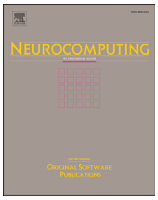
Neurocomputing 662 (2026) 131978

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Spectral decomposition and adaptation for non-stationary time series anomaly detection

Huanyu Zhang [a, b,*], Yi-Fan Zhang [a, b], Jian Liang [a, b], Zhang Zhang [a, b], Liang Wang [a, b]

[a] NLPR, CASIA, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China

HIGHLIGHTS

- We propose a spectral decomposition based approach for unsupervised time series anomaly detection, which reduces the complexity of reconstruction on non-stationary time series. Additionally, we introduce an entropy regularization to promote the model in fostering strong correlations between normal time steps, thereby enhancing the distinguishability between normal and abnormal instances.
- We design a new test-time adaptation (TTA) method for unsupervised time series anomaly detection to address the distribution gap between training and inference. During inference, we assign pseudo labels based on the distribution of anomaly scores and expand the gap between the scores of different instances to improve the performance.
- Our SDA achieves superior performance while cutting training time by two-thirds and using a model size of less than 2MB across three real-world benchmarks. Extensive ablation experiments verify the efficiency of each component of our method.

ARTICLE INFO

Communicated by R. Zhu

Keywords:
Time series
Anomaly detection
Test-time adaptation

ABSTRACT

Unsupervised anomaly detection in time series data is crucial for identifying unusual patterns across various fields. However, existing methods often struggle when dealing with non-stationary time series, constraining their practical application. In this paper, we delve into the challenges surrounding non-stationary time series and put forward a novel framework along with a test-time adaptation strategy. When it comes to the framework, non-stationary time series pose difficulties for modeling due to their blend of time-invariant statistics and evolving temporal dependencies. To address this issue, we explicitly break down the input into variant and invariant components through spectral analysis, with the aim of separately modeling these aspects. Besides, the absence of anomalies during training leads to significant distribution discrepancies between training and testing phases, which is ignored by most existing methods. To deal with this, we propose a flexible test-time adaptation strategy to further amplify the normal-abnormal distinguishability. Our proposed **S**pectral **D**ecomposition and **A**daptation method (SDA) outperforms existing detection frameworks in terms of effectiveness and efficiency. Specifically, compared to state-of-the-art models, SDA achieves superior performance while reducing training time by **66.2** % and memory usage by **98.8** %.

1. Introduction

Time series analysis encompasses a wide range of tasks, including forecasting [1–3], classification [4–6], and anomaly detection [7,8]. Time series anomaly detection aims to detect abnormal patterns or events in the sequence data collected over time [9]. It facilitates early warnings and precautions in advance that potentially prevent large malfunctions, which are crucial for a broad variety of real-world applications, such as finance, healthcare, transportation, manufacturing and other fields [4,10–14]. An anomaly is also referred to as an outlier or novelty, denoting an observation that is deemed unusual, irregular, inconsistent, or unexpected [15–17]. However, anomalies are usually rare and hidden by vast normal points, making the data labeling hard and

* Corresponding author at: NLPR, CASIA, Beijing, China.
Email address: huanyu.zhang@cripac.ia.ac.cn (H. Zhang).

https://doi.org/10.1016/j.neucom.2025.131978
Received 18 May 2025; Received in revised form 12 August 2025; Accepted 29 October 2025
Available online 3 November 2025

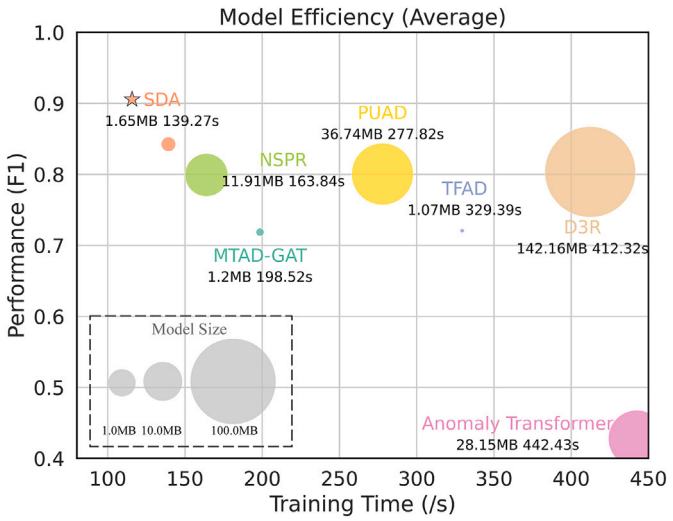


Fig. 1. Average model efficiency comparison over three benchmarks (PSM, SMD, and SWaT). Our SDA achieves the best performance with a compact model size and low time cost compared to other baselines.

expensive. Therefore, unsupervised time series anomaly detection is an essential problem in data mining and industrial applications(Fig. 1).

Many unsupervised methodologies entail computing of an anomaly score at each time step. The score is subsequently compared to a threshold in order to determine whether a time step is an anomaly or not. Researchers have designed various methods to deal with the issue of the computation of anomaly scores, which can be categorized into three main groups: dissimilarity-based methods [18–20], prediction-based methods [21–23] and reconstruction-based methods [24–27]. Recently, reconstruction-based methods have been developing rapidly due to their ability to handle complex data by integrating with machine learning models, as well as their interpretability in identifying anomalies.

However, due to the inherent non-stationarity in real-world time series, there still exist some challenges in practice. Non-stationary time series are characterized by time-variant statistics and temporal dependencies across various periods [28,29]. This dynamic nature often results in a substantial distribution gap between different periods, making accurate modeling and reconstruction a non-trivial task. Additionally, the distribution gap between training and inference is also crucial to the ultimate performance of unsupervised time series anomaly detection, which is ignored by most existing methods [24,26,27]. A recent work [30] has attempted to handle the ”new normal” instances within the test set, which removes the trend component and only updates model parameters with the remainder. However, rare abnormal instances may contain unseen patterns that could offer more valuable knowledge compared to commonplace normal points.

In this paper, we propose a novel **S**pectral **D**ecomposition and Reconstruction model as well as a new Test-Time **A**daptation method (**SDA**) to address these issues. The proposed SDA model is composed of several stackable SDA Blocks. To deal with non-stationary time series, each SDA Block aims to decompose a sequence into time-variant and time-invariant components through spectral analysis. Through our spectral decomposition, simple multi-layer perceptrons (MLPs) can be used to reconstruct sequential associations, thereby significantly reducing model complexity significantly. Moreover, for time-variant components that exhibit strong local dependencies, we introduce an entropy regularization to promote normal time steps forming strong associations with their nearby neighbors. In order to address the distribution gap between training and inference, we assign pseudo labels based on the distribution of anomaly scores during inference. Subsequently, the gap between scores of pseudo-normal and pseudo-abnormal instances will be further amplified, leading to better performance. Through our adaptation method, SDA can leverage patterns underlying in both abnormal and normal data. Consequently, our SDA effectively performs unsupervised anomaly detection in non-stationary time series and exhibits robustness to distribution shifts between the training and test sets. Additionally, due to MLP-based model and parameter-free decomposition, our SDA achieves superior performance compared to the baselines with reduced model size and training time.

The contributions of our paper are summarised as follows:

- We propose a spectral decomposition based approach for unsupervised time series anomaly detection, which reduces the complexity of reconstruction on non-stationary time series. Additionally, we introduce an entropy regularization to encourage the model to foster strong correlations between normal time steps, thereby enhancing the distinguishability between normal and abnormal instances.
- We design a new test-time adaptation (TTA) method for unsupervised time series anomaly detection to address the distribution gap between training and inference. During inference, we assign pseudo labels based on the distribution of anomaly scores and expand the gap between the scores of different instances to improve the performance.
- Our SDA achieves superior performance while cutting training time by two-thirds and using a model size of less than 2MB across three real-world benchmarks. Extensive ablation experiments verify the efficiency of each component of our method.

2. Related work

Unsupervised time series anomaly detection. Unsupervised time-series anomaly detection methods aim to identify observations that significantly deviate from normal patterns without relying on labeled data [31–33]. Early approaches include classical outlier detection algorithms, such as Local Outlier Factor (LOF)[18] and Isolation Forest[34,35], which estimate anomaly scores based on the density or isolation of data points in a feature space. Variants like Deep Isolation Forest (DIF) combine neural networks with isolation mechanisms to improve performance[36]. Traditional one-class classification techniques (e.g., One-Class SVM/SVDD) have also been applied by learning a decision boundary around normal data[37]. Modern deep learning approaches for time-series anomalies can be broadly categorized by their learning objectives: Forecasting-based methods [21–23] rely on detecting anomalies by comparing the prediction of the subsequent value within the time series with the actual value. Density estimation-based methods [18,38] transform time series data into a feature space and estimate the probability density function of normal data points. Reconstruction-based methods [24,27,32,33,39], reconstruct normal time-series data and identify abnormal time-series data with high reconstruction errors. Another line of work uses contrastive learning to learn discriminative representations: DCdetector[40] is a recent method that foregoes explicit forecasting/reconstruction and instead trains a model on self-supervised tasks. Similarly, one-class neural networks have been adapted for time series: for example, COUTA[41] learns a robust boundary of normality by penalizing uncertain predictions and by generating synthetic anomalies to inform the model. These methods are effective in capturing complex patterns, but may be sensitive to the choice of reconstruction models. Additionally, they are also computationally expensive. To handle non-stationary time series, [27] propose a dynamic decomposition reconstruction method based on transformers. However, it only reconstructs the stable component of input, which limits its performance in the real-world time series. In contrast, SDA proposes a novel spectral decomposition method, which can better understand the underlying dynamics and identify anomalies more efficiently. Additionally, with MLP-based modules, SDA significantly reduces the training time and model size.

