# C2FAD: A Coarse-to-Fine Anomaly Detection for Time Series


1st Wei Xia
*Chongqing Normal University*
Chongqing, China
xiaweibb@foxmail.com

2nd Chao Wang*
*Chongqing Normal University*
Chongqing, China
chaosimpler@gmail.com

3rd Songbo Su
*Chongqing Normal University*
Chongqing, China
1937828330@qq.com

4th Haoyue Zheng
*Chongqing Normal University*
Chongqing, China
1113672940@qq.com



***Abstract*—Deep learning-based methods dominate time series anomaly detection. However, these methods typically rely on a single detection paradigm, which struggle to achieve an optimal balance between macro trends and micro fluctuations. To address this issue, this paper simulates a coarse-to-fine analysis logic and proposes a novel coarse-to-fine time series anomaly detection model (C2FAD). The first stage of this model employs a coarse-grained anomaly collector, CgAC, which efficiently filters out significant and coarse-grained macroscopic anomalies by capturing long-term dependencies and global context between sequences. In the second stage, we designed a fine-grained anomaly collector, FgAC, which extracts microscopic morphological features within local windows, deeply analyzes the local context, and identifies complex fine-grained anomalies that are not obvious in the global view but are incompatible with neighboring patterns. We conducted comparative experiments on seven datasets with ten mainstream anomaly detection methods and the proposed C2FAD. Experiments show that compared to the best baseline model, C2FAD improves the AUC score by 6.14%, F1 score by 9.02%, and AFF-F1 score by 3.81%.**



***Index Terms*—time series, anomaly detection, Coarse-grained Anomaly,Fine-grained Anomaly**


## I. Introduction

Time series anomaly detection is a pivotal task in fields such as industrial monitoring[1], financial risk control[2], and healthcare[3]. The objective of the system is to accurately identify abnormal points or segments that deviate from normal patterns, thereby enabling timely intervention to minimize risks and losses.

In recent years, deep learning-based approaches—including reconstruction-based[4, 5], prediction-based[6], representation learning[7], and generative adversarial methods[8]-have achieved notable improvements over traditional techniques. However, the majority of extant methods rely on a single detection paradigm, which exhibits fundamental limitations when dealing with multi-scale and heterogeneous anomalies in real-world scenarios. Models that focus on global modeling[7, 9] frequently overlook subtle local anomalies, while those that emphasise local details[10, 11] can be misled by normal fluctuations, resulting in an increased number of false alarms. Whilst certain studies[12, 13] endeavour to capture both coarse- and fine-grained information, they generally employ parallel collection strategies. This can result in the production of overlapping representations, redundant information, and even information loss, thereby rendering anomaly filtering less effective.

In order to address these issues, this paper proposes C2FAD, a novel coarse-to-fine anomaly detection model. The system utilises a two-stage detection architecture, comprising a coarse-grained anomaly collector (CgAC) that employs temporal attention to capture global dependencies and expeditiously identify broad anomalous regions. Subsequently, a fine-grained anomaly collector (FgAC) employs a shallow temporal convolutional network (TCN) with a narrow receptive field to analyse local micro-features, thereby detecting anomalies that are obscured from a global perspective. In conclusion, the Reconstruction Series (RS) network is employed to reconstruct the input from the anomaly-enhanced representation and detects anomalies based on the reconstruction error. The CgAC and FgAC modules function in a complementary, role-defined manner, forming a two-level anomaly collection framework.

To validate the effectiveness of C2FAD, systematic experiments were conducted, comparing it with nine mainstream baselines across seven public datasets. The results demonstrate that C2FAD outperforms state-of-the-art models in terms of the AUC, F1 and AFF-F1 metrics, thereby proving the effectiveness of its two-stage framework for complex, multi-scale anomaly detection. Further confirmation of the contribution of each module is provided by ablation studies, offering valuable insights for the design of future hierarchical anomaly detection models.

The main contributions of this study are as follows:

- Starting from cognitive mechanisms, this study proposes for the first time a hierarchical detection approach of "from coarse to fine, multi-layered progression," simulating the human analytical logic of "from the whole to the part" when observing complex systems.
- This study introduces a CgAC and a FgAC to capture long-term dependencies, trend anomalies, and local mutation features across time-series data in stages, enhancing sensitivity to transient anomalies and pattern shifts.
- This study conducted comparative and ablation experiments on seven publicly available benchmark datasets. The results show that C2FAD significantly outperforms existing mainstream methods in AUC, F1, and AFF-F1 metrics. Compared to the best baseline model, C2FAD

improves AUC, F1, and AFF-F1 metrics by 6.14%, 9.02%, and 3.81%, respectively.

## II. Related Work

In the contemporary field of data science, the accurate identification of anomalies within voluminous datasets has become a critical challenge. Time series anomaly detection methods are typically categorised into three types: traditional statistical methods, machine learning methods, and deep learning methods. Whilst traditional and machine learning approaches frequently underperform in the capture of deep temporal features, thus limiting their effectiveness in complex anomaly cases[14], deep learning has been shown to excel in this regard. It explicitly models explicitly incorporates complex temporal and feature dependencies in multivariate time series, enhancing interpretability by identifying anomalous variables, thus establishing itself as the prevailing and preeminent approach[9].

Notwithstanding its prevalence, deep learning for anomaly detection is confronted with considerable challenges. In order to model complex dependencies, models such as OmniAnomaly[7] combine GRU and VAE to learn normal patterns and use reconstruction probability for detection. However, due to their rarity, establishing a complete sequence of events is challenging. Anomaly_Transformer[15] addresses this issue by reconstructing series based on normal point patterns. Further challenges are presented by the suboptimal adaptation of contrastive learning from images. CARLA[16] addresses this issue through the implementation of a time-series-specific contrastive method, which involves the incorporation of prior anomaly knowledge into negative samples. SimAD[17] enhances the extraction of contextual features and the integration of normal behaviour, thus addressing the limitation of temporal context.

A fundamental objective of research in this field is to enhance the computational efficiency of the system while maintaining the accuracy of the results. Lightweight frameworks, exemplified by TranAD[6], have been shown to achieve robust feature extraction through the utilisation of adaptive focal scores and adversarial training. DTAAD[12] employs a single-layer Transformer with scaling and feedback mechanisms to reduce costs while enhancing accuracy.

