C. 2023. Self-supervised contrastive representation learning for semi-supervised time-series classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 15604–15618.

Fang, Y.; Xie, J.; Zhao, Y.; Chen, L.; Gao, Y.; and Zheng, K. 2024. Temporal-frequency masked autoencoders for time series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 1228–1241. IEEE.

Farahani, M. A.; McCormick, M.; Gianinny, R.; Hudacheck, F.; Harik, R.; Liu, Z.; and Wuest, T. 2023. Time-series pattern recognition in smart manufacturing systems: A literature review and ontology. *Journal of Manufacturing Systems*, 69: 208–241.

Goh, J.; Adepu, S.; Junejo, K. N.; and Mathur, A. 2017. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*, 88–99. Springer.

Huang, S.; Zhou, Q.; Shen, J.; Zhou, H.; and Yong, B. 2024. Multistage spatio-temporal attention network based on NODE for short-term PV power forecasting. *Energy*, 290: 130308.

Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; and 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*, 387–395.

Jhin, S. Y.; Shin, H.; Kim, S.; Hong, S.; Jo, M.; Park, S.; Park, N.; Lee, S.; Maeng, H.; and Jeon, S. 2024. Attentive neural controlled differential equations for time-series classification and forecasting. *Knowledge and Information Systems*, 66(3): 1885–1915.

Kamm, S.; Veekati, S. S.; Müller, T.; Jazdi, N.; and Weyrich, M. 2023. A survey on machine learning based analysis of heterogeneous data in industrial automation. *Computers in Industry*, 149: 103930.

Kidger, P. 2022. On neural differential equations. *arXiv preprint arXiv:2202.02435*.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33: 6696–6707.

Li, J.; Izakian, H.; Pedrycz, W.; and Jamal, I. 2021. Clustering-based anomaly detection in multivariate time series data. *Applied Soft Computing*, 100: 106919.

Li, J.; Zhu, Z.; et al. 2023. Neural lad: a neural latent dynamics framework for times series modeling. *Advances in Neural Information Processing Systems*, 36: 17345–17356.

Li, M.; Feng, X.; and Belgiu, M. 2024. Mapping tobacco planting areas in smallholder farmlands using Phenological-Spatial-Temporal LSTM from time-series Sentinel-1 SAR images. *International Journal of Applied Earth Observation and Geoinformation*, 129: 103826.

Liu, Q.; and Paparrizos, J. 2024. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37: 108231–108261.

Liu, Z.; Du, B.; Ye, J.; Wen, X.; and Sun, L. 2024. An NCDE-based framework for universal representation learning of time series. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 4623–4633.

Paparrizos, J.; Boniol, P.; Palpanas, T.; Tsay, R. S.; Elmore, A.; and Franklin, M. J. 2022. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11): 2774–2787.

Shao, Z.; Zhang, Z.; Wang, F.; and Xu, Y. 2022. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 1567–1577.

Shentu, Q.; Li, B.; Zhao, K.; Shu, Y.; Rao, Z.; Pan, L.; Yang, B.; and Guo, C. 2024. Towards a General Time Series Anomaly Detector with Adaptive Bottlenecks and Dual Adversarial Decoders. *arXiv preprint arXiv:2405.15273*.

Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; and Pei, D. 2019. 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*, 2828–2837.

Sun, X.; Zhou, H.; and Li, C. 2025. Multivariate Time Series Anomaly Detection with Idempotent Reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wi, H.; Shin, Y.; and Park, N. 2024. Continuous-time autoencoders for regular and irregular time series imputation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 826–835.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Wu, X.; Qiu, X.; Li, Z.; Wang, Y.; Hu, J.; Guo, C.; Xiong, H.; and Yang, B. 2024. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. *arXiv preprint arXiv:2410.12261*.

Xu, J.; Wu, H.; Wang, J.; and Long, M. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *International Conference on Learning Representations*.

Xu, Z.; Zeng, A.; and Xu, Q. 2023. FITS: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*.

Xu, Z.; Zeng, A.; and Xu, Q. 2024. FITS: Modeling Time Series with $10k$ Parameters. In *The Twelfth International Conference on Learning Representations*.

Yairi, T.; Takeishi, N.; Oda, T.; Nakajima, Y.; Nishimura, N.; and 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.

Yang, Y.; Zhang, C.; Zhou, T.; Wen, Q.; and Sun, L. 2023. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3033–3045.

Yu, Y.; Si, X.; Hu, C.; and Zhang, J. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7): 1235–1270.

Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; and Xu, B. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 8980–8987.

Zamanzadeh Darban, Z.; Webb, G. I.; Pan, S.; Aggarwal, C.; and Salehi, M. 2024. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1): 1–42.

Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; and Eickhoff, C. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2114–2124.

Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; and 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*, volume 33, 1409–1416.

Zhang, K.; Yang, Q.; Li, C.; Sun, X.; and Chen, J. 2025. Missing Data Recovery Methods on Multivariate Time Series in IoT: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*.

Zhang, Y.; Chen, Y.; Wang, J.; and Pan, Z. 2021. Unsupervised deep anomaly detection for multi-sensor time-series signals. *IEEE Transactions on Knowledge and Data Engineering*, 35(2): 2118–2132.

Zhao, H.; Wang, Y.; Duan, J.; Huang, C.; Cao, D.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; and Zhang, Q. 2020. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, 841–850. IEEE.

Zhou, H.; Zhou, Q.; Tang, X.; Shen, J.; Yong, B.; and Huang, Y. 2025. Electrical load forecasting based on the fusion of multi-scale features extracted by using neural ordinary differential equation. *The Journal of Supercomputing*, 81(1): 49.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.

Zong, B.; Song, Q.; Min, M. R.; Cheng, W.; Lumezanu, C.; Cho, D.; and Chen, H. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.