Test-time adaptation. In order to mitigate the performance degradation caused by distribution shifts, a range of fully TTA methods

[42–44] have been developed. In the broader machine learning literature, representative approaches include entropy minimization strategies, such as Tent[43], which adjust model parameters to make predictions more confident on test samples, and pseudo-label–based self-training approaches[45–47], which iteratively update the model using test instances with high-confidence predictions. These methods have been extensively explored in computer vision tasks such as image classification and segmentation, showing strong robustness under domain shifts. A recent work [30] has attempted to employ TTA methods in unsupervised time series anomaly detection by denormalizing the input and minimizing the reconstruction loss of the pseudo-normal instances. However, using moving average to estimate the trend may cause extra reconstruction errors. Besides, updating only with all pseudo-normal instances can also introduce bias. In contrast, SDA designs a new TTA strategy using both pseudo-normal and pseudo-abnormal instances for model updating, thereby expanding the gap between the scores of the normal and abnormal instances.

3. Proposed method

Preliminary: An input of multivariate time series anomaly detection is denoted by $x = [x^{(1)}, x^{(2)}, \dots, x^{(n)}] \in \mathbb{R}^{n\times c}$, where n is the length of timestamps, c is the number of channels, and $x^{(k)}$ represents the kth instance. The task is to produce an output vector $y = [y^{(1)}, y^{(2)}, \dots, y^{(n)}] \in \mathbb{R}^n$, where $y^{(k)} \in \{0, 1\}$ denotes whether the kth timestamp is an anomaly.

Overview: The overall SDA framework is shown in Fig. 2. We propose a model architecture for spectral decomposition and reconstruction. The model is composed of N SDA Blocks. Each SDA Block is designed to decompose and reconstruct the non-stationary time series. Inspired by [48], a residual structure is employed to capture hierarchical dynamics and facilitate deep decomposition. The detailed model architecture is shown in Fig. 3(a). During the training phase, the model is trained adhering to the reconstruction loss and entropy regularization. During the test phase, the model allocates pseudo labels to the test samples in terms of the anomaly score. And a mixture of Gaussians is adopted for TTA, which confers an advantage in augmenting the distinction between anomaly scores of normal and abnormal samples. The overall SDA framework results in improved performance for unsupervised anomaly detection.

3.1. SDA block

3.1.1. Spectral decomposition:

We adopt a divide-and-conquer strategy to decompose a complex non-stationary time series into various dynamic factors and then reconstruct each independent component separately. By breaking down time series into time-invariant and time-variant components, the model can better learn the underlying dynamics and identify anomalies more effectively. Specifically, we perform the Fast Fourier Transform (FFT) of each input, to compute the average amplitude of each spectrum $\mathcal{S} = \{0, 1, \dots, [n/2]\}$, and order them based on their respective amplitudes. We select the upper α percentile spectrums, encompassing dominant spectrums shared across all subsequences and reflecting time-invariant dynamics inherent in the dataset. The remaining spectrums constitute the distinctive elements for varying subsequences during different periods. Consequently, we partition the spectrum set $\mathcal{S}$ into $\mathcal{S}_\alpha$ and its complementary set $\overline{\mathcal{S}}_\alpha$. Then the decomposition outputs x^{var} and x^{inv} are obtained by employing inverse FFT for further reconstruction. The input of the ith SDA Block is denoted as $x_i \in \mathbb{R}^{n\times c}$. As shown in Fig. 3(a), the decomposition process is formulated as follows:

$$x_i^{var}, x_i^{inv} = \text{Fourier Filter}(x_i). \tag{1}$$

3.1.2. Reconstruction modules:

We introduce two separate MLP-based modules to reconstruct time-varying and time-invariant components, respectively.

Time-invariant reconstruction module: The module is designed to learn the globally shared dynamics, which model the long-term temporal patterns for reconstruction. We use the MLP as the Encoder and Decoder in the module. The complete reconstruction process is depicted in the following formula:

$$z^{inv} = \text{Encoder}(x^{inv}),\ \hat{x}^{inv} = \text{Decoder}(z^{inv}). \tag{2}$$

Time-variant reconstruction module: The time-variant dynamics change continuously and present a greater level of complexity compared to the time invariant component. Hence, a self-attention mechanism is employed in preparation for subsequent encoding in the *Time-variant Reconstruction Module*. To reduce the computational complexity, the input x^{var} is first divided into $m = \lfloor \frac{(n-l)}{s} \rfloor + 1$ patches $x_p^{var} \in \mathbb{R}^{l\times c}$ of length l, where s represents the non-overlapping stride between adjacent patches. Before dividing, s repeated numbers of the last time step value are padded to the end of the original sequence. Then, we calculate self-attention in each patch as follows:

$$\begin{aligned} [Q_p, K_p, V_p] &= x_p^{var}[W_p^Q, W_p^K, W_p^V], \\ \text{Correlation}: T_p &= \text{Softmax}(\frac{Q_p \cdot K_p^T}{\sqrt{d_m}}), \\ \widetilde{x}_p^{var} &= f(T_p \cdot V_p), \end{aligned} \tag{3}$$

where $Q_p, K_p, V_p \in \mathbb{R}^{l\times d_m}$ represent the query, key, and value. $W_p^Q, W_p^K, W_p^V \in \mathbb{R}^{c\times d_m}$ represent the learnable projection matrices for Q_p, K_p, V_p, respectively. $T_p \in \mathbb{R}^{l\times l}$ denotes the learned correlations across time steps within the pth patch. And $f(\cdot)$ is a linear layer for mapping.

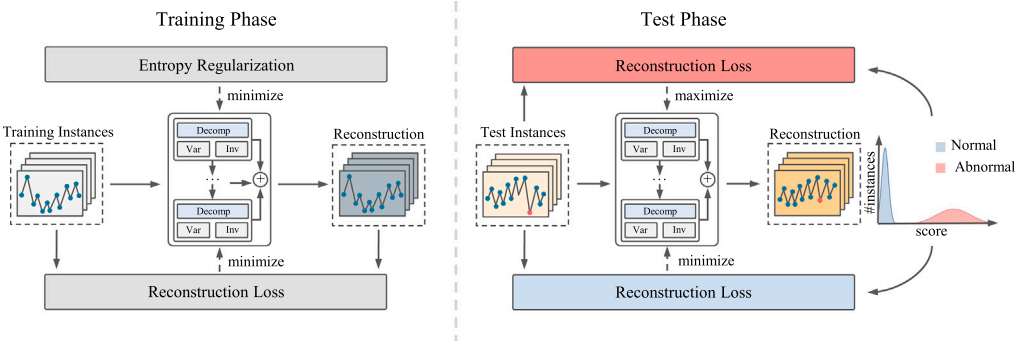
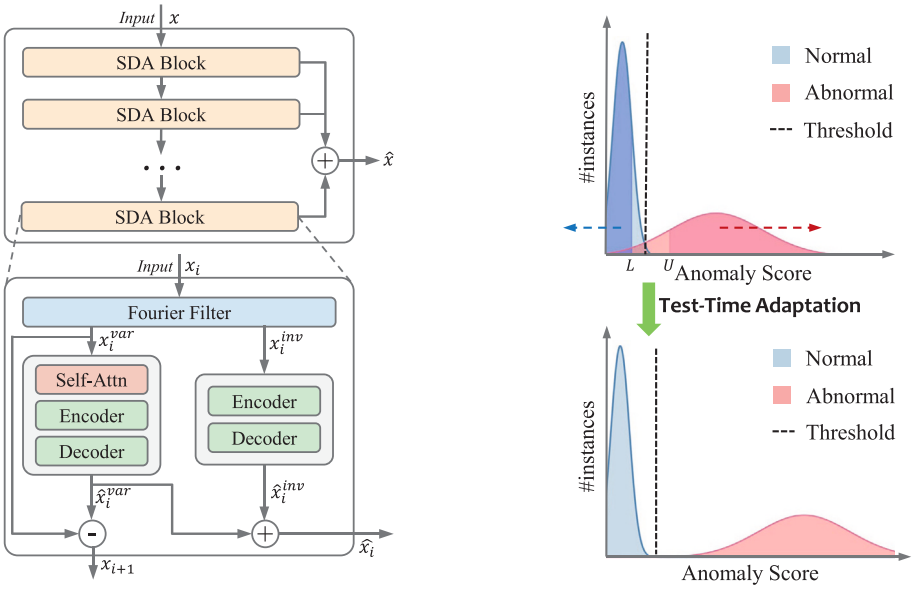


Fig. 2. The overview of SDA. During training, the model decomposes and reconstructs the input from the training set under the constraints of reconstruction loss and the proposed entropy regularization. During test, we employ a Gaussian mixture model to fit the anomaly scores of normal and abnormal instances and expand the gap between them to make anomalies easier to be detected.



(a) **Model Architecture** (b) **Test-Time Adaptation**

Fig. 3. (a) Model architecture: Our SDA is composed of N stackable SDA Blocks and each SDA Block contains a Fourier Filter, a Time-Variant Reconstruction Module and a Time-invariant Reconstruction Module. **(b) Test-Time adaptation:** The anomaly score of all instances resembles a mixture of Gaussian distributions. As shown in the upper plot, we identify pseudo-normal instances (the dark blue shaded area) and pseudo-abnormal instances (the dark red shaded area) by setting two bounds U and L. Our TTA strategy is to expand the gap between their scores during inference.