However, extant methods predominantly rely on a single detection paradigm and continue to exhibit limitations when confronted with multi-scale and heterogeneous anomaly patterns in real-world scenarios. Despite the fact that certain research methodologies[12, 13] have sought to concurrently consider both coarse-grained and fine-grained anomaly information in time series, their particular implementations frequently employ parallel collection strategies for these two information types. This strategy is subject to certain limitations in practical applications. The lack of coordination and differentiation between coarse-grained and fine-grained information during the collection process results in significant overlap between their representations.

In order to address the shortcomings of existing research in this area, the present paper proposes a novel coarse-to-fine time series anomaly detection model, C2FAD. The model utilises a two-stage detection architecture. Initially, it rapidly identifies coarse-grained anomalous regions through global perception. Subsequently, it focuses on local fine-grained features, thereby achieving accurate anomaly detection through sequence reconstruction error.

## III. Method

### A. Overview

In this paper, our training dataset consists of sensor data at one or more training timestamps, denoted as $X_{train} = [x_{train}^1, ..., x_{train}^{T_{train}}]$, which is used to train C2FAD. The aim is to detect anomalies in the test data at a separate set of test timestamps, $T_{test}$, with the test data denoted as $X_{test} = [x_{test}^1, ..., x_{test}^{T_{test}}]$.

C2FAD outputs a set of binary labels $Y(t) \in \{0, 1\}$, where $Y(t) = 1$ indicates that time $t$ is an anomaly timestamp.

Figure 1 shows the overall structure of C2FAD, which consists of four main parts:

- **Data Preprocessor**: Ensures time series continuity through imputation and applies min-max normalization to accelerate convergence and enhance training stability.
- **Coarse-grained Anomaly Collector (CgAC)**: An attention-based encoder that captures significant anomaly patterns from a global, macroscopic perspective.
- **Fine-grained Anomaly Collector (FgAC)**: A three-layer TCN that acts as a "magnifying glass" to perform detailed, local analysis around each timestamp.
- **Reconstruction Series (RS)**: Integrates and refines the anomaly information from both collectors to generate the final reconstructed series $\widehat{X}$.

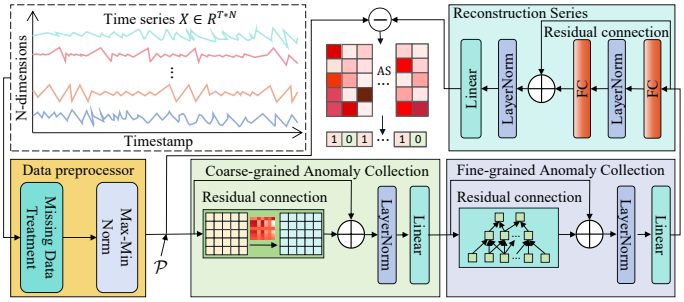


Fig. 1: The overall architecture of the proposed C2FAD model.

### B. Data Preprocessor

The Data Preprocessor consists of two parts: *Handling Missing Values* and *Min-Max Normalization*. For missing values, the model finds the $K$ nearest complete samples to the missing value in the sequence and fills the missing value with the average of these $K$ complete samples. Second, C2FAD standardizes a time series $X$ using the maximum and minimum values from the training data:

$$\widetilde{x} = \frac{x - \min(X_{train})}{\max(X_{train}) - \min(X_{train}) + \tau} \tag{1}$$

Where $\max(X_{train})$ and $\min(X_{train})$ are the maximum and minimum values of the training set, respectively, and $\tau$ represents a very small value to prevent the denominator from being zero.

### C. Coarse-grained Anomaly Collector

The CgAC of C2FAD is a coarse-grained anomaly detector based on an attention mechanism. It is designed to capture long-term dependencies and overall trend patterns in time series, thereby filtering out significant sequences of coarse-grained anomaly candidates.Since the CgAD network itself does not have temporal awareness capabilities, we add absolute positional encoding $\mathcal{P}$ based on trigonometric functions to each sequence to ensure the uniqueness and continuity of positional representations between sentences of different lengths. The absolute positional encoding $\mathcal{P}$ is added to the pre-processed time series and input into the CgAC network for the first anomaly detection, resulting in $X^C$:

$$\begin{aligned} X^C &= \mathrm{CgAC}(\mathrm{M}) \\ \mathrm{M} &= \widetilde{X} + \mathcal{P} \end{aligned} \tag{2}$$

We now provide details about the workings of the CgAC network. First, CgAC projects the series $\mathrm{M}$ through a fully connected layer to fuse features from different time steps, generating a global contextual representation $f_{CgAC}$:

$$f_{CgAC} = \mathrm{M}W + b \tag{3}$$

Here, $W \in \mathbb{R}^{N*d}$ and $b \in \mathbb{R}^{=1*d}$ are learnable weights. Subsequently, we calculate the state $S_t$ of the sequence at the current timestamp $t$ in the feature space, the coarse-grained temporal state $S_a$, and the anomalous information content $S_i$ contained in the sequence. This allows the model to dynamically reference and integrate information from different timestamps throughout the entire sequence during subsequent encoding:

$$S_t, S_a, S_i = f_{CgAC}W^t, f_{CgAC}W^a, f_{CgAC}W^i \tag{4}$$

Here,$S^t \in \mathbb{R}^{T*d}$, $S^a \in \mathbb{R}^{T*d}$, and $S^i \in \mathbb{R}^{T*d}$. Then, we calculate the correlation score $A_S$ using state $S_t$ and state $S_a$:

$$A_S = \sigma(\frac{S_t(S_a)^\top}{\sqrt{d}}) \tag{5}$$

Here, $\sigma(\cdot)$ is the activation function, set to the $Soft\max(\cdot)$ function. Then, we compute the time series representation $T_S$, which contains coarse-grained anomaly information:

$$T_S = A_S S_i \tag{6}$$

To make the network easier to train, CgAC represents time series data by using residual connections to build an "identity mapping" prior for sequence $\mathrm{M}$. Since residual connections can cause abnormal fluctuations in the distribution of time series data, CgAC applies LayerNorm to the result of the residual connection to ensure a stable distribution, providing a clean starting point for collecting fine-grained anomaly information. Finally, a fully connected layer is used to obtain the time series $X^C$ containing coarse-grained anomaly states:

$$X^C = FFC(\mathrm{LayerNorm}(T_S + \mathrm{M})) \tag{7}$$

Each scalar value in $X^C$ represents a preliminary confidence score indicating the presence of a coarse-grained macroscopic anomaly at the corresponding time step, and this output will serve as important guiding information for FgAC.