We obtain the output $\widetilde{x}^{var} = [\widetilde{x}_0^{var}, \ldots, \widetilde{x}_p^{var}, \ldots, \widetilde{x}_m^{var}]$. Similar to the *Time-invariant Reconstruction Module*, we also employ MLP as the Encoder and Decoder. The subsequent reconstruction process is depicted in the following formula:

$$z^{var} = \text{Encoder}(\widetilde{x}^{var}),\ \hat{x}^{var} = \text{Decoder}(z^{var}). \tag{4}$$

In general, the different modules achieve reconstruction of distinct components respectively:

$$\hat{x}_i^{var} = \text{TimeVarRec}(x_i^{var}),\ \hat{x}_i^{inv} = \text{TimeInvRec}(x_i^{inv}). \tag{5}$$

The final reconstructed data $\hat{x}_i$ is the sum of the outputs derived from both *Time-invariant Reconstruction Module* and *Time-variant Reconstruction Module*. Meanwhile, the residual x_{i+1} is fed to next SDA Block for further decomposition and reconstruction. The outputs are achieved as follows:

$$\hat{x}_i = \hat{x}_i^{var} + \hat{x}_i^{inv},\ x_{i+1} = x_i^{var} - \hat{x}_i^{var}. \tag{6}$$

It is noted that both reconstruction modules are primarily based on MLP, which significantly reduces the model size compared to transformer-based models.

3.2. Reconstruction with entropy regularization

Following previous work [27,33], we minimize the reconstruction loss $\mathcal{L}_{\text{recon}}$ during training. Due to the intricate nature of time variant components, which encompass detailed patterns and demonstrate strong local dependencies, normal time steps tend to form more robust correlations with the nearest normal steps within the patch. By contrast, it is harder for anomalous time steps to establish strong correlations with other normal steps. Thus, we further introduce an entropy regularization loss to make normal time steps foster close associations with their local neighbors as much as possible, as the training set only contains normal samples. The entropy regularization loss is defined as:

$$\mathcal{L}_{\text{ent}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{m} -T_p^i \log T_p^i, \tag{7}$$

where T_p^i represents the pth correlation in ith Block. Consequently, we minimize the entropy regularization along with the reconstruction loss during training. The specific training loss is:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{ent}} \cdot \mathcal{L}_{\text{ent}}, \tag{8}$$

where λ_{ent} is the weight of the entropy regularization.

3.3. Anomaly detection with test-time adaptation

With the assistance of our SDA model, the majority of abnormal instances can be identified through the anomaly score. The score of each instance $x^{(i)}$ is obtained through the reconstruction error between test samples and their respective reconstructions:

$$\text{Score}(x^{(i)}) = \frac{1}{c}\sum_{j=1}^{c}(x^{(i,j)} - \hat{x}^{(i,j)})^2. \tag{9}$$

To further increase the the disparity between normal and abnormal instances, we minimize the reconstruction loss of pseudo-normal instances, denoted as $\mathcal{L}_{\text{nor}}$, and maximize the reconstruction loss of pseudo-abnormal instances, denoted as $\mathcal{L}_{\text{abn}}$. However, the allocation of pseudo labels is challenging in unsupervised time series anomaly detection. In our empirical studies, we noted that the distribution of normal instances and abnormal instances closely resembles a Gaussian mixture model, akin to the upper plot in Fig. 3(b).[1] Although most anomalies are detected by the threshold, it is typical for certain abnormal instances to escape accurate differentiation because of low anomaly scores. Furthermore, some normal instances may be erroneously classified as

[1] We visualize the anomaly score distribution for all instances in Sec. 4.8.

anomalies due to the elevated anomaly scores. Therefore, directly assigning pseudo labels based on the threshold may lead to a significant drop in performance due to numerous misclassified instances. To address this issue, we first fit a Gaussian mixture model with p modes based on the anomaly scores of all test samples. Then the first two dominant modes ($\mathcal{N}_1(\mu_1, \sigma_1)$ and $\mathcal{N}_2(\mu_2, \sigma_2)$) are selected as the distributions of normal instances and abnormal instances, respectively. Next, we define the upper bound U and lower bound L as follows:

$$L = \mu_1 + \sigma_1;\ U = \mu_2 - \sigma_2. \tag{10}$$

Instances with anomaly scores less than L are displayed in the dark blue shaded area in the upper plot of Fig. 3(b), labeled as *pseudo-normal* instances. Conversely, instances with scores greater than U, are labeled as *pseudo-abnormal* instances, as shown in the dark red shaded area. By introducing U and L, the model can update with samples of high confidence without label. The specific loss functions during inference are formulated as follows:

$$\begin{aligned}
\mathcal{L}_{\text{nor}} &= \frac{1}{nc}\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbb{1}(\text{Score}(x^{(i)}) < L)(x^{(i,j)} - \hat{x}^{(i,j)})^2, \\
\mathcal{L}_{\text{abn}} &= \frac{1}{nc}\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbb{1}(\text{Score}(x^{(i)}) > U)(x^{(i,j)} - \hat{x}^{(i,j)})^2, \\
\mathcal{L}_{\text{test}} &= \mathcal{L}_{\text{nor}} - \lambda_{\text{abn}} \cdot S(\mathcal{L}_{\text{abn}}),
\end{aligned} \tag{11}$$

where $\mathcal{L}_{\text{test}}$ is the final loss during testing, λ_{abn} is the weight of $\mathcal{L}_{\text{abn}}$ and S represents the sigmoid operation for avoiding disruption caused by extreme values. After adaptation, the anomaly scores of real normal and abnormal instances will differ more significantly, making them easier to distinguish.

4. Experiments

4.1. Datasets

We evaluate SDA extensively on three real-world multivariate datasets: **Pooled server metrics dataset:** The Pooled Server Metrics (PSM) dataset is proposed by eBay and consists of 26 dimensional data captured internally from application server nodes [49]. **Server machine dataset:** The Server Machine Dataset (SMD) is a 5-week-long server status log dataset, which is collected by large Internet company. It contains performance metrics from servers with 38 dimensions [31]. **Secure water treatment dataset:** The Secure Water Treatment (SWaT) dataset is a reduced representation of a real industrial water treatment plant and has 51 different values in an observation [50]. **HexagonML (UCR) dataset:** This is a dataset of multiple univariate time series that was used in the KDD 2021 Cup [51]. We include only the datasets obtained from natural sources (the InternalBleeding and ECG datasets). Since Fourier transform requires continuous signals, we follow [27] and do not employ MSL (Mars Science Laboratory rover) and SMAP (Soil Moisture Active Passive satellite) [22]. The statistical details of these datasets are provided in Table 1. We split the training data into 80 % for training and 20 % for validation. Besides, anomalies are only present in the testing data.

4.2. Metrics

For performance evaluation, we use precision, recall, and F1 score in our main experiments and ablation analysis. The point adjustment method has been widely used in previous work [33,39,52]. However, as [27,53] pointed out, it is unreasonable and creates the illusion of progress. Therefore, we employ the recently proposed evaluation measures based on affiliation [54], which are a theoretically grounded, robust and interpretable extensions to precision/recall metrics.

Table 1
Statistics of the datasets.

Dataset	Training Size	Test Size	Anomaly Durations	Anomaly Rate	Frequency
PSM	132,481	87,841	1~8,861	0.2776	1 minute
SMD	23,688	23,689	3~3,161	0.1565	1 minute
SWaT	6840	7500	3~599	0.1263	1 minute
UCR	1600	5900	12~358	0.0188	–

4.3. Baselines

We extensively compare our method with 17 baselines for comprehensive evaluations, including the probability-based methods: COPOD [55], ECOD [56]; the linear transformation-based methods: OCSVM [57]; the outlier-based methods: IForest [34], LODA [58]; the proximity-based methods: CBLOF [59], HBOS [60], and MSAD [61]; the neural network-based methods: VAE [62], DeepSVDD [63], LSTM-AE [64], MTAD-GAT [39], TFAD [24], Anomaly Transformer [33], PUAD [65], LUAD [8], NPSR [26] and D3R [27]. All baselines are based on our runs or sourced from previous work [27], We employ official or open-source implementations published on GitHub and follow the configurations recommended in their papers.