### D. Fine-grained Anomaly Collector

The coarse-grained anomaly information output by the CgAC module of the C2FAD model identifies macroscopic regions in the sequence that may contain anomalies. However, it should be noted that these regions may contain complex local normal fluctuations (false positives) or miss subtle anomalies that are only significant in a local context (false negatives). In order to achieve accurate anomaly detection, the FgAC guided by $X^C$ has been introduced.

As shown in Figure 2, FgAC stacks three dilated TCN blocks. This design forms an efficient multi-scale feature pyramid. The bottom-layer convolutions capture point-level instantaneous changes and noise, the middle-layer convolutions identify morphological anomalies within short time segments, such as irregular fluctuations or stagnation, and the top-layer convolutions supply contextual information over a broader time window to assess the relevance of a micro-event within a slightly longer context, aiding in distinguishing true local anomalies from normal transient fluctuations.

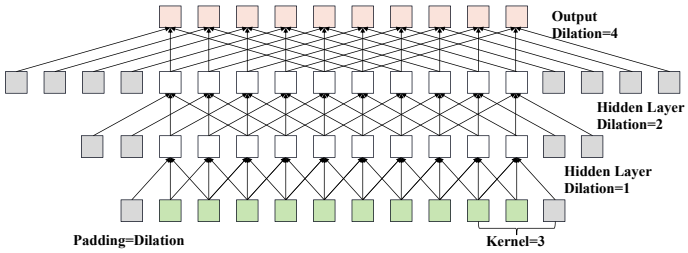


Fig. 2: The TCN architecture in FgAC. This structure consists of three TCN blocks with expansion factors set $d$ to 1, 2, and 4 respectively, and a kernel size of $K = 3$.

For the $l$-th dilated TCN block, given the input $\mathrm{H}^{l-1} \in \mathbb{R}^{T*N_{in}^l}$ to this layer (where $\mathrm{H}^0 = X^C$ when $l = 1$), its output $\mathrm{H}^l$ is:

$$\mathrm{H}^l = \mathrm{LayerNorm}(A^l(t,c) + \mathrm{H}^{l-1}) \tag{8}$$

For a TCN block with dilation factor $d^l$, kernel size $K$, weight matrix $W^l \in \mathbb{R}^{K*N_{in}\mathbb{R}^l*N_{out}^l}$, and bias $b^l \in \mathbb{R}^{N_{out}^l}$, the convolution result $\mathrm{Z}^l(t,c)$ at timestamp $t$ and output channel $c$ is given by:

$$\mathrm{Z}^l(t,c) = \sum_{i=0}^{K-1}\sum_{j=1}^{N_{in}^l} W^l(i,j,c)\cdot \mathrm{H}^{l-1}(t-d^l\cdot i, j) + b^l(c) \tag{9}$$

Then, the $ReLU(\cdot)$ activation function allows the entire network to fit complex nonlinear functions and determine

which anomalous information is significant and worth further processing, and which information can be ignored, thus helping to highlight significant local changes:

$$A^l(t,c) = ReLU(Norm(\mathrm{Z}^l(t,c))) \tag{10}$$

To facilitate stable training and information flow in deep networks, we introduce residual connections and layer normalization to obtain the time series $X^F$, which contains fine-grained anomaly information:

$$X^F = FFC(\mathrm{LayerNorm}(X^C + \mathrm{Z})) \tag{11}$$

### E. Reconstruction Series

The RS network is responsible for generating the final reconstructed sequence $\widehat{X}$ based on the fine-grained anomaly state $X^F$. First, the RS network compresses the high-dimensional concatenated features into a lower-dimensional latent space to remove redundant information and capture common information across multi-scale features:

$$Z_1 = ReLU(X^F W_1 + b_1) \tag{12}$$

Here, $W_1 \in \mathbb{R}^{N*d_1}$, $b_1 \in \mathbb{R}^{1*d_1}$, and $d_1 < D$. Subsequently, the RS network performs the first layer normalization. Then, the bottleneck features are expanded back to the same dimension as the input, preparing for residual fusion:

$$Z_2 = ReLU(LayerNorm(Z_1)W_2 + b_2) \tag{13}$$

Here, $W_2 \in \mathbb{R}^{d_1*N}$ and $b_2 \in \mathbb{R}^{1*N}$. Then, the RS network uses a residual connection to add the decompressed feature correction term $Z_2$ to $X^F$ and normalizes the residual output. Finally, a linear layer maps the result back to the original data space to provide the basis for the final anomaly score:

$$\widehat{X} = \mathrm{Linear}(\mathrm{LayerNorm}(Z_2 + X^F) \tag{14}$$

### F. Training Procedure

The C2FAD model uses mean squared error (MSE) as the training loss function, aiming to minimize the reconstruction error of the model on normal time series segments. Further experimental settings will be described in section IV. The overall loss function of the model is as follows:

$$L = \frac{\sum_{t=1}^{T}\sum_{n=1}^{N}\left(\widehat{x}_{tn} - \widetilde{x}\right)^2}{T * N} \tag{15}$$

Here, $T$ represents the timestamp, $N$ represents the data dimension and $\widehat{x}_{tn}$ represents the reconstructed data from the $n$-th sensor of the multivariate time series at timestamp $t$. The ultimate goal of training is to find a set of parameters $\{\Theta, \theta, \Phi\}$ (where $\Theta, \theta$ and $\Phi$ includes the network parameters of CgAC, FgAC, and RS) that minimizes the overall loss of C2FAD:

$$\{\Theta, \theta, \Phi\} = \arg\min_{\Theta,\theta,\Phi,\lambda} L \tag{16}$$

### G. Anomaly Detection

To detect whether the data at a specific timestamp is abnormal, C2FAD calculates an anomaly score for the current timestamp based on the collected anomaly information. The anomaly score $As_t$ of a time series at timestamp $t$ is defined as:

$$As_t = \sum_{n=0}^{N}\left(\widehat{x}_{tn} - \widetilde{x}_{tn}\right)^2 \tag{17}$$

Here, $\widetilde{x}_{tn}$ represents the preprocessed data, and $\widehat{x}_{tn}$ represents the reconstructed data. When the anomaly score $As_t$ at timestamp $t$ exceeds the threshold $\varepsilon$, we consider timestamp $t$ to be an anomalous timestamp. We sort all anomaly scores in the test set from largest to smallest, and the anomaly score ranked $\gamma * Ar$ is used as the threshold $\varepsilon$, where $Ar$ is the anomaly rate of the test set:

$$\varepsilon = Quantile(As, \gamma * Ar) \tag{18}$$

Therefore, the anomaly detection result $y_t$ for $x_t$ is defined as:

$$y_t = 1 \quad if \quad As_t > \varepsilon \quad else \quad 0 \tag{19}$$

## IV. Experiments

### A. Experimental Settings

*1) Datasets:* The study utilised seven widely employed time series anomaly detection datasets: **MSL and SMAP**[18] from NASA spacecraft sensor telemetry; **MBA**[19] with 48 sets of dual-channel electrocardiogram records; **SMD**[7] with operating metrics from 28 servers; **MSDS**[20] that integrates distributed tracking, application logs and performance metrics; **NAB**[21] streaming detection benchmark covering real and synthetic time series; **PSM**[22] from eBay application server nodes. Table I summarizes the detailed characteristics of these datasets.

TABLE I: Dataset statistics

| | Dimensions | Train | Test | Anomalies(%) |
|---|---|---|---|---|
| MSL | 55 | 58317 | 73729 | 10.72 |
| SMAP | 25 | 135183 | 427617 | 13.13 |
| MBA | 2 | 100000 | 100000 | 0.14 |
| SMD | 38 | 708405 | 708420 | 4.16 |
| MSDS | 10 | 146430 | 146430 | 5.37 |
| NAB | 1 | 4033 | 4033 | 0.92 |
| PSM | 25 | 132481 | 87841 | 27.8 |

*2) Evaluation Metrics:* To systematically evaluate the detection capability of the proposed C2FAD model, we employ a multi-metric comprising Precision (P), Recall (R), F1-score, and the area under the ROC curve (ROC-AUC). While these metrics provide fundamental performance assessment, recent research [23, 24] has critically demonstrated their inherent limitations in capturing temporal dependencies and anomaly segment characteristics in time series data. To address these methodological gaps, we further incorporate the Affiliation metrics(includes Aff-Pre, Aff-Rec and Aff-F1) [25].

*3) Comparison Models:* We benchmarked our proposed C2FAD model against 10 state-of-the-art deep neural network-based methods: **OmniAnomaly** [7] employs a variational autoencoder for low-dimensional reconstruction; **MTAD_GAT** [9] combines graph attention networks with temporal modeling; **GDN** [26] utilizes graph neural networks for structural pattern reconstruction; **CAE-M** [27] combines a convolutional autoencoder with multi-scale error metrics; **MAD-GAN** [8] uses adversarial training for distribution learning; **USAD** [4] implements an adversarial stacked autoencoder; **DAGMM** [5] fuses an autoencoder with a Gaussian mixture model; **TranAD** [6] employs Transformer-based long-range dependency modeling; **DTAAD** [12] combines Transformer with a dual temporal convolutional network and scaled reconstruction; and **Dcdetector** [10]is a multi-scale dual-attention contrastive representation learning model for anomaly detection.

### *B. Overall Performance*

Table II shows the AUC, Precision, Recall, F1, Aff-Pre, Aff-Rec, and Aff-F1 scores of the proposed C2FAD model and other comparison models on the seven datasets. The experimental results show that our method performs excellently on most datasets and key metrics.

Specifically, C2FAD significantly outperforms all baseline models on the MSL, SMAP, SMD, MSDS, and NAB datasets. Compared to the best baseline model on these dataset, C2FAD improved the F1 score by 25.10% (0.2797→0.3499), 3.75% (0.6832→0.7088), 2.65% (0.6045→0.6205), 0.77% (0.9749→0.9824), and 18.13% (0.2775→0.3278), respectively. On datasets with high anomaly rates (PSM, 27.8%), the C2FAD model tends to improve the AUC score while slightly reducing F1 and Precision scores. Conversely, on datasets with low anomaly rates (MBA, 0.14%), the model tends to improve Precision and F1 scores while slightly lowering the AUC score.

C2FAD also demonstrated significant advantages in the recently proposed Affiliation metric, achieving near-perfect Aff-Rec scores across seven datasets. While maintaining high Aff-Rec scores, C2FAD also maintained competitive Aff-Pre scores, achieving the best results on the MSL (0.6130), SMAP (0.6804), MBA (0.6123), and SMD (0.8060) datasets. Exception NAB dataset, where C2FAD's Aff-F1 score was 0.8604, which is marginally lower than the optimal baseline model MTAD_GAT's 0.8617.

Figure 3 shows the average performance of all methods, and our model achieved the best results across all seven evaluation metrics on the seven experimental datasets. Specifically, compared to the best baseline model, our model achieved improvements of 6.14% (0.6937→0.7363) in AUC, 9.02% (0.5188→0.5656) in F1 score, and 3.81% (0.7654→0.7946) in AFF-F1 score.

### *C. Ablation Experiment*

In this section, we demonstrate the effectiveness of each component of the C2FAD model by comparing AUC, F1, and Aff-F1 metrics. We investigate the impact of each module in our model on anomaly detection performance by disabling the

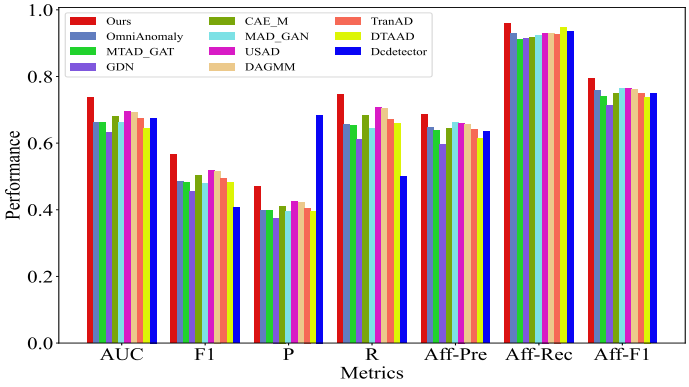


Fig. 3: Average Performance Across All Datasets

CgAC (denoted as w/o CgAC), FgAC (denoted as w/o FgAC), and RS (denoted as w/o RS) modules one by one. From the experimental results in Table III, it is clear that the complete architecture of the C2FAD model achieves the best AUC, F1, and Aff-F1 metrics across all datasets, while removing or replacing any module of the model leads to performance degradation.