4.4. Implementation details

In our experiment, the sliding window has a fixed size of 64 for all datasets. We use grid search to obtain the best SPOT parameters for each dataset and record the results with the highest F1 scores. All experiments are performed on an Ubuntu Server with a 12th Gen Intel(R) Core(TM) i9-12900 K @ 3.60 GHz processor and an NVIDIA GeForce RTX 4090 Graphics Card. For reproducibility, we list detailed hyperparameters of model structure below and report the following best hyperparameters for our method:

1. For PSM, we train for 8 epochs, with a learning rate of $1.0\times e^{-3}$, a weight decay of $1.0\times e^{-4}$, and the weight of entropy loss λ_{ent} set to 0.006. Besides, the length of patch is set to 16 and the number of SDA Blocks is 2. In Fourier Filter, the α is set to 0.2. During testing, we set the learning rate to $5.0\times e^{-3}$ and the maximum loss weight $\lambda_{\max}$ to 1.0. And we set the dimensionality n of the multivariate Gaussian distribution to 3.
2. For SMD, we train for 8 epochs, with a learning rate of $1.0\times e^{-3}$, a weight decay of $1.0\times e^{-4}$, and the weight of entropy loss λ_{ent} is set to 1.0. Besides, the length of patch is set to 32 and the number of SDA Blocks is 2. In Fourier Filter, the α is set to 0.1.During testing, we set the learning rate to $5.0\times e^{-4}$ and the maximum loss weight $\lambda_{\max}$ to 1.0. And we set the dimensionality n of the multivariate Gaussian distribution to 3.
3. For SWaT, we train for 8 epochs, with a learning rate of $1.0\times e^{-3}$, a weight decay of $1.0\times e^{-4}$, and the weight of entropy loss λ_{ent} set to 0.7. Besides, the length of patch is set to 8 and the number of SDA Blocks is 3. In Fourier Filter, the α is set to 0.3.During testing, we set the learning rate to $7.0\times e^{-4}$ and the maximum loss weight $\lambda_{\max}$ to 0.08. And we set the dimensionality n of the multivariate Gaussian distribution to 3.
4. For UCR, we train for 8 epochs, with a learning rate of $1.0\times e^{-3}$, a weight decay of $1.0\times e^{-4}$, and the weight of entropy loss λ_{ent} is set to 1.0. Besides, the length of patch is set to 8 and the number of SDA Blocks is 3. In Fourier Filter, the α is set to 0.1.During testing, we set the learning rate to $5.0\times e^{-4}$ and the maximum loss weight $\lambda_{\max}$ to 0.08. And we set the dimensionality n of the multivariate Gaussian distribution to 3.

4.5. Main results

The SDA without TTA is denoted as SDA(base). As shown in Table 2, both SDA(base) and SDA outperform all the baselines across three real-world datasets, which verifies the superiority of our method. Specifically, SDA achieves the best F1 performance with an average improvement of 4.33 % compared to the strongest baseline. Besides, our

Table 2
Results on real-world multivariate datasets. The higher values for all metrics represent the better performance, and the best F1 scores are highlighted in bold. The SDA without TTA is denoted as SDA(base).

Method	PSM			SMD			SWaT			UCR			Average
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1
COPOD [55]	0.7602	0.3175	0.4479	0.6676	0.1366	0.2268	0.9876	0.1180	0.2108	0.7251	0.2208	0.3385	0.3059
ECOD [56]	0.7460	0.3384	0.4656	0.7398	0.1615	0.2651	0.9761	0.1151	0.2059	0.6279	0.4592	0.5305	0.3667
OCSVM [57]	0.8761	0.4744	0.6155	0.0000	0.0000	0.0000	0.6196	0.7558	0.6810	0.5978	0.8687	0.7082	0.5011
CBLOF [59]	0.5990	0.9845	0.7449	0.8667	0.3352	0.4834	0.6308	0.7091	0.6677	0.4772	0.8525	0.6119	0.6269
HBOS [60]	1.0000	0.0654	0.1228	0.5628	0.8007	0.6610	0.5771	0.8049	0.6722	0.6529	0.6683	0.6605	0.5291
IForest [34]	1.0000	0.0335	0.0648	1.0000	0.0937	0.1713	0.6127	0.6280	0.6203	0.5394	0.8654	0.6646	0.3802
LODA [58]	0.9266	0.4017	0.5605	0.5902	0.6618	0.6240	0.6117	0.7014	0.6535	0.6184	0.5526	0.5837	0.6053
VAE [62]	0.6221	0.8772	0.7280	0.8209	0.4349	0.5686	0.6355	0.7218	0.6759	0.7468	0.6113	0.6723	0.6611
DeepSVDD [63]	0.7405	0.5064	0.6015	0.6498	0.6477	0.6488	0.5911	0.9353	0.7244	0.7853	0.7144	0.7482	0.6806
LSTM-AE [64]	0.7511	0.7586	0.7548	0.8496	0.4349	0.5753	0.6018	0.7219	0.6564	0.7354	0.6295	0.6783	0.6662
MTAD-GAT [39]	0.7990	0.6014	0.6863	0.8590	0.6769	0.7571	0.6590	0.7751	0.7123	0.7917	0.7255	0.7572	0.7282
TFAD [24]	0.7914	0.7163	0.7520	0.5632	0.9783	0.7149	0.6038	0.8196	0.6953	0.6556	0.6956	0.6750	0.7092
Anomaly Transformer [33]	0.5201	0.8504	0.6455	1.0000	0.0319	0.0619	0.5541	0.5994	0.5759	0.4567	0.5981	0.5179	0.4503
LUAD [8]	0.7962	0.7214	0.7569	0.7352	0.9516	0.8295	0.6914	0.8726	0.7715	0.6681	0.8366	0.7429	0.7751
PUAD [65]	0.8319	0.7167	0.7675	0.7471	0.9559	0.8387	0.7165	0.8448	0.7754	0.7779	0.7193	0.7475	0.7871
NPSR [26]	0.6296	0.8537	0.7247	0.8249	0.9603	0.8875	0.7501	0.8238	0.7852	0.7414	0.8019	0.7705	0.7919
MSAD [61]	0.7832	0.7211	0.7805	0.8612	0.9257	0.8923	0.7288	0.7951	0.7605	0.7045	0.8217	0.7586	0.7979
D3R [27]	0.6294	0.9619	0.7609	0.7715	0.9926	0.8682	0.7206	0.8529	0.7812	0.7915	0.8351	0.8127	0.8057
SDA(base)	0.8314	0.7330	0.7791	0.8358	0.9941	0.9081	0.6628	0.9647	0.7857	0.7983	0.8664	0.8310	0.8259
SDA	0.8610	0.7542	**0.8041**	0.8747	0.9867	**0.9273**	0.6700	0.9796	**0.7958**	0.8211	0.8917	**0.8549**	**0.8455**

Table 3
Ablation studies on SDA(base). F1 scores are reported, with higher values indicating better performance. Seasonal-Trend and High-Low represent seasonal-trend decomposition and High-Low Pass Filter. Spectrum and Entropy represent our spectral decomposition and entropy regularization.

Decomposition Module			Entropy	PSM	SMD	SWaT	UCR	Average
Seasonal-Trend	High-Low	Spectrum						
✗	✗	✗	–	0.7233	0.7838	0.7006	0.7222	0.7325
✓	✗	✗	–	0.7426	0.8355	0.7483	0.7674	0.7735
✗	✓	✗	–	0.7361	0.7628	0.7115	0.7911	0.7503
✗	✗	✓	✗	0.7667	0.8832	0.7626	0.8195	0.8080
✗	✗	✓	✓	**0.7791**	**0.9081**	**0.7857**	**0.8310**	**0.8259**

TTA method also achieves an average improvement of 2.20 %, which verifies the effectiveness of the proposed adaptation strategy. These findings highlight the significance of our contributions and underscore the value of our proposed method in addressing non-stationary time series.

4.6. Ablation studies

4.6.1. Training phase

To assess the effectiveness of each design in our SDA during training, we carry out an ablation study on SDA(base) and present the results in Table 3. The first line does not utilize any of our designs, instead it uses only MLP as the encoder and decoder for reconstruction.

Decomposition module: Based on the MLP-based model, we use various decomposition methods as shown in the second to fourth line. In the 2nd and 3rd lines, a seasonal-trend decomposition using STL [66] or a High-Low Pass Filter is employed to replace our decomposition module. Compared to them, the employment of decomposition, as shown in the 4th line, leads to a significant improvement in all the datasets. It demonstrates that the proposed Fourier Filter conducts effective disentanglement, where the amplitude statistics of frequency spectrums from different periods are utilized to exhibit time-agnostic information. Therefore, the model is able to capture temporal patterns underlying in the data and reconstruct non-stationary time series.

Entropy regularization: Subsequently, to further encourage the establishment of correlation between normal steps, the entropy regularization is employed. As a result, the performance is further enhanced and surpasses the best baseline, as shown in the last line.

4.6.2. Test phase

We compare different Test-Time Adaptation (TTA) methods and present the results in Table 4. The recent work M2N2 [30] designed a TTA method based on trend estimation and updated the model only with pseudo-normal instances during inference. We evaluate its performance without the point adjustment method, which is provided in the second line.

Because M2N2 only uses MLP as the backbone and removes the trend component for adaptation, its performance is even worse than the SDA(base). Similar to M2N2, we attempt to assign pseudo labels based on the threshold and only minimize the reconstruction loss of pseudo-normal instances. As indicated in the third row, the performance improves on the PSM and SMD datasets due to high precision before adaptation. However, on the SWaT dataset, directly using the threshold to differentiate normal and abnormal results in performance degradation because of numerous misclassified samples. Besides, we also update the model with both pseudo normal and abnormal instances, as shown in the fourth line. Since the pseudo labels are still allocated by the threshold, the misclassified instances result in decreased performance on the PSM and SWaT datasets. On the contrary, high precision and recall on the SMD dataset mean few misclassified samples. Thus, the performance gets improved.

To address the decrease caused by numerous misclassified samples, our proposed TTA method assigns pseudo labels based on the distribution of anomaly scores. With the defined upper and lower bounds in Eq. (10), the model is updated with confident instances, resulting in improved performance across all datasets. In addition, learning unseen patterns in abnormal instances further enhances the results, as illustrated in the last line.

Table 4
Results of different test-time adaptation methods. The SDA without TTA is denoted as SDA(base) and the best F1 scores are highlighted in bold. SDA(threshold) represents that the assignment of pseudo labels is based on the threshold.