Furthermore, none of the CgAC, FgAC, and RS modules in C2FAD holds an absolutely central position. For different datasets, the contributions of CgAC, FgAC, and RS to the anomaly detection results vary, but they are all equally important. This demonstrates the necessity and synergistic effectiveness of each component in the model.

### *D. Anomaly detection performance analysis*

To verify the effectiveness of the two-stage architecture in C2FAD, we selected typical anomaly segments from MSDS, MBA, and SMD datasets for case analysis. We compared against baselines: OmniAnomaly (coarse-grained focus), Dcdetector (fine-grained focus), DTAAD (parallel coarse and fine-grained strategy), w/o CgAC and w/o FgAC.

As shown in Figure 4, the complete C2FAD achieved the best overall detection results across all datasets. OmniAnomaly excelled at macroscopic anomalies by learning overall data distributions; Dcdetector performed best on microscopic anomalies through its channel-independent, localized approach. DTAAD, using a parallel strategy to capture both granularities, outperformed the single-focus models.

Unlike single-granularity models (OmniAnomaly, Dcdetector), C2FAD systematically handles both anomaly region localization and pattern identification. Compared to the parallel strategy of DTAAD, C2FAD's two-stage design aligns better with cognitive logic: the coarse stage provides contextual guidance, allowing the fine stage to focus on suspected regions, improving both efficiency and accuracy. Thus, C2FAD combines the strengths of different baseline models and, through complementary two-stage collection, achieves more robust and comprehensive anomaly detection. This explains its superior performance in multiple experiments.

TABLE II: Experimental results comparison on seven public datasets with ten state-of-the-art methods. The best in **bold**, and the second in underlined.

| Method | MSL | | | | | | | SMAP | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 |
| Ours | **0.6416** | **0.2974** | **0.4248** | **0.3499** | **0.6130** | 0.9885 | **0.7567** | **0.7274** | **0.5521** | 0.9895 | **0.7088** | **0.6804** | **1.0000** | **0.8098** |
| OmniAnomaly | 0.5315 | 0.1663 | 0.3245 | 0.2199 | 0.5288 | 0.9873 | 0.6887 | 0.6745 | 0.4912 | 0.9811 | 0.6546 | 0.6515 | **1.0000** | 0.7890 |
| MTAD_GAT | 0.5842 | 0.2098 | 0.4195 | 0.2797 | 0.5813 | **0.9962** | 0.7342 | 0.6472 | 0.4741 | 0.9479 | 0.6321 | 0.6396 | 0.9999 | 0.7802 |
| GDN | 0.5452 | 0.1755 | 0.3513 | 0.2341 | 0.5302 | 0.9870 | 0.6898 | 0.6371 | 0.4699 | 0.9345 | 0.6254 | 0.6395 | 0.9999 | 0.7801 |
| CAE_M | 0.5453 | 0.1757 | 0.3514 | 0.2343 | 0.5316 | 0.9868 | 0.6910 | 0.6633 | 0.4844 | 0.9584 | 0.6435 | 0.6494 | **1.0000** | 0.7874 |
| MAD_GAN | 0.5218 | 0.1546 | 0.3098 | 0.2063 | 0.5234 | 0.9873 | 0.6841 | 0.7134 | 0.5221 | 0.9882 | 0.6832 | 0.6699 | **1.0000** | 0.8023 |
| USAD | 0.5460 | 0.1759 | 0.3514 | 0.2344 | 0.5318 | 0.9876 | 0.6913 | 0.6879 | 0.4989 | **0.9977** | 0.6652 | 0.6582 | **1.0000** | 0.7939 |
| DAGMM | 0.5099 | 0.1451 | 0.2913 | 0.1937 | 0.5112 | 0.9873 | 0.6736 | 0.6835 | 0.4962 | 0.9918 | 0.6615 | 0.6562 | **1.0000** | 0.7924 |
| TranAD | 0.5395 | 0.1712 | 0.3418 | 0.2281 | 0.5243 | 0.9872 | 0.6849 | 0.6768 | 0.4922 | 0.9819 | 0.6557 | 0.6543 | **1.0000** | 0.7910 |
| DTAAD | 0.5505 | 0.1818 | 0.3611 | 0.2418 | 0.5247 | 0.9883 | 0.6855 | 0.6312 | 0.4642 | 0.9361 | 0.6206 | 0.6333 | 0.9998 | 0.7754 |
| Dcdetector | 0.5402 | 0.1756 | 0.3546 | 0.2349 | 0.5195 | 0.9714 | 0.6770 | 0.6428 | 0.4564 | 0.9164 | 0.6093 | 0.6233 | 0.9999 | 0.7679 |

| Method | MBA | | | | | | | SMD | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 |
| Ours | 0.5723 | **0.3902** | **0.7804** | **0.5203** | **0.6123** | 0.9959 | **0.7584** | **0.9192** | **0.4833** | 0.8666 | **0.6205** | **0.8060** | 0.9948 | **0.8905** |
| OmniAnomaly | 0.5611 | 0.3786 | 0.7565 | 0.5046 | 0.6046 | 0.9902 | 0.7508 | 0.7053 | 0.2794 | 0.5592 | 0.3726 | 0.7525 | 0.9951 | 0.8570 |
| MTAD_GAT | 0.5451 | 0.3685 | 0.7365 | 0.4912 | 0.5995 | 0.9956 | 0.7484 | 0.7098 | 0.2841 | 0.5675 | 0.3786 | 0.6233 | 0.9946 | 0.7663 |
| GDN | 0.4559 | 0.3091 | 0.6185 | 0.4122 | 0.5316 | 0.9831 | 0.6901 | 0.6613 | 0.2402 | 0.4781 | 0.3198 | 0.5669 | 0.9867 | 0.7201 |
| CAE_M | 0.5268 | 0.3566 | 0.7123 | 0.4753 | 0.5832 | 0.9877 | 0.7334 | 0.7874 | 0.3548 | 0.7091 | 0.4730 | 0.6792 | 0.9992 | 0.8087 |
| MAD_GAN | 0.5578 | 0.3768 | 0.7535 | 0.5024 | 0.6023 | 0.9953 | 0.7505 | 0.6716 | 0.2493 | 0.4983 | 0.3323 | 0.7858 | 0.9893 | 0.8759 |
| USAD | 0.5564 | 0.3761 | 0.7515 | 0.5013 | 0.5976 | 0.9915 | 0.7457 | 0.8941 | 0.4511 | 0.9016 | 0.6013 | 0.7506 | 0.9994 | 0.8573 |
| DAGMM | **0.5726** | 0.3862 | 0.7719 | 0.5148 | 0.596 | **0.9965** | 0.7459 | 0.8969 | 0.4535 | **0.9063** | 0.6045 | 0.7328 | 0.9995 | 0.8456 |
| TranAD | 0.5518 | 0.3721 | 0.7435 | 0.496 | 0.5991 | 0.9911 | 0.7468 | 0.8261 | 0.3891 | 0.7782 | 0.5188 | 0.6306 | **0.9996** | 0.7733 |
| DTAAD | 0.4979 | 0.3371 | 0.6738 | 0.4494 | 0.5494 | 0.9808 | 0.7043 | 0.7598 | 0.3293 | 0.6585 | 0.4390 | 0.6848 | 0.9981 | 0.8123 |
| Dcdetector | 0.5350 | 0.3256 | 0.6614 | 0.4364 | 0.5355 | 0.9858 | 0.6940 | 0.8598 | 0.4253 | 0.8458 | 0.5660 | 0.7643 | 0.9989 | 0.8660 |

| Method | MSDS | | | | | | | NAB | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 |
| Ours | **0.9613** | **0.9774** | **0.9874** | **0.9824** | 0.7339 | **0.7509** | **0.7423** | **0.7097** | **0.2458** | **0.4917** | **0.3278** | 0.7561 | **0.9979** | 0.8604 |
| OmniAnomaly | 0.9457 | 0.9703 | 0.9699 | 0.9701 | 0.7066 | 0.6819 | 0.694 | 0.7031 | 0.2081 | 0.4164 | 0.2775 | 0.7132 | 0.9875 | 0.8282 |
| MTAD_GAT | 0.9514 | 0.9732 | 0.973 | 0.9731 | 0.7085 | 0.5241 | 0.6025 | 0.6412 | 0.1554 | 0.2916 | 0.2028 | **0.7592** | 0.9963 | **0.8617** |
| GDN | 0.9299 | 0.9613 | 0.9611 | 0.9612 | 0.6682 | 0.5951 | 0.6295 | 0.6829 | 0.1884 | 0.3755 | 0.2509 | 0.7423 | 0.9965 | 0.8508 |
| CAE_M | 0.9494 | 0.9723 | 0.9719 | 0.9721 | 0.7211 | 0.7222 | 0.7216 | 0.6827 | 0.1886 | 0.3753 | 0.2510 | 0.7396 | 0.9968 | 0.8492 |
| MAD_GAN | 0.9467 | 0.9711 | 0.9704 | 0.9707 | 0.6914 | 0.6816 | 0.6865 | 0.6824 | 0.1878 | 0.3759 | 0.2505 | 0.7451 | 0.9959 | 0.8524 |
| USAD | 0.9546 | 0.9749 | 0.9747 | 0.9748 | **0.7523** | 0.6852 | 0.7172 | 0.6822 | 0.1879 | 0.3754 | 0.2504 | 0.7222 | 0.9961 | 0.8373 |
| DAGMM | 0.9541 | 0.9747 | 0.9746 | 0.9746 | 0.7458 | 0.6853 | 0.7143 | 0.6831 | 0.1881 | 0.3761 | 0.2508 | 0.7276 | 0.9968 | 0.8412 |
| TranAD | 0.9533 | 0.9742 | 0.9741 | 0.9741 | 0.7178 | 0.6442 | 0.6790 | 0.6205 | 0.1251 | 0.2507 | 0.1669 | 0.7149 | 0.9951 | 0.8320 |
| DTAAD | 0.9406 | 0.9668 | 0.9666 | 0.9667 | 0.6452 | 0.6606 | 0.6528 | 0.6623 | 0.1668 | 0.3334 | 0.2224 | 0.7358 | 0.9961 | 0.8464 |
| Dcdetector | 0.9475 | 0.9716 | 0.9714 | 0.9715 | 0.6896 | 0.6952 | 0.6924 | 0.6413 | 0.1796 | 0.3631 | 0.2403 | 0.7362 | 0.9954 | 0.8464 |