Method	Normal Only	PSM			SMD			SWaT			Average
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	F1
SDA(base)	–	0.8314	0.7330	0.7791	0.8358	0.9941	0.9081	0.6628	0.9647	0.7857	0.8243
M2N2 [30]	✓	0.7632	0.7661	0.7478	0.8299	0.9553	0.8882	0.6108	0.8848	0.7227	0.7862
SDA(threshold)	✓	0.8126	0.7536	0.7820	0.8621	0.9789	0.9168	0.6702	0.9167	0.7743	0.8244
SDA(threshold)	✗	0.8146	0.7493	0.7806	0.8721	0.9867	0.9259	0.6615	0.8891	0.7586	0.8217
SDA	✓	0.8199	0.7587	0.7881	0.8638	0.9869	0.9212	0.6777	0.9449	0.7893	0.8329
SDA (ours)	✗	0.8610	0.7542	**0.8041**	0.8747	0.9867	**0.9273**	0.6700	0.9796	**0.7958**	**0.8424**

Table 5
The performance (F1) of different modes p in Gaussian mixture model.

p	PSM	SMD	SWaT	UCR	Average
3	0.8041	0.9273	0.7958	0.8549	0.8455
5	0.8040	0.9250	0.7892	0.8412	0.8398
10	0.6943	0.9243	0.7889	0.8399	0.8118
20	0.6943	0.9243	0.7888	0.8345	0.8105

4.7. Mixture of Gaussian distributions

We evaluate the impact of using a mixture of Gaussian distributions with varying modes (p) for fitting as shown in Table 5, where each mode represents a distinct Gaussian distribution. The results indicate that the choice of dimensionality has a notable effect on the performance. When p is less than 3, the existence of samples with extremely high scores (greater than 10^3) causes underfitting, resulting in a negative upper bound. As the value of p increases, the process of data fitting becomes more precise, consequently impacting the two Gaussian distributions with the highest weights. As illustrated in Table 5, it is evident that with the increase in p, the performance on all three datasets shows a varying degree of decrease. As higher value of p leads to overfitting, the model updated with fewer instances results in limited performance. The trade-off is an important consideration in the process of data fitting. Consequently, we set the value of p as 3 in order to achieve the best performance.

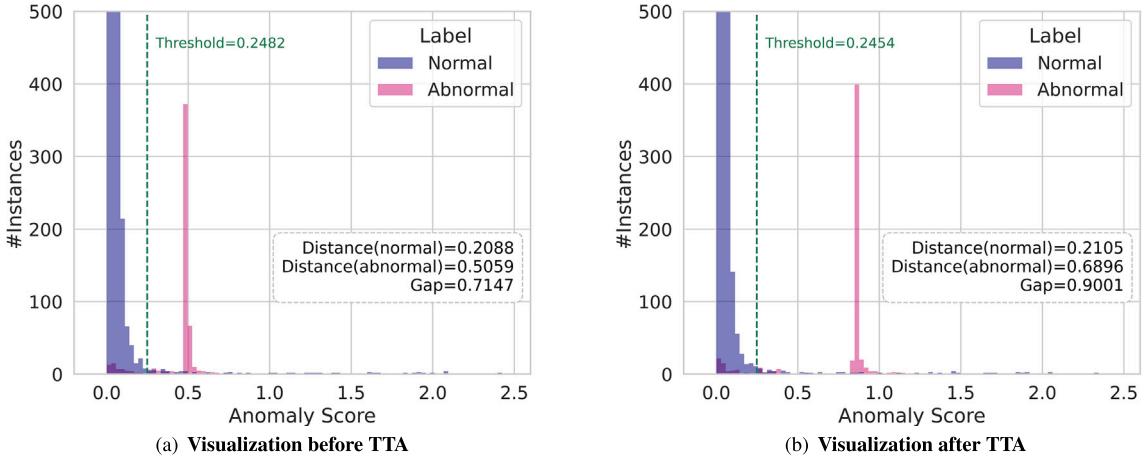
4.8. Anomaly score analysis

The visualization of anomaly scores on the SWaT dataset before and after TTA is shown in Fig. 4. The x-axis depicts anomaly score values, while the y-axis illustrates the corresponding instance counts, with the blue and red denoting normal and abnormal instances, respectively. To enhance clarity, we omit the portions where the sample count exceeds 500 and the score exceeds 3.5. Specifically, the distribution of anomaly scores before TTA is shown in Fig. 4(a), while Fig. 4(b) illustrates the distribution after TTA.

As shown in Fig. 4(a), the distribution of normal instances and abnormal instances closely approximates a Gaussian mixture model. Most normal instances and abnormal instances are clustered within the first two dominant Gaussian distributions, which facilitate the assignment of pseudo labels. In addition, the average distance from the threshold to all normal instances is 0.2088, while the average distance from the threshold to all abnormal instances is 0.5059. Therefore, the disparity in anomaly scores between normal and abnormal instances is 0.7147. After employing TTA, it is observed that the disparity in anomaly scores between normal and abnormal instances increased to 0.9001, as depicted in Fig. 4(b). Besides, the anomaly score distribution of both normal and abnormal instances becomes more concentrated. The visualizations of other datasets are provided in Fig. 5, which demonstrate similar observations. These findings confirm that our proposed TTA method successfully amplifies the normal-abnormal distinguishability. Furthermore, this enhancement in the anomaly score disparity indicates a more robust differentiation capability of the model, allowing for better identification of abnormal instances within the dataset.

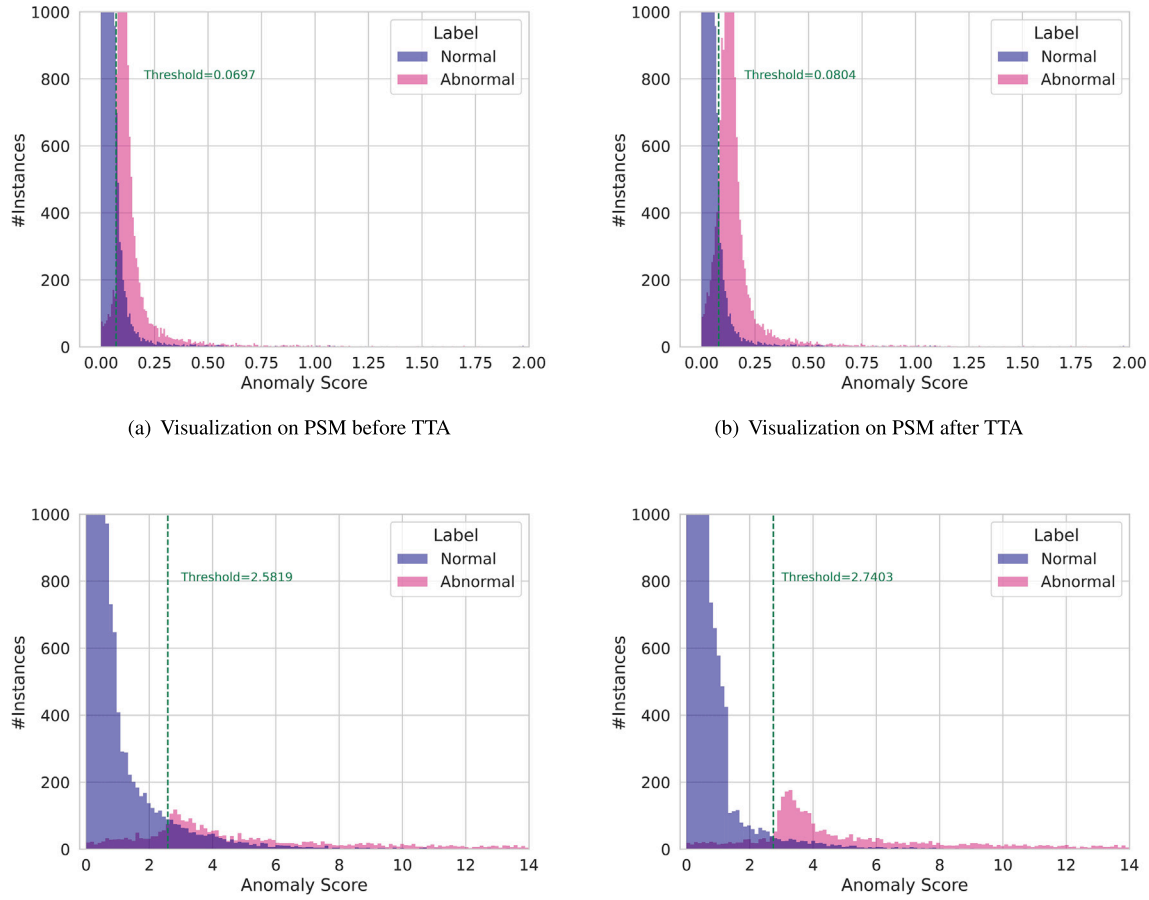
4.9. Distribution comparison

We compare different mixed distribution fitting in Fig. 6. We use log-normal mixture and mixed Gaussian models to fit the anomaly scores. As illustrated in Fig. 6, the mixed Gaussian distribution is more suitable for our TTA method. Furthermore, as shown in the visualization



(a) **Visualization before TTA** (b) **Visualization after TTA**

Fig. 4. Visualization of anomaly scores before and after TTA on SWaT dataset. The distribution of anomaly scores before TTA and after TTA are illustrated in Fig. 4(a) and Fig. 4(b). After employing TTA, the distinction in anomaly scores between normal and abnormal instances becomes more significant.



(a) Visualization on PSM before TTA

(b) Visualization on PSM after TTA

(c) Visualization on SMD before TTA

(d) Visualization on SMD after TTA

Fig. 5. The visualization of anomaly scores before and after TTA on PSM and SMD dataset. The distributions of anomaly scores before TTA are illustrated in Fig. 5(a) and Fig. 5(c), while Fig. 5(b) and Fig. 5(d) illustrate the distributions after TTA. The green dashed line represents the threshold.

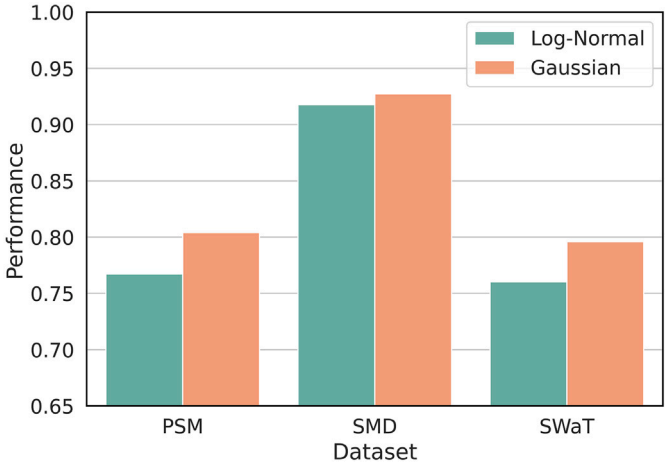


Fig. 6. Mixed distribution fitting comparison. Compared to log-normal mixture, the mixed Gaussian distribution is more suitable for our TTA method.

in Sec. 4.8, we observe that the log-normal mixture fails to capture the underlying patterns of the anomaly scores, which do not align well with the actual data distribution. In contrast, the mixed Gaussian model demonstrates a tighter fit. This outcome confirms our hypothesis that the mixed Gaussian distribution is not only more flexible but also better at representing the distinct clusters present in the data.

4.10. Model efficiency

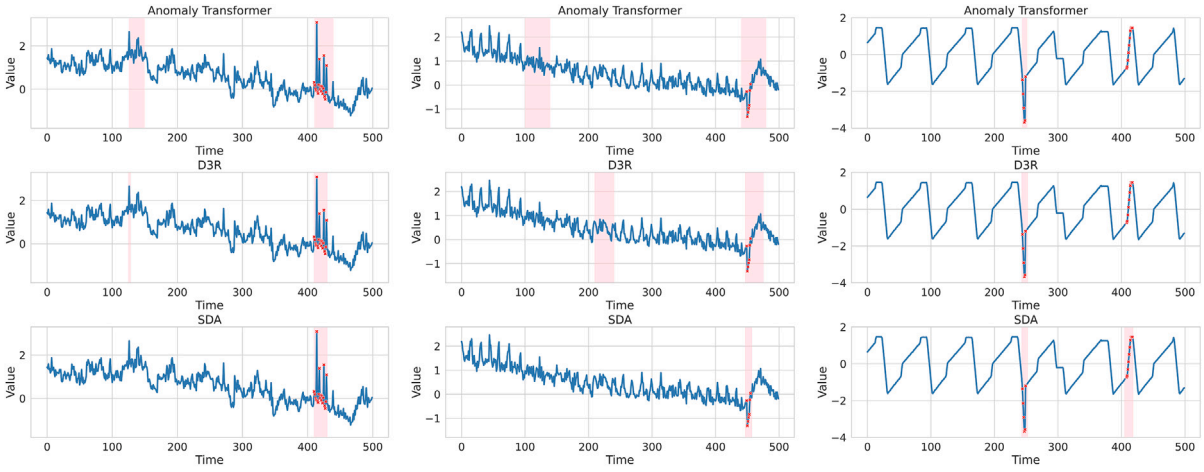
We provide the entire model efficiency comparison in Table 6. Training time represents the time required to train the data for 5 epochs with the same batch size. Inference time is the duration needed to process the entire test dataset. Compared to the state-of-the art model D3R, SDA saves 66.2 % training time and 98.5 % memory across the three real-world datasets with superior performance. Since our TTA method needs to obtain anomaly scores for all the samples and update the model during inference, our inference time is slightly higher than baselines. Compared to the substantial enhancement (4.33 %) in detection accuracy that we have achieved, the inference time remains affordable within the context of the expanding hardware resources of the present era. The combination of reduced training time, lower memory usage, and enhanced accuracy presents a compelling case for the adoption of our TTA method in practical applications. We anticipate that ongoing advancements in hardware will further optimize our model's performance, paving the way for broader implementation in various domains.

4.11. Anomaly detection visualization

In Fig. 7, we present the anomaly detection results of different methods. Compared to the baselines, our SDA demonstrates superior detection accuracy, once again confirming the effectiveness of our model architecture and TTA method. Furthermore, our visualization reveals that the proposed approach not only improves detection accuracy but also enhances the model's robustness against various types of anomalies.

Table 6
Model Efficiency on real-world multivariate datasets. Training time represents the time required to train the data for 1 epoch with the same batch size. Inference time is the duration needed to process the entire test dataset.

Method	PSM			SMD			SWaT		
	Training (s)	Inference (s)	Model Size (MB)	Training (s)	Inference (s)	Model Size (MB)	Training (s)	Inference (s)	Model Size (MB)
VAE [62]	189.24	47.61	0.02	157.91	30.90	0.02	49.45	8.27	0.02
DeepSVDD [63]	81.09	17.25	0.01	60.70	12.61	0.01	24.58	3.15	0.01
LSTM-AE [64]	437.55	125.18	0.03	283.61	72.82	0.04	204.58	15.77	0.05
MTAD-GAT [39]	242.78	107.84	1.25	188.52	60.23	1.20	164.35	12.18	1.24
TFAD [24]	390.14	75.12	1.04	315.39	38.54	1.04	282.64	8.97	1.25
A-Transformer [33]	529.47	151.82	37.27	422.43	94.36	28.15	399.07	30.31	27.49
TranAD [25]	275.14	82.49	10.58	153.84	37.62	11.47	89.14	8.25	11.02
PUAD [65]	350.94	137.41	43.81	256.85	78.67	38.27	225.67	18.53	28.14
NPSR [26]	206.78	124.16	15.34	152.58	55.31	11.05	132.16	14.17	9.35
D3R [27]	533.08	280.68	108.46	399.32	104.12	109.35	126.20	25.91	208.68
SDA	177.18	343.57	1.11	136.27	121.25	0.86	84.36	17.13	2.98



(a) Anomaly Detection Visualization of PSM (b) Anomaly Detection Visualization of SMD (c) Anomaly Detection Visualization of SWaT

Fig. 7. The visualization of anomaly detection results on real-world multivariate datasets. The red markers represent real anomalies. The red shaded area represents the detected anomalies.

4.12. Additional detection results

In order to avoid the influence of SPOT parameters and thresholds on the results, we evaluate the methods based on the original anomaly scores. Prolonged anomalies receive increased weight, potentially leading to inaccurate metrics. Thus, we aggregate each anomaly event. An anomaly event after aggregation is equivalent to only one timestamp, labeled as an anomaly, and scored as the maximum of the original anomaly range. We use area under the ROC curve (AUC) to evaluate the anomaly scores of each method. The value of AUC ranges between 0.5 and 1, with the closer it is to 1.0, the better the method. The results of the experiments are shown in Table 7.

4.13. Hyperparameter analysis

We conduct a series of hyperparameter analysis across three datasets and present the experimental results in Fig. 8. We study the number of SDA blocks, the length of segments, the weight of entropy regularization and abnormal loss, α, and test-time learning rate. The results show that the number of blocks has a significant impact on the overall performance, with a clear trade-off between model complexity and generalization. The length of segments also plays a crucial role, as different datasets have varying temporal patterns. Additionally, we find that the weight of entropy constraint and max loss has a non-linear effect on the model's performance, with an optimal value that varies across different datasets. These findings provide valuable insights for fine-tuning hyperparameters in real-world applications.

Table 7
AUC results on real-world multivariate datasets. The higher values represent the better performance.

Method	PSM	SMD	SWaT	Average
COPOD [55]	0.8526	0.9230	0.7396	0.8384
ECOD [56]	0.8394	0.9220	0.7532	0.8382
OCSVM [57]	0.8708	0.6789	0.5379	0.6959
CBLOF [59]	0.8681	0.9694	0.5378	0.7918
HBOS [60]	0.8150	0.7393	0.8085	0.7876
IForest [34]	0.8892	0.9218	0.7238	0.8450
LODA [58]	0.8619	0.9180	0.6780	0.8193
VAE [62]	0.8583	0.9674	0.5325	0.7861
DeepSVDD [63]	0.8100	0.9187	0.5063	0.7450
LSTM-AE [64]	0.8894	0.9698	0.6255	0.8283
MTAD-GAT [39]	0.9093	0.9443	0.6386	0.8307
TFAD [24]	0.8185	0.9386	0.6966	0.8179
Anomaly Transformer [33]	0.7074	0.7150	0.6638	0.6954
LUAD [8]	0.6914	0.8726	0.7715	0.7859
PUAD [65]	0.8962	0.9148	0.8224	0.8778
NPSR [26]	0.8875	0.9783	0.8561	0.9073
MSAD [61]	0.8962	0.9729	0.8413	0.9035
D3R [27]	0.9223	0.9759	0.8554	0.9179
SDA	**0.9358**	**0.9816**	**0.8579**	**0.9251**

4.14. Decomposition visualization

We present the original data, along with the time-variant and time-invariant components from each dataset in Fig. 9. The red markers