| Method | PSM | | | | | | | Average | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 | AUC | P | R | F1 | Aff-Pre | Aff-Rec | Aff-F1 |
| Ours | **0.6224** | 0.3354 | 0.6831 | 0.4499 | 0.5974 | 0.9853 | **0.7438** | **0.7363** | **0.4688** | **0.7462** | **0.5656** | **0.6856** | **0.9590** | **0.7946** |
| OmniAnomaly | 0.5212 | 0.2929 | 0.5857 | 0.3905 | 0.5805 | 0.8611 | 0.6935 | 0.6632 | 0.3981 | 0.6562 | 0.4843 | 0.6482 | 0.9290 | 0.7573 |
| MTAD_GAT | 0.5529 | 0.3158 | 0.6304 | 0.4208 | 0.5576 | 0.8739 | 0.6808 | 0.6617 | 0.3973 | 0.6523 | 0.4826 | 0.6384 | 0.9115 | 0.7392 |
| GDN | 0.5072 | 0.2828 | 0.5656 | 0.3771 | 0.4963 | 0.8461 | 0.6256 | 0.6314 | 0.3753 | 0.6121 | 0.4544 | 0.5964 | 0.9135 | 0.7123 |
| CAE_M | 0.5957 | **0.3467** | **0.6933** | **0.4622** | 0.6011 | 0.7309 | 0.6597 | 0.6787 | 0.4113 | 0.6817 | 0.5016 | 0.6436 | 0.9177 | 0.7501 |
| MAD_GAN | 0.5336 | 0.3021 | 0.6032 | 0.4026 | 0.6192 | 0.8208 | 0.7059 | 0.6610 | 0.3948 | 0.6428 | 0.4783 | 0.6624 | 0.9243 | 0.7654 |
| USAD | 0.5348 | 0.3031 | 0.6051 | 0.4039 | 0.6105 | 0.8525 | 0.7115 | 0.6937 | 0.4240 | 0.7082 | 0.5188 | 0.6605 | 0.9303 | 0.7649 |
| DAGMM | 0.5455 | 0.3106 | 0.6211 | 0.4141 | 0.6135 | 0.8321 | 0.7063 | 0.6922 | 0.4221 | 0.7047 | 0.5163 | 0.6547 | 0.9282 | 0.7599 |
| TranAD | 0.5497 | 0.3138 | 0.6271 | 0.4183 | **0.6432** | 0.8743 | 0.7412 | 0.6740 | 0.4054 | 0.6710 | 0.4940 | 0.6406 | 0.9274 | 0.7497 |
| DTAAD | 0.4723 | 0.3109 | 0.6762 | 0.4260 | 0.5302 | **0.9976** | 0.6924 | 0.6449 | 0.3938 | 0.6580 | 0.4808 | 0.6148 | 0.9459 | 0.7384 |
| Dcdetector | 0.5612 | 0.3213 | 0.6784 | 0.4360 | 0.5816 | 0.8992 | 0.7063 | 0.6754 | 0.4079 | 0.6844 | 0.4992 | 0.6357 | 0.9351 | 0.7500 |

### *E. Sensitivity Analysis*

In the context of the C2FAD model, the batch size constitutes a pivotal hyperparameter. The CgAC and FgAC modules exhibit divergent sensitivities and requirements with regard to batch size, a consequence of their disparate functional objectives.

In CgAC, a large batch size can encompass longer continuous or segmented sequences, making it easier to identify coarse-grained anomalies that require a long time window to become apparent. In FgAC, a small batch size enhances sensitivity to local anomalies because noisy gradients can help the model capture subtle anomalies.

As demonstrated in Figure 5, the experimental performance of C2FAD on seven datasets with batch sizes of 32, 64, 128, 256, and 512 is shown. Through the implementation of comparative experiments on disparate datasets, it was ascertained that a batch size of 128 provides an optimal balance between the performance of the various modules in the model and attains the most favourable experimental results.

TABLE III: Ablation study on seven public datasets. The best in **bold**, and the second in underlined.

| Method | MSL | | | SMAP | | | MBA | | | SMD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | F1 | Aff-F1 | AUC | F1 | Aff-F1 | AUC | F1 | Aff-F1 | AUC | F1 | Aff-F1 |
| Ours | **0.6416** | **0.3499** | **0.7567** | **0.7274** | **0.7088** | **0.8098** | **0.5723** | **0.5203** | **0.7584** | **0.9192** | **0.6205** | **0.8905** |
| w/o CgAC | 0.5111 | 0.1957 | 0.6692 | 0.6840 | 0.6613 | 0.7899 | 0.3606 | 0.3285 | 0.6584 | 0.7055 | 0.3732 | 0.8588 |
| w/o FgAC | 0.5523 | 0.2430 | 0.6839 | 0.6730 | 0.6523 | 0.7863 | 0.4891 | 0.4418 | 0.7108 | 0.6871 | 0.3510 | 0.8512 |
| w/o RS | 0.5485 | 0.2387 | 0.6825 | 0.6764 | 0.6550 | 0.7867 | 0.4170 | 0.3782 | 0.6909 | 0.7125 | 0.3817 | 0.8584 |

| Method | MSDS | | | NAB | | | PSM | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | F1 | Aff-F1 | AUC | F1 | Aff-F1 | AUC | F1 | Aff-F1 | AUC | F1 | Aff-F1 |
| Ours | **0.9613** | **0.9824** | **0.7423** | **0.7097** | **0.3278** | **0.8604** | **0.6224** | **0.4499** | **0.7438** | **0.7363** | **0.5656** | **0.7946** |
| w/o CgAC | 0.9388 | 0.9660 | 0.6592 | 0.7036 | 0.2778 | 0.8450 | 0.5761 | 0.4434 | 0.6724 | 0.6400 | 0.4637 | 0.7361 |
| w/o FgAC | 0.9294 | 0.9608 | 0.6651 | 0.6407 | 0.1944 | 0.8183 | 0.5488 | 0.4171 | 0.6687 | 0.6458 | 0.4658 | 0.7406 |
| w/o RS | 0.9372 | 0.9652 | 0.6702 | 0.6716 | 0.2396 | 0.8308 | 0.5149 | 0.3844 | 0.7034 | 0.6397 | 0.4633 | 0.7461 |

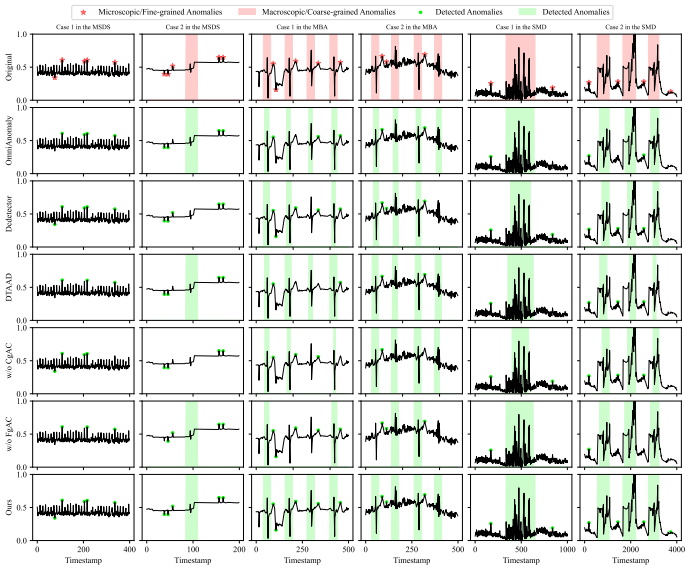


Fig. 4: Anomaly detection performance analysis. The red stars represent macroscopic/coarse-grained anomalies, the red squares represent microscopic/fine-grained anomalies, and the green markers represent anomalies detected by C2FAD or the comparison model.