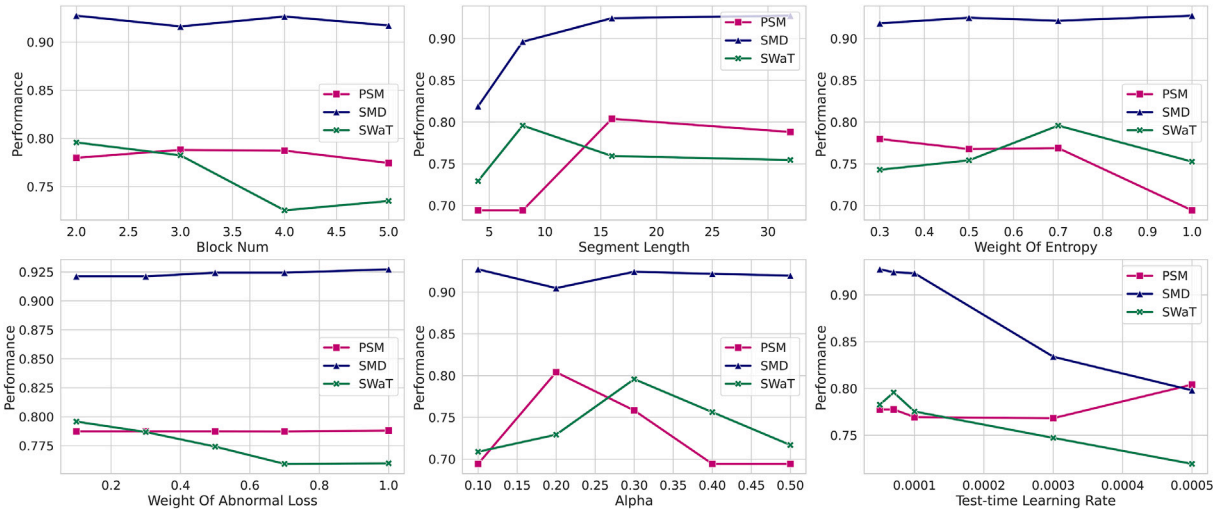
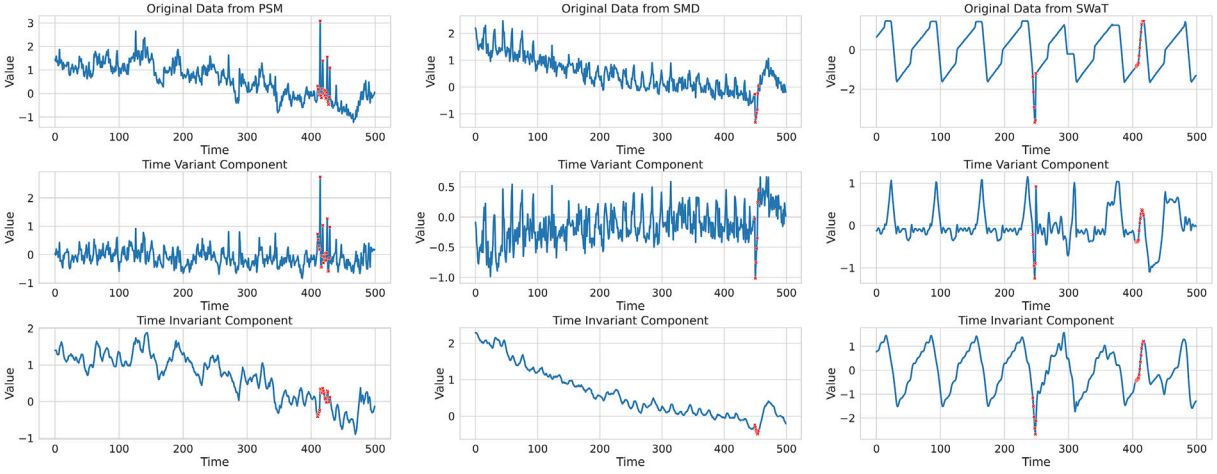


Fig. 8. Results of hyperparameter analysis. The better performance means higher F1 score.



(a) Decomposition Visualization of PSM (b) Decomposition Visualization of SMD (c) Decomposition Visualization of SWaT

Fig. 9. The visualization of decomposition results. The original data, along with the time-variant and time-invariant components from the PSM, SMD, and SWaT datasets, are presented from left to right. The red markers represent anomalies.

represent abnormal time steps. To provide a clearer comparison with the original data, the displayed results are the outcomes of the first Fourier Filter in SDA. As shown in Fig. 9, our Fourier Filter robustly extracts the time-invariant and time-variant components. The abnormal pattern becomes more distinctive after decomposition. Compared to the normal instances, abnormal patterns exhibit a significantly greater degree of irregularity, making them challenging to reconstruct. Besides, owing to the constraints of entropy regularization, abnormal steps encounter difficulty in establishing correlations with normal steps within the time-variant component, leading to higher anomaly scores. Therefore, our approach demonstrates its effectiveness in isolating abnormal patterns, which are characterized by increased irregularity and reduced correlation with normal instances. This spectral decomposition module provides a clearer understanding of the underlying structures in the data, enabling enhanced anomaly detection and interpretation.

5. Conclusion

This paper proposes a novel unsupervised time series anomaly detection algorithm, named SDA. We design a spectral decomposition model and a test-time adaptation method to deal with the non-stationary data in the real world. Our model decomposes time series into time-variant and time-invariant components to capture the underlying temporal patterns for reconstruction. Besides, we introduce an entropy regularization to encourage the establishment of relationships between normal time steps. To address the distribution gap between training and inference, our adaptation method enhances the distinction between normal and abnormal instances during inference. Furthermore, our algorithm has been extensively evaluated on various real-world datasets, demonstrating superior and effective performance in detecting anomalies. As our TTA method is currently offline, we plan to explore online options in the future. In summary, the proposed method exhibits robustness in identifying anomalies in the presence of complex and evolving patterns, making it a valuable tool for practical applications where non-stationary data is prevalent.

CRediT authorship contribution statement

Huanyu Zhang: Methodology. **Yi-Fan Zhang:** Supervision. **Jian Liang:** Writing – review & editing. **Zhang Zhang:** Supervision. **Liang Wang:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

[1] C. Dong, W. Xu, F. Zhang, Q. Hua, Y. Zhang, Meai-net: Multiview embedding and attention interaction for multivariate time series forecasting, Neurocomputing 633 (2025) 129769.

[2] C. Salazar, A. G. Banerjee, A distance correlation-based approach to characterize the effectiveness of recurrent neural networks for time series forecasting, Neurocomputing 629 (2025) 129641.

[3] H. Zhang, C. Xu, Y.-F. Zhang, Z. Zhang, L. Wang, J. Bian, T. Tan, Timeraf: Retrieval-augmented foundation model for zero-shot time series forecasting, arXiv preprint arXiv:2412.20810 (2024).

[4] H. Zhang, Y.-F. Zhang, Z. Zhang, Q. Wen, L. Wang, Logora: Local-global representation alignment for robust time series classification, IEEE Transactions on Knowledge and Data Engineering 36 (12) (2024) 8718–8729. doi:https://doi.org/10.1109/TKDE.2024.3459908.

[5] M. Du, Y. Wei, X. Zheng, C. Ji, Causal and local correlations based network for multivariate time series classification, Neurocomputing 634 (2025) 129884.

[6] H. Lyu, D. Huang, S. Li, W. W.Y. Ng, Q. Ma, Multiscale echo self-attention memory network for multivariate time series classification, Neurocomputing 520 (2023) 60–72.

[7] K. Wang, J. Kong, M. Zhang, M. Jiang, T. Liu, Mad-dgtd: Multivariate time series anomaly detection based on dynamic graph structure learning with time delay, Neurocomputing 635 (2025) 129887.

[8] J. Fan, Z. Liu, H. Wu, J. Wu, Z. Si, P. Hao, T. H. Luan, Luad: A lightweight unsupervised anomaly detection scheme for multivariate time series data, Neurocomputing 557 (2023) 126644.

[9] A. Blázquez-García, A. Conde, U. Mori, J. A. Lozano, A review on outlier/anomaly detection in time series data, ACM Computing Surveys (CSUR) 54 (3) (2021) 1–33.

[10] M. Bawaneh, V. Simon, Anomaly detection in smart city traffic based on time series analysis, in: 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, 2019, pp. 1–6.

[11] T. Chen, et al, Anomaly detection in semiconductor manufacturing through time series forecasting using neural networks, Ph.D. thesis, Massachusetts Institute of Technology (2018).

[12] K. Choi, J. Yi, C. Park, S. Yoon, Deep learning for anomaly detection in time-series data: review, analysis, and guidelines, IEEE Access 9 (2021) 120043–120065.

[13] S. Crépey, N. Lehdili, N. Madhar, M. Thomas, Anomaly detection in financial time series by principal component analysis and neural networks, Algorithms 15 (10) (2022) 385.

[14] J. Pereira, M. Silveira, Learning representations from healthcare time series data for unsupervised anomaly detection, in: 2019 IEEE international conference on big data and smart computing (BigComp), IEEE, 2019, pp. 1–7.

[15] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE 109 (5) (2021) 756–795.

[16] R. Gao, J. Wang, Y. Yu, J. Wu, L. Zhang, Enhanced graph diffusion learning with dynamic transformer for anomaly detection in multivariate time series, Neurocomputing 619 (2025) 129168.

[17] M. Wen, Z. Chen, Y. Xiong, Y. Zhang, Lgat: A novel model for multivariate time series anomaly detection with improved anomaly transformer and learning graph structures, Neurocomputing 617 (2025) 129024.

[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[19] B. Liu, Y. Xiao, L. Cao, Z. Hao, F. Deng, Svdd-based outlier detection on uncertain data, Knowledge and information systems 34 (2013) 597–618.

[20] F.-K. Sun, C. Lang, D. Boning, Adjusting for autocorrelated errors in neural networks for time series, in: Proc. NeuralPS, Vol. 34, 2021, pp. 29806–29819.

[21] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, in: Proc. AAAI, Vol. 35, 2021, pp. 4027–4035.

[22] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proc. SIGKDD, 2018, pp. 387–395.

[23] N. Ding, H. Ma, H. Gao, Y. Ma, G. Tan, Real-time anomaly detection based on long short-term memory and gaussian mixture model, Computers & Electrical Engineering 79 (2019) 106458.

[24] C. Zhang, T. Zhou, Q. Wen, L. Sun, Tfad: A decomposition time series anomaly detection architecture with time-frequency analysis, in: Proc. CIKM, 2022, pp. 2497–2507.

[25] S. Tuli, G. Casale, N. R. Jennings, Tranad: deep transformer networks for anomaly detection in multivariate time series data, Proceedings of the VLDB Endowment 15 (6) (2022) 1201–1214.

[26] C.-Y. A. Lai, F.-K. Sun, Z. Gao, J. H. Lang, D. Boning, Nominality score conditioned time series anomaly detection by point/sequential reconstruction, Advances in Neural Information Processing Systems 36 (2024).

[27] C. Wang, Z. Zhuang, Q. Qi, J. Wang, X. Wang, H. Sun, J. Liao, Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection, in: Proc. NeurIPS, 2023.

[28] D. S. Broomhead, R. Jones, Time-series analysis, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 423 (1864) (1989) 103–121.

[29] Y. Liu, C. Li, J. Wang, M. Long, Koopa: Learning non-stationary time series dynamics with koopman predictors, in: Proc. NerualPS, 2023.

[30] D. Kim, S. Park, J. Choo, When model meets new normals: Test-time adaptation for unsupervised time-series anomaly detection, in: Proc. AAAI, 2024.

[31] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proc. SIGKDD, 2019, pp. 2828–2837.

[32] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proc. SIGKDD, 2020, pp. 3395–3404.

[33] J. Xu, H. Wu, J. Wang, M. Long, Anomaly transformer: Time series anomaly detection with association discrepancy, in: Proc. ICLR, 2021.

[34] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[35] Z. Cheng, C. Zou, J. Dong, Outlier detection using isolation forest and local outlier factor, in: Proceedings of the conference on research in adaptive and convergent systems, 2019, pp. 161–168.

[36] H. Xu, G. Pang, Y. Wang, Y. Wang, Deep isolation forest for anomaly detection, IEEE Transactions on Knowledge and Data Engineering 35 (12) (2023) 12591–12604.

[37] A. Koran, H. Hojjati, N. Armanfard, Unveiling the flaws: A critical analysis of initialization effect on time series anomaly detection, arXiv preprint arXiv:2408.06620 (2024).

[38] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, N. Takata, A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction, IEEE Transactions on Aerospace and Electronic Systems 53 (3) (2017) 1384–1401.

[39] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, Multivariate time-series anomaly detection via graph attention network, 2020 IEEE International Conference on Data Mining (ICDM) (2020) 841–850.

[40] Y. Yang, C. Zhang, T. Zhou, Q. Wen, L. Sun, 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, 2023, pp. 3033–3045.

[41] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, G. Pang, Calibrated one-class classification for unsupervised time series anomaly detection, IEEE Transactions on Knowledge and Data Engineering 36 (11) (2024) 5723–5736.

[42] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, M. Hardt, Test-time training with self-supervision for generalization under distribution shifts, in: Proc. ICML, 2020, pp. 9229–9248.

[43] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, T. Darrell, Tent: Fully test-time adaptation by entropy minimization, in: Proc. ICLR, 2020.

[44] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, in: Proc. ICML, PMLR, 2020, pp. 6028–6039.

[45] Q. Wang, O. Fink, L. Van Gool, D. Dai, Continual test-time domain adaptation, in: Proc. CVPR, 2022, pp. 7201–7211.

[46] M. Boudiaf, R. Mueller, I. Ben Ayed, L. Bertinetto, Parameter-free online test-time adaptation, in: Proc. CVPR, 2022, pp. 8344–8353.

[47] H. Kingetsu, K. Kobayashi, Y. Okawa, Y. Yokota, K. Nakazawa, Multi-step test-time adaptation with entropy minimization and pseudo-labeling, in: Proc. ICIP, IEEE, 2022, pp. 4153–4157.

[48] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, in: International Conference on Learning Representations, 2019.

[49] A. Abdulaal, Z. Liu, T. Lancewicki, Practical approach to asynchronous multivariate time series anomaly detection and localization, in: Proc. SIGKDD, 2021, pp. 2485–2494.

[50] J. Goh, S. Adepu, K. N. Junejo, A. Mathur, 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, Springer, 2017, pp. 88–99.

[51] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, The ucr time series archive, IEEE/CAA Journal of Automatica Sinica 6 (6) (2019) 1293–1305.

[52] L. Shen, Z. Li, J. Kwok, Timeseries anomaly detection using temporal hierarchical one-class network, in: Proc. NeurIPS, 2020, pp. 13016–13026.

[53] S. Kim, K. Choi, H.-S. Choi, B. Lee, S. Yoon, Towards a rigorous evaluation of time-series anomaly detection, in: Proc. AAAI, 2022, pp. 7194–7201.

[54] A. Huet, J. M. Navarro, D. Rossi, Local evaluation of time series anomaly detection algorithms, in: Proc. SIGKDD, 2022, pp. 635–645.

[55] Z. Li, Y. Zhao, N. Botta, C. Ionescu, X. Hu, Copod: Copula-based outlier detection, 2020 IEEE International Conference on Data Mining (ICDM) (2020) 1118–1123. URL:https://api.semanticscholar.org/CorpusID:221819414

[56] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, G. Chen, Ecod: Unsupervised outlier detection using empirical cumulative distribution functions, IEEE Transactions on Knowledge and Data Engineering (2022).

[57] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural computation 13 (7) (2001) 1443–1471.

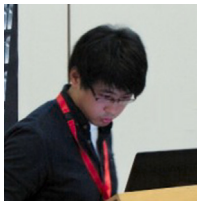
[58] T. Pevný, Loda: Lightweight on-line detector of anomalies, Machine Learning 102 (2016) 275–304.

[59] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern recognition letters 24 (9-10) (2003) 1641–1650.
[60] M. Goldstein, A. R. Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, 2012.
[61] H. Tian, H. Kong, S. Lu, K. Li, Unsupervised anomaly detection of multivariate time series based on multi-standard fusion, Neurocomputing 611 (2025) 128634.
[62] D. P. Kingma, M. Welling, Auto-encoding variational bayes, CoRR abs/1312.6114 (2013).
[63] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Proc. ICLR, 2018.
[64] T. Kieu, B. Yang, C. S. Jensen, Outlier detection for multidimensional time series using deep neural networks, 2018 19th IEEE International Conference on Mobile Data Management (MDM) (2018) 125–134.
[65] Y. Li, W. Chen, B. Chen, D. Wang, L. Tian, M. Zhou, Prototype-oriented unsupervised anomaly detection for multivariate time series, in: International Conference on Machine Learning, PMLR, 2023.
[66] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, et al, Stl: A seasonal-trend decomposition, J. Off. Stat 6 (1) (1990) 3–73.

Author biography

Huanyu Zhang is currently pursuing his Ph.D. degree of Computer Science at the New Laboratory of Pattern Recognition (NLPR), State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests mainly include time series analysis.

Yifan Zhang is currently pursuing his Ph.D. degree of Computer Science at the New Laboratory of Pattern Recognition (NLPR), State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests mainly include robust and reliable machine learning (ML) systems.



Jian Liang received the PhD degree from National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA) in 2019. From 2019 to 2021, he was a research fellow at the at the Vision and Learning Group, National University of Singapore (NUS). In June 2021, he joined the NLPR, Institute of Automation, Chinese Academy of Sciences(CASIA). Now, he is an Associate Professor at the New Laboratory of Pattern Recognition (NLPR), CASIA.

Zhang Zhang received the PhD degree from National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA) in 2009. From 2009 to 2010, he was a research fellow at the School of Computer Science and Engineering, Nanyang Technological University (NTU). In September 2010, he joined the NLPR, Institute of Automation, Chinese Academy of Sciences(CASIA). Now, he is an Associate Professor at the New Laboratory of Pattern Recognition (NLPR), CASIA.

Liang Wang received both the BEng and MEng degrees from Anhui University in 1997 and 2000, respectively, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA) in 2004. From 2004 to 2010, he was a research assistant at Imperial College London, United Kingdom, and Monash University, Australia, a research fellow at the University of Melbourne, Australia, and a lecturer at the University of Bath, United Kingdom, respectively. Currently, he is a full professor of the Hundred Talents Program at the State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA. He has widely published in highly ranked international journals such as IEEE TPAMI and IEEE TIP, and leading international conferences such as CVPR, ICCV, and ECCV. He has served as an Associate Editor of IEEE TPAMI, IEEE TIP, and PR. He is an IEEE Fellow and an IAPR Fellow.