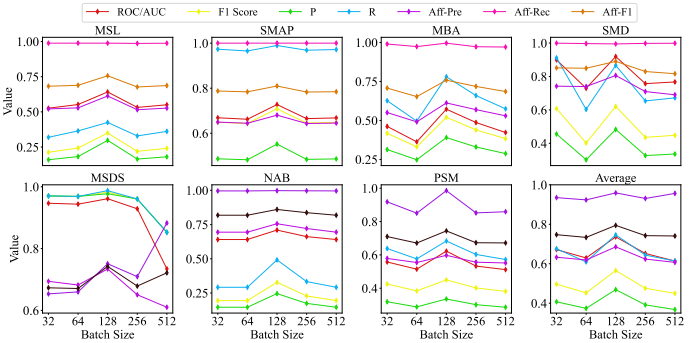


Fig. 5: Performance comparison of C2FAD at different batch sizes.

## V. Conclusions

This paper addresses the limitation of single-paradigm models in handling multi-scale anomaly patterns for time series anomaly detection. It introduces a novel hierarchical detection model called C2FAD, offering a scalable and robust solution. C2FAD captures global trend anomalies via a coarse-grained anomaly collector (CgAC), focuses on local abrupt changes with a fine-grained anomaly collector (FgAC), and reconstructs the sequence using multi-grained features through a reconstruction module (RS), employing reconstruction error for anomaly detection. Experiments on seven public benchmark datasets show that the full C2FAD model surpasses existing methods, achieving gains of 6.14%, 9.02%, and 3.81% in AUC, F1, and AFF-F1, respectively, over the best baseline. In the future, the research aims to integrate self-supervised pre-training and causal inference mechanisms into C2FAD to improve its representation and generalization ability in scenarios with limited labeled data.

## VI. Acknowledgment

This work is supported by The Research Foundation of Chongqing Normal University (Grant No.22XLB025), The Open Foundation of Yunnan Key Laboratory of Software Engineering (Grant No.2023SE204), The Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No.KJQN202300522, KJZD-K202500506)

## References

[1] M. Ouhssini, K. Afdel, E. Agherrabi, M. Akouhar, A. Abarda, Deepdefend: A comprehensive framework for ddos attack detection and prevention in cloud computing, Journal of King Saud University-Computer and Information Sciences 36 (2024) 101938.

[2] S. R. Banu, T. N. Gongada, K. Santosh, H. Chowdhary, R. Sabareesh, S. Muthuperumal, Financial fraud detection using hybrid convolutional and recurrent neural networks: An analysis of unstructured data in banking, in: 2024 10th International Conference on Communication and Signal Processing (ICCSP), IEEE, 2024, pp. 1027–1031.

[3] H. Ni, S. Meng, X. Geng, P. Li, Z. Li, X. Chen, X. Wang, S. Zhang, Time series modeling for heart rate prediction: From arima to transformers, in: 2024 6th International Conference on Electronic Engineering and Informatics (EEI), IEEE, 2024, pp. 584–589.

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

[5] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International conference on learning representations, 2018.

[6] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, arXiv preprint arXiv:2201.07284 (2022).

[7] 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: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2828–2837.

[8] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, Madgan: Multivariate anomaly detection for time series data with generative adversarial networks, in: International conference on artificial neural networks, Springer, 2019, pp. 703–716.

[9] 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, in: 2020 IEEE international conference on data mining (ICDM), IEEE, 2020, pp. 841–850.

[10] 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.

[11] S.-E. Benkabou, K. Benabdeslem, V. Kraus, K. Bourhis, B. Canitia, Local anomaly detection for multivariate time series by temporal dependency based on poisson model, IEEE Transactions on Neural Networks and Learning Systems 33 (2021) 6701–6711.

[12] L.-r. Yu, Q.-h. Lu, Y. Xue, Dtaad: Dual tcn-attention networks for anomaly detection in multivariate time series data, Knowledge-Based Systems 295 (2024) 111849.

[13] X. Zhou, C. Dai, W. Wang, T. Qiu, Global–local association discrepancy for multivariate time series anomaly detection in iiot, IEEE Internet of Things Journal 11 (2023) 11287–11297.

[14] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, S. Pan, A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[15] J. Xu, H. Wu, J. Wang, M. Long, Anomaly transformer: Time series anomaly detection with association discrepancy, arXiv preprint arXiv:2110.02642 (2021).

[16] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, M. Salehi, Carla: Self-supervised contrastive representation learning for time series anomaly detection, Pattern Recognition 157 (2025) 110874.

[17] Z. Zhong, Z. Yu, X. Xi, Y. Xu, W. Cao, Y. Yang, K. Yang, J. You, Simad: A simple dissimilarity-based approach for time-series anomaly detection, IEEE Transactions on Neural Networks and Learning Systems (2025).

[18] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 387–395.

[19] G. B. Moody, R. G. Mark, The impact of the mit-bih arrhythmia database, IEEE engineering in medicine and biology magazine 20 (2001) 45–50.

[20] S. Nedelkoski, J. Bogatinovski, A. K. Mandapati, S. Becker, J. Cardoso, O. Kao, Multi-source distributed system data for ai-powered analytics, in: European conference on service-oriented and cloud computing, Springer, 2020, pp. 161–176.

[21] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147.

[22] A. Abdulaal, Z. Liu, T. Lancewicki, Practical approach to asynchronous multivariate time series anomaly detection and localization, in: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, 2021, pp. 2485–2494.

[23] K. Doshi, S. Abudalou, Y. Yilmaz, Reward once, penalize once: Rectifying time series anomaly detection, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–8.

[24] R. Wu, E. J. Keogh, Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress, IEEE transactions on knowledge and data engineering 35 (2021) 2421–2429.

[25] A. Huet, J. M. Navarro, D. Rossi, Local evaluation of time series anomaly detection algorithms, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 635–645.

[26] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 4027–4035.